SOLUTIONS MANUAL

NETWORK SECURITY ESSENTIALS: APPLICATIONS AND STANDARDS FIFTH EDITION

WILLIAM STALLINGS

NOTICE

This manual contains solutions to the review questions and homework problems in *Cryptography and Network Security, Sixth Edition*. If you spot an error in a solution or in the wording of a problem, I would greatly appreciate it if you would forward the information via email to wllmst@me.net. An errata sheet for this manual, if needed, is available at http://www.box.net/shared/nh8hti5167 . File name is S-NetSec5e-mmyy

W.S.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

ANSWERS TO QUESTIONS

1.1 The OSI Security Architecture is a framework that provides a systematic way of defining the requirements for security and characterizing the approaches to satisfying those requirements. The document defines security attacks, mechanisms, and services, and the relationships among these categories.

1.2 **Passive attacks** have to do with eavesdropping on, or monitoring, transmissions. Electronic mail, file transfers, and client/server exchanges are examples of transmissions that can be monitored. **Active attacks** include the modification of transmitted data and attempts to gain unauthorized access to computer systems.

1.3 **Passive attacks:** release of message contents and traffic analysis.
Active attacks: masquerade, replay, modification of messages, and denial of service.

1.4 **Authentication:** The assurance that the communicating entity is the one that it claims to be.
Access control: The prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).
Data confidentiality: The protection of data from unauthorized disclosure.
Data integrity: The assurance that data received are exactly as sent by an authorized entity (i.e., contain no modification, insertion, deletion, or replay).
Nonrepudiation: Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.
Availability service: The property of a system or a system resource being accessible and usable upon demand by an authorized system entity, according to performance specifications for the system (i.e., a system is available if it provides services according to the system design whenever users request them).

1.5 See Table 1.3.

ANSWERS TO PROBLEMS

1.1 The system must keep personal identification numbers confidential, both in the host system and during transmission for a transaction. It must protect the integrity of account records and of individual transactions. Availability of the host system is important to the economic well being of the bank, but not to its fiduciary responsibility. The availability of individual teller machines is of less concern.

1.2 The system does not have high requirements for integrity on individual transactions, as lasting damage will not be incurred by occasionally losing a call or billing record. The integrity of control programs and configuration records, however, is critical. Without these, the switching function would be defeated and the most important attribute of all - availability - would be compromised. A telephone switching system must also preserve the confidentiality of individual calls, preventing one caller from overhearing another.

1.3 a. The system will have to assure confidentiality if it is being used to publish corporate proprietary material.

b. The system will have to assure integrity if it is being used to laws or regulations.

c. The system will have to assure availability if it is being used to publish a daily paper.

1.4 a. An organization managing public information on its web server determines that there is no potential impact from a loss of confidentiality (i.e., confidentiality requirements are not applicable), a moderate potential impact from a loss of integrity, and a moderate potential impact from a loss of availability.

b. A law enforcement organization managing extremely sensitive investigative information determines that the potential impact from a loss of confidentiality is high, the potential impact from a loss of integrity is moderate, and the potential impact from a loss of availability is moderate.

c. A financial organization managing routine administrative information (not privacy-related information) determines that the potential impact from a loss of confidentiality is low, the potential impact from a loss of integrity is low, and the potential impact from a loss of availability is low.

d. The management within the contracting organization determines that: (i) for the sensitive contract information, the potential impact from a loss of confidentiality is moderate, the potential impact from a loss of

integrity is moderate, and the potential impact from a loss of availability is low; and (ii) for the routine administrative information (non-privacy-related information), the potential impact from a loss of confidentiality is low, the potential impact from a loss of integrity is low, and the potential impact from a loss of availability is low.

e. The management at the power plant determines that: (i) for the sensor data being acquired by the SCADA system, there is no potential impact from a loss of confidentiality, a high potential impact from a loss of integrity, and a high potential impact from a loss of availability; and (ii) for the administrative information being processed by the system, there is a low potential impact from a loss of confidentiality, a low potential impact from a loss of integrity, and a low potential impact from a loss of availability. Examples from FIPS 199.

1.5	**Release of message contents**	**Traffic analysis**	**Masquerade**	**Replay**	**Modification of messages**	**Denial of service**
Peer entity authentication			**Y**			
Data origin authentication			**Y**			
Access control			**Y**			
Confidentiality	**Y**					
Traffic flow confidentiality		**Y**				
Data integrity				**Y**	**Y**	
Non-repudiation			**Y**			
Availability						**Y**

1.6	Release of message contents	Traffic analysis	Masquerade	Replay	Modification of messages	Denial of service
Encipherment	**Y**					
Digital signature			**Y**	**Y**	**Y**	
Access control	**Y**	**Y**	**Y**	**Y**		**Y**
Data integrity				**Y**	**Y**	
Authentication exchange	**Y**		**Y**	**Y**		**Y**
Traffic padding		**Y**				
Routing control	**Y**	**Y**				**Y**
Notarization			**Y**	**Y**	**Y**	

CHAPTER 2 SYMMETRIC ENCRYPTION AND MESSAGE CONFIDENTIALITY

ANSWERS TO QUESTIONS

2.1 Plaintext, encryption algorithm, secret key, ciphertext, decryption algorithm.

2.2 Permutation and substitution.

2.3 One secret key.

2.4 A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time. A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

2.5 Cryptanalysis and brute force.

2.6 In some modes, the plaintext does not pass through the encryption function, but is XORed with the output of the encryption function. The math works out that for decryption in these cases, the encryption function must also be used.

2.7 With triple encryption, a plaintext block is encrypted by passing it through an encryption algorithm; the result is then passed through the same encryption algorithm again; the result of the second encryption is passed through the same encryption algorithm a third time. Typically, the second stage uses the decryption algorithm rather than the encryption algorithm.

2.8 There is no cryptographic significance to the use of decryption for the second stage. Its only advantage is that it allows users of 3DES to decrypt data encrypted by users of the older single DES by repeating the key.

ANSWERS TO PROBLEMS

2.1 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  LIDLP  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.

2.2 a. Let –X be the additive inverse of X. That is –X $\boxed{+}$ X = 0. Then:

$$P = (C \boxed{+} -K_1) \oplus K_0$$

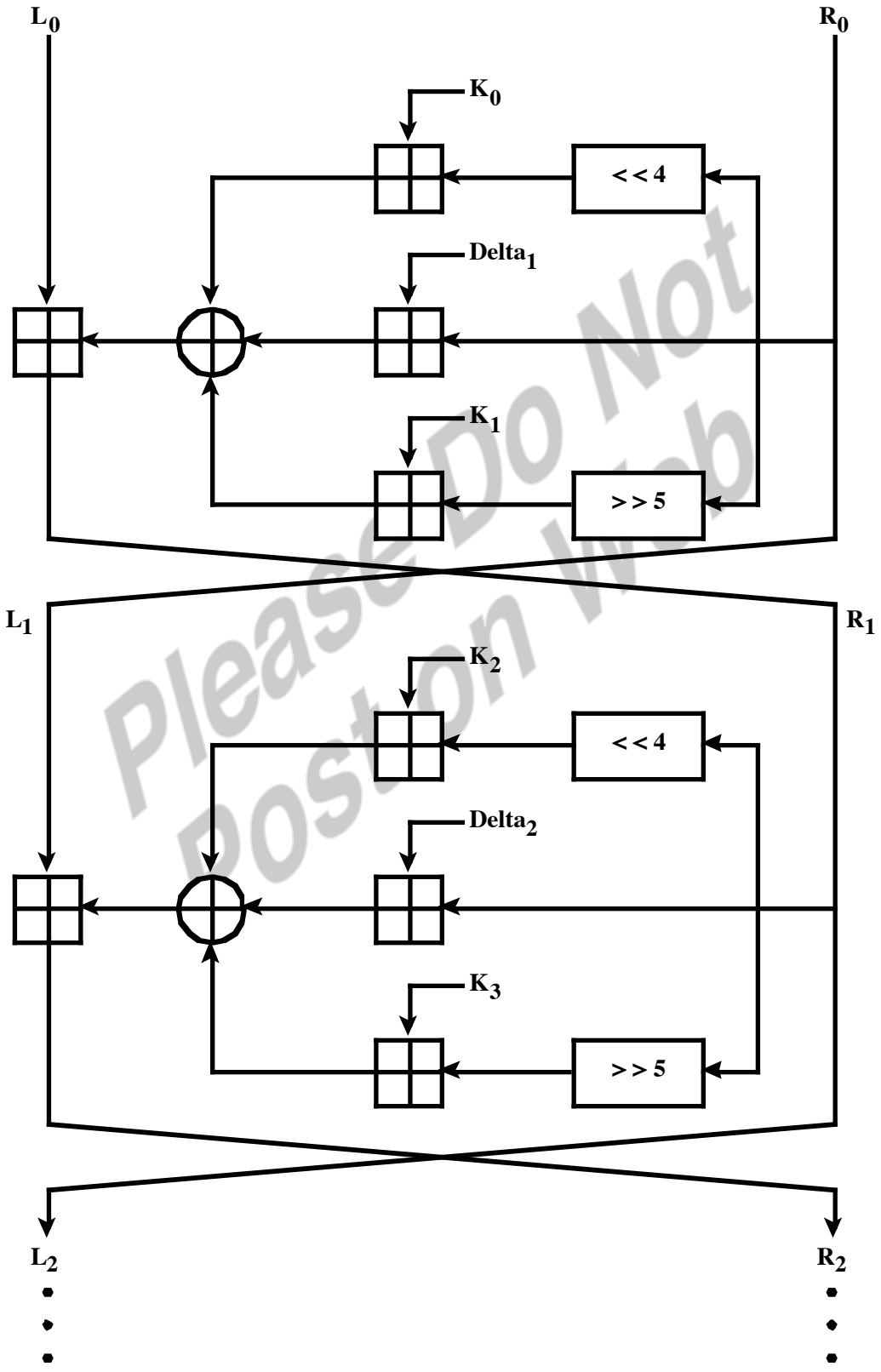
b. First, calculate –C'. Then –C' = (P' $\oplus$ K_0) $\boxed{+}$ (–K_1). We then have:

C $\boxed{+}$ –C' = (P $\oplus$ K_0) $\boxed{+}$ (P' $\oplus$ K_0)

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

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

b. First two r.
ounds:



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

$L_2 = L_0 \boxed{+} [(R_0 << 4) \boxed{+} K_0] \oplus [R_0 \boxed{+} \delta_1] \oplus [(R_0 >> 5) \boxed{+} K_1]$

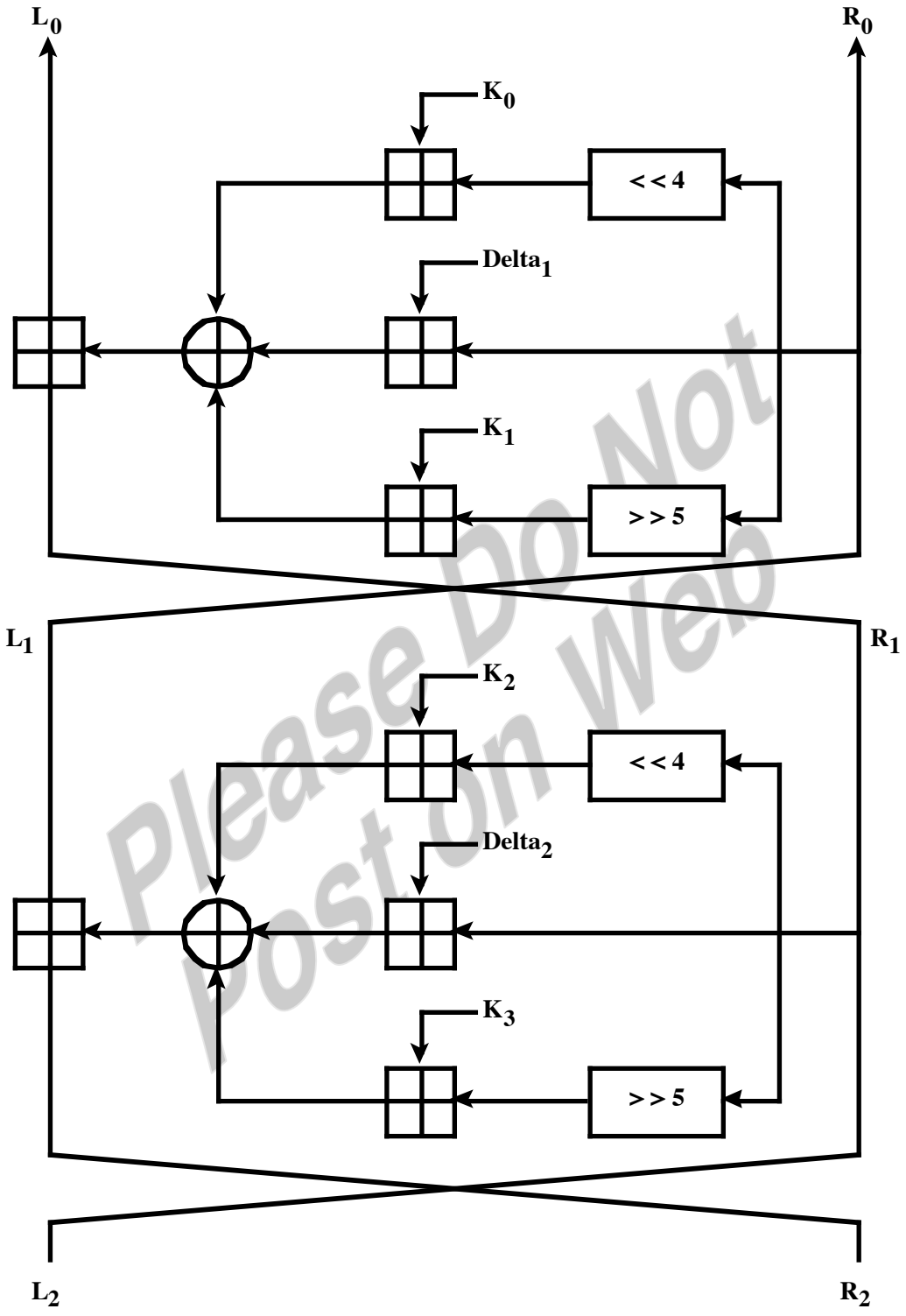
$R_2 = R_0 \boxed{+} [(L_2 << 4) \boxed{+} K_2] \oplus [L_2 \boxed{+} \delta_2] \oplus [(L_2 >> 5) \boxed{+} K_3]$

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.

$R_0 = R_2 \boxed{+} [(L_2 << 4) \boxed{+} K_2] \oplus [L_2 \boxed{+} \delta_2] \oplus [(L_2 >> 5) \boxed{+} K_3]$

$L_0 = L_2 \boxed{+} [(R_0 << 4) \boxed{+} K_0] \oplus [R_0 \boxed{+} \delta_1] \oplus [(R_0 >> 5) \boxed{+} K_1]$

d.



2.4 To see that the same algorithm with a reversed key order produces the correct result, consider the Figure 2.2, which shows the encryption process going down the left-hand side and the decryption process going up the right-hand side for a 16-round algorithm (the result would be the same for any number of rounds). For clarity, we use the notation LE_i

and RE_i for data traveling through the encryption algorithm and LD_i and RD_i for data traveling through the decryption algorithm. The diagram indicates that, at every round, the intermediate value of the decryption process is equal to the corresponding value of the encryption process with the two halves of the value swapped. To put this another way, let the output of the ith encryption round be $LE_i||RE_i$ (L_i concatenated with R_i). Then the corresponding input to the (16 – i)th decryption round is $RD_i||LD_i$.

Let us walk through the figure to demonstrate the validity of the preceding assertions. To simplify the diagram, it is unwrapped, not showing the swap that occurs at the end of each iteration. But note that the intermediate result at the end of the ith stage of the encryption process is the 2w-bit quantity formed by concatenating LE_i and RE_i, and that the intermediate result at the end of the ith stage of the decryption process is the 2w-bit quantity formed by concatenating LD_i and RD_i. After the last iteration of the encryption process, the two halves of the output are swapped, so that the ciphertext is $RE_{16}||LE_{16}$. The output of that round is the ciphertext. Now take that ciphertext and use it as input to the same algorithm. The input to the first round is $RE_{16}||LE_{16}$, which is equal to the 32-bit swap of the output of the sixteenth round of the encryption process.

Now we would like to show that the output of the first round of the decryption process is equal to a 32-bit swap of the input to the sixteenth round of the encryption process. First, consider the encryption process. We see that:

$$LE_{16} = RE_{15}$$
$$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$$

On the decryption side:

$$LD_1 = RD_0 = LE_{16} = RE_{15}$$
$$RD_1 = LD_0 \oplus F(RD_0, K_{16})$$
$$= RE_{16} \oplus F(RE_{15}, K_{16})$$
$$= [LE_{15} \oplus F(RE_{15},K_{16})] \oplus F(RE_{15}, K_{16})$$

The XOR has the following properties:

$$[A \oplus B] \oplus C = A \oplus [B \oplus C]$$
$$D \oplus D = 0$$
$$E \oplus 0 = E$$

Thus, we have $LD_1 = RE_{15}$ and $RD_1 = LE_{15}$. Therefore, the output of the first round of the decryption process is $LE_{15}||RE_{15}$, which is the 32-

bit swap of the input to the sixteenth round of the encryption. This correspondence holds all the way through the 16 iterations, as is easily shown. We can cast this process in general terms. For the ith iteration of the encryption algorithm:

$$LE_i = RE_{i-1}$$
$$RE_i = LE_{i-1} \oplus F(RE_{i-1}, K_i)$$

Rearranging terms:

$$RE_{i-1} = LE_i$$
$$LE_{i-1} = RE_i \oplus F(RE_{i-1}, K_i) = RE_i \oplus F(LE_i, K_i)$$

Thus, we have described the inputs to the ith iteration as a function of the outputs, and these equations confirm the assignments shown in the right-hand side of the following figure.

Finally, we see that the output of the last round of the decryption process is $RE_0||LE_0$. A 32-bit swap recovers the original plaintext, demonstrating the validity of the Feistel decryption process.

2.5 Because of the key schedule, the round functions used in rounds 9 through 16 are mirror images of the round functions used in rounds 1 through 8. From this fact we see that encryption and decryption are identical. We are given a ciphertext *c*. Let *m*' = *c*. Ask the encryption oracle to encrypt *m*'. The ciphertext returned by the oracle will be the decryption of *c*.

2.6 For $1 \le i \le 128$, take $c_i \in \{0, 1\}^{128}$ to be the string containing a 1 in position i and then zeros elsewhere. Obtain the decryption of these 128 ciphertexts. Let $m_1, m_2, \ldots, m_{128}$ be the corresponding plaintexts. Now, given any ciphertext c which does not consist of all zeros, there is a unique nonempty subset of the c_i's which we can XOR together to obtain c. Let $I(c) \subseteq \{1, 2, \ldots, 128\}$ denote this subset. Observe

$$c = \bigoplus_{i \in I(c)} c_i = \bigoplus_{i \in I(c)} E(m_i) = E\left(\bigoplus_{i \in I(c)} m_i \right)$$

Thus, we obtain the plaintext of c by computing $\bigoplus_{i \in I(c)} m_i$. Let **0** be the all-zero string. Note that **0** = **0** ⊕ **0**. From this we obtain E(**0**) = E(**0** ⊕ **0**) = E(**0**) ⊕ E(**0**) = **0**. Thus, the plaintext of c = **0** is m = **0**. Hence we can decrypt every $c \in \{0, 1\}^{128}$.

2.7 a.

Pair	Probability
00	$(0.5 - \partial)^2 = 0.25 - \partial + \partial^2$
01	$(0.5 - \partial) \times (0.5 + \partial) = 0.25 - \partial^2$
10	$(0.5 + \partial) \times (0.5 - \partial) = 0.25 - \partial^2$
11	$(0.5 + \partial)^2 = 0.25 + \partial + \partial^2$

b. Because 01 and 10 have equal probability in the initial sequence, in the modified sequence, the probability of a 0 is 0.5 and the probability of a 1 is 0.5.

c. The probability of any particular pair being discarded is equal to the probability that the pair is either 00 or 11, which is $0.5 + 2\partial^2$, so the expected number of input bits to produce x output bits is $x/(0.25 - \partial^2)$.

d. The algorithm produces a totally predictable sequence of exactly alternating 1's and 0's.

2.8 a. For the sequence of input bits $a_1, a_2, \ldots, a_n$, the output bit b is defined as:

$$b = a_1 \oplus a_2 \oplus \ldots \oplus a_n$$

b. $0.5 - 2\partial^2$

c. $0.5 - 8\partial^4$

d. The limit as n goes to infinity is 0.5.

2.9 Use a key of length 255 bytes. The first two bytes are zero; that is K[0] = K[1] = 0. Thereafter, we have: K[2] = 255; K[3] = 254; … K[255]= 2.

2.10 a. Simply store i, j, and S, which requires 8 + 8 + (256 × 8) = 2064 bits

b. The number of states is $[256! \times 256^2] \approx 2^{1700}$. Therefore, 1700 bits are required.

2.11 a. By taking the first 80 bits of $v \| c$, we obtain the initialization vector, v. Since v, c, k are known, the message can be recovered (i.e., decrypted) by computing RC4($v \| k$) $\oplus$ c.

b. If the adversary observes that $v_i = v_j$ for distinct i, j then he/she knows that the same key stream was used to encrypt both m_i and m_j. In this case, the messages m_i and m_j may be vulnerable to the type of cryptanalysis carried out in part (a).

c. Since the key is fixed, the key stream varies with the choice of the 80-bit v, which is selected randomly. Thus, after approximately

$\sqrt{\frac{\pi}{2}2^{80}} \approx 2^{40}$ messages are sent, we expect the same *v*, and hence the same key stream, to be used more than once.

d. The key *k* should be changed sometime before 2^{40} messages are sent.

2.12 a. No. For example, suppose C_1 is corrupted. The output block P_3 depends only on the input blocks C_2 and C_3.

b. An error in P_1 affects C_1. But since C_1 is input to the calculation of C_2, C_2 is affected. This effect carries through indefinitely, so that all ciphertext blocks are affected. However, at the receiving end, the decryption algorithm restores the correct plaintext for blocks except the one in error. You can show this by writing out the equations for the decryption. Therefore, the error only effects the corresponding decrypted plaintext block.

2.13 In CBC encryption, the input block to each forward cipher operation (except the first) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel. In CBC decryption, however, the input blocks for the inverse cipher function (i.e., the ciphertext blocks) are immediately available, so that multiple inverse cipher operations can be performed in parallel.

2.14 If an error occurs in transmission of ciphertext block C_i, then this error propagates to the recovered plaintext blocks P_i and P_{i+1}.

2.15 After decryption, the last byte of the last block is used to determine the amount of padding that must be stripped off. Therefore there must be at least one byte of padding.

2.16 a. Assume that the last block of plaintext is only *L* bytes long, where $L < 2w/8$. The encryption sequence is as follows (The description in RFC 2040 has an error; the description here is correct.):

1. Encrypt the first $(N - 2)$ blocks using the traditional CBC technique.
2. XOR P_{N-1} with the previous ciphertext block C_{N-2} to create Y_{N-1}.
3. Encrypt Y_{N-1} to create E_{N-1}.
4. Select the first *L* bytes of E_{N-1} to create C_N.
5. Pad P_N with zeros at the end and exclusive-OR with E_{N-1} to create Y_N.
6. Encrypt Y_N to create C_{N-1}.

The last two blocks of the ciphertext are C_{N-1} and C_N.

b. $P_{N-1} = C_{N-2} \oplus D(K, [C_N \| X])$

$P_N \| X = (C_N \| 00\ldots0) \oplus D(K, [C_{N-1}])$

P_N = left-hand portion of $(P_N \| X)$

where $\|$ is the concatenation function

2.17 a. Assume that the last block (P_N) has j bits. After encrypting the last full block (P_{N-1}), encrypt the ciphertext (C_{N-1}) again, select the leftmost j bits of the encrypted ciphertext, and XOR that with the short block to generate the output ciphertext.

b. While an attacker cannot recover the last plaintext block, he can change it systematically by changing individual bits in the ciphertext. If the last few bits of the plaintext contain essential information, this is a weakness.

2.18 Nine plaintext characters are affected. The plaintext character corresponding to the ciphertext character is obviously altered. In addition, the altered ciphertext character enters the shift register and is not removed until the next eight characters are processed.

CHAPTER 3 PUBLIC-KEY CRYPTOGRAPHY AND MESSAGE AUTHENTICATION

ANSWERS TO QUESTIONS

3.1 Message encryption, message authentication code, hash function.

3.2 An authenticator that is a cryptographic function of both the data to be authenticated and a secret key.

3.3 **(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.

3.4

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. This is sometimes referred to in the literature as the **one-way** property.
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).

3.5 The compression function is the fundamental module, or basic building block, of a hash function. The hash function consists of iterated application of the compression function.

3.6 **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.

3.7 **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.

3.8 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**.

3.9 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.

ANSWERS TO PROBLEMS

3.1 **a.** Yes. The XOR function is simply a vertical parity check. If there is an odd number of errors, then there must be at least one column that contains an odd number of errors, and the parity bit for that column will detect the error. Note that the RXOR function also catches all errors caused by an odd number of error bits. Each RXOR bit is a function of a unique "spiral" of bits in the block of data. If there is an odd number of errors, then there must be at least one spiral that

contains an odd number of errors, and the parity bit for that spiral will detect the error.

b. No. The checksum will fail to detect an even number of errors when both the XOR and RXOR functions fail. In order for both to fail, the pattern of error bits must be at intersection points between parity spirals and parity columns such that there is an even number of error bits in each parity column and an even number of error bits in each spiral.

c. It is too simple to be used as a secure hash function; finding multiple messages with the same hash function would be too easy.

3.2 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.

3.3 **a.** 1 bit

b. 1024 bits

c. 1023 bits

3.4 **a.** 1919

b. 1920

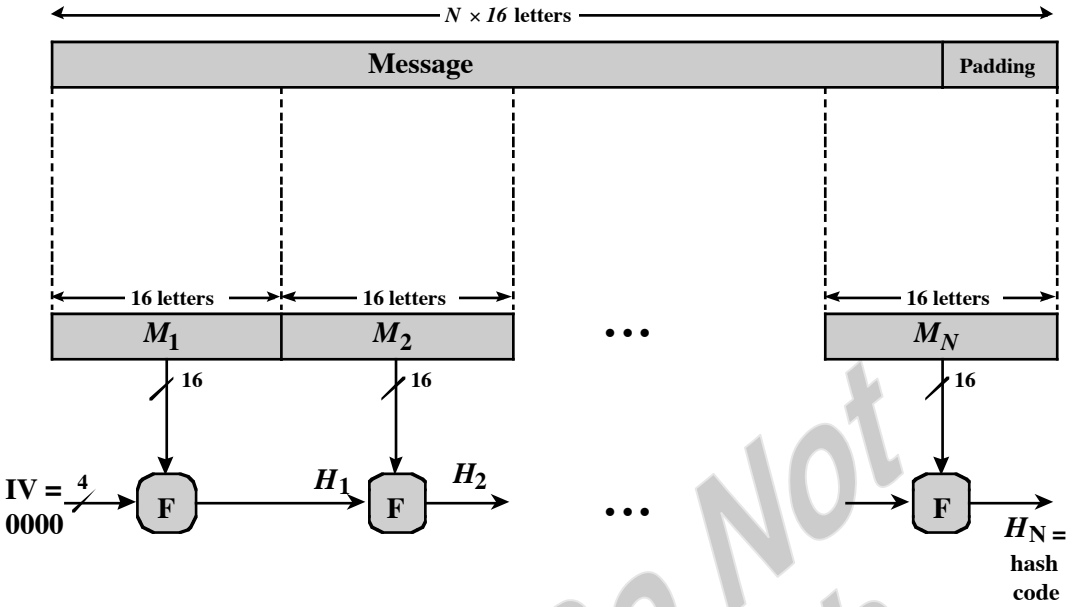
c. 1921

3.5 **a.** It satisfies properties 1 through 3 but not the remaining properties. For example, for property 4, a message consisting of the value h satisfies $H(h) = h$. For property 5, take any message M and add the decimal digit 0 to the sequence; it will have the same hash value.

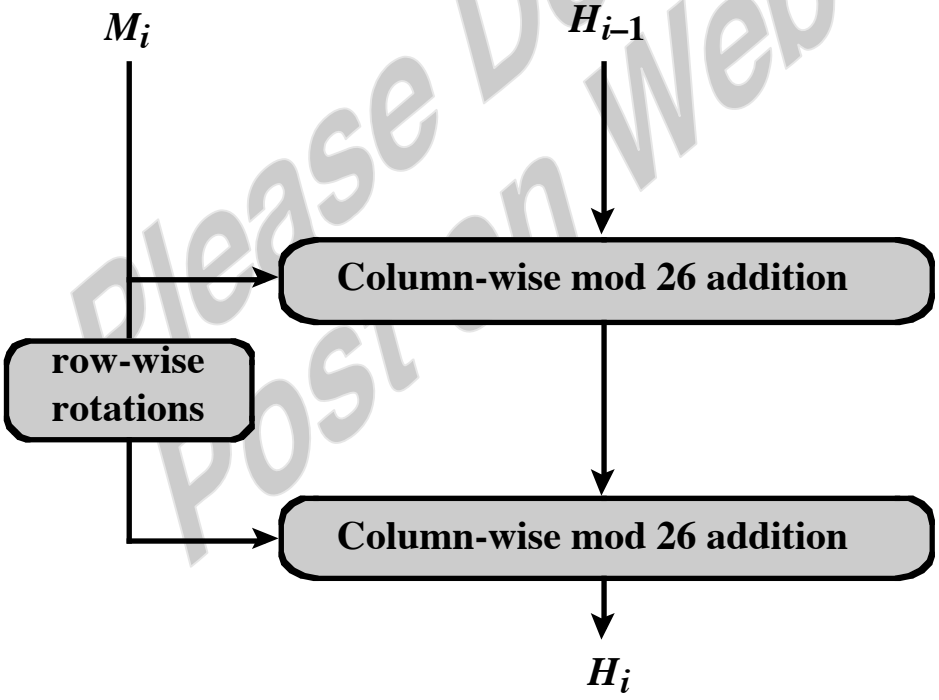
b. It satisfies properties 1 through 3. Property 4 is also satisfied if n is a large composite number, because taking square roots modulo such an integer n is considered to be infeasible. Properties 5 and 6 are not satisfied because –M will have the same value as M.

c. 955

3.6 a. Overall structure:



Compression function F:



b. BFQG

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

AYHGDAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAA

3.7 If you examine the structure of a single round of DES, you see that the round includes a one-way function, F, and an XOR:

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$

For DES, the function F is depicted in Figure 2.2. It maps a 32-bit R and a 48-bit K into a 32-bit output. That is, it maps an 80-bit input into a 32-bit output. This is clearly a one-way function. Any hash function that produces a 32-bit output could be used for F. The demonstration in the text that decryption works is still valid for any one-way function F.

3.8 The opponent has the two-block message B1, B2 and its hash RSAH(B1, B2). The following attack will work. Choose an arbitrary C1 and choose C2 such that:

$$C2 = RSA(C1) \oplus RSA(B1) \oplus B2$$

then

$$RSA(C1) \oplus C2 = RSA(C1) \oplus RSA(C1) \oplus RSA(B1) \oplus B2$$
$$= RSA(B1) \oplus B2$$

so $RSAH(C1, C2) = RSA[RSA(C1) \oplus C2)] = RSA[RSA(B1) \oplus B2]$
$= RSAH(B1, B2)$

3.9 The CBC mode with an IV of 0 and plaintext blocks D1, D2, . . ., Dn and 64-bit CFB mode with IV = D1 and plaintext blocks D2, D3, . . ., Dn yield the same result.

3.10 **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.

3.11 **a.** Two quantities are precomputed:

$f(IV, (K^+ \oplus ipad))$
$f(IV, (K^+ \oplus opad))$

where f(cv, block) is the compression function for the hash function, which takes as arguments a chaining variable of *n* bits and a block of *b* bits and produces a chaining variable of *n* bits. These quantities only need to be computed initially and every time the key changes. In effect, the precomputed quantities substitute for the initial value (IV) in the hash function. With this implementation,

only one additional instance of the compression function is added to the processing normally produced by the hash function.

b. This is a more efficient implementation. This more efficient implementation is especially worthwhile if most of the messages for which a MAC is computed are short.

3.12 We use Figure 3.7a but put the XOR with K_1 after the final encryption. For this problem, there are two blocks to process. The output of the encryption of the first message block is E(K, **0**) = CBC(K, **0**) = $T_0 \oplus K_1$. This is XORed with the second message block ($T_0 \oplus T_1$), so that the input to the second encryption is ($T_1 \oplus K_1$) = CBC(K, **1**) = E(K, **1**). So the output of the second encryption is E(K, [E(K, **1**)]) = CBC(K, [CBC(K, **1**)]) = $T_2 \oplus K_1$. After the final XOR with K_1, we get VMAC(K, [**0** $\|$ ($T_0 \oplus T_1$)]) = T_2.

3.13

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.

3.14 **a.** $n = 33$; $\phi(n) = 20$; $d = 3$; C = 26.

b. $n = 55$; $\phi(n) = 40$; $d = 27$; C = 14.

c. $n = 77$; $\phi(n) = 60$; $d = 53$; C = 57.

d. $n = 143$; $\phi(n) = 120$; $d = 11$; C = 106.

e. $n = 527$; $\phi(n) = 480$; $d = 343$; C = 128. For decryption, we have

$$128^{343} \bmod 527 = 128^{256} \times 128^{64} \times 128^{16} \times 128^{4} \times 128^{2} \times 128^{1} \bmod 527$$
$$= 35 \times 256 \times 35 \times 101 \times 47 \times 128 = 2 \bmod 527$$
$$= 2 \bmod 257$$

3.15 $M = 5$

3.16 $d = 3031$

3.17 Yes. If a plaintext block has a common factor with n, modulo n then the encoded block will also have a common factor with n, modulo n. Because we encode blocks that are smaller than pq, the factor must be p or q and the plaintext block must be a multiple of p or q. We can test each block for primality. If prime, it is p or q. In this case we divide into n to find the other factor. If not prime, we factor it and try the factors as divisors of n.

3.18 Refer to Figure 3.10 The private key k is the pair $\{d, n\}$; the public key x is the pair $\{e, n\}$; the plaintext p is M; and the ciphertext z is C. M1 is formed by calculating $d = e^{-1} \bmod \phi(n)$. M2 consists of raising M to the power e (mod n). M2 consists of raising C to the power d (mod n).

3.19 Yes.

3.20 Consider a set of alphabetic characters {A, B, …, Z}. The corresponding integers, representing the position of each alphabetic character in the alphabet, form a set of message block values SM = {0, 1, 2, …, 25}. The set of corresponding ciphertext block values SC = $\{0^e \bmod N, 1^e \bmod N, \ldots, 25^e \bmod N\}$, and can be computed by everybody with the knowledge of the public key of Bob.

Thus, the most efficient attack against the scheme described in the problem is to compute $M^e \bmod N$ for all possible values of M, then create a look-up table with a ciphertext as an index, and the corresponding plaintext as a value of the appropriate location in the table.

3.21 **a.** $X_A = 6$

b. K = 3

CHAPTER 4 KEY DISTRIBUTION AND USER AUTHENTICATION

ANSWERS TO QUESTIONS

4.1 For two parties A and B, key distribution can be achieved in a number of ways, as follows:

1. A can select a key and physically deliver it to B.
2. A third party can select the key and physically deliver it to A and B.
3. If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
4. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B.

4.2 A **session key** is a temporary encryption key used between two principals. A **master key** is a long-lasting key that is used between a key distribution center and a principal for the purpose of encoding the transmission of session keys. Typically, the master keys are distributed by noncryptographic means.

4.3 A key distribution center is a system that is authorized to transmit temporary session keys to principals. Each session key is transmitted in encrypted form, using a master key that the key distribution center shares with the target principal.

4.4 A full-service Kerberos environment consists of a Kerberos server, a number of clients, and a number of application servers.

4.5 A realm is an environment in which: **1.** The Kerberos server must have the user ID (UID) and hashed password of all participating users in its database. All users are registered with the Kerberos server. **2.** The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server.

4.6 Version 5 overcomes some environmental shortcomings and some technical deficiencies in Version 4.

4.7 A nonce is a value that is used only once, such as a timestamp, a counter, or a random number; the minimum requirement is that it differs with each transaction.

4.8 **1.** The distribution of public keys. **2.** The use of public-key encryption to distribute secret keys

4.9 **1.** The authority maintains a directory with a {name, public key} entry for each participant. **2.** Each participant registers a public key with the directory authority. Registration would have to be in person or by some form of secure authenticated communication. **3.** A participant may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way. **4.** Periodically, the authority publishes the entire directory or updates to the directory. For example, a hard-copy version much like a telephone book could be published, or updates could be listed in a widely circulated newspaper. **5.** Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory.

4.10 A public-key certificate contains a public key and other information, is created by a certificate authority, and is given to the participant with the matching private key. A participant conveys its key information to another by transmitting its certificate. Other participants can verify that the certificate was created by the authority.

4.11 **1.** Any participant can read a certificate to determine the name and public key of the certificate's owner. **2.** Any participant can verify that the certificate originated from the certificate authority and is not counterfeit. **3.** Only the certificate authority can create and update certificates. **4.** Any participant can verify the currency of the certificate.

4.12 X.509 defines a framework for the provision of authentication services by the X.500 directory to its users. The directory may serve as a repository of public-key certificates. Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.

4.13 A chain of certificates consists of a sequence of certificates created by different certification authorities (CAs) in which each successive certificate is a certificate by one CA that certifies the public key of the next CA in the chain.

4.14 The owner of a public-key can issue a certificate revocation list that revokes one or more certificates.

ANSWERS TO PROBLEMS

4.1 i) Sending to the server the source name A, the destination name Z (his own), and E(K_a, R), as if A wanted to send him the same message encrypted under the same key R as A did it with B

ii) The server will respond by sending E(K_z, R) to A and Z will intercept that

iii) because Z knows his key K_z, he can decrypt E(K_z, R), thus getting his hands on R that can be used to decrypt E(R, M) and obtain M.

4.2 All three really serve the same purpose. The difference is in the vulnerability. In **Usage 1**, an attacker could breach security by inflating N_a and withholding an answer from B for future replay attack, a form of suppress-replay attack. The attacker could attempt to predict a plausible reply in **Usage 2**, but this will not succeed if the nonces are random. In both Usage 1 and 2, the messages work in either direction. That is, if N is sent in either direction, the response is E[K, N]. In **Usage 3**, the message is encrypted in both directions; the purpose of function f is to assure that messages 1 and 2 are not identical. Thus, Usage 3 is more secure.

4.3 An error in C_1 affects P_1 because the encryption of C_1 is XORed with IV to produce P_1. Both C_1 and P_1 affect P_2, which is the XOR of the encryption of C_2 with the XOR of C_1 and P_1. Beyond that, P_{N-1} is one of the XORed inputs to forming P_N.

4.4 Let us consider the case of the interchange of C_1 and C_2. The argument will be the same for any other adjacent pair of ciphertext blocks. First, if C_1 and C_2 arrive in the proper order:

$P_1 = E[K, C_1] \oplus IV$
$P_2 = E[K, C_2] \oplus C_1 \oplus P_1 = E[K, C_2] \oplus C_1 \oplus E[K, C_1] \oplus IV$
$P_3 = E[K, C_3] \oplus C_2 \oplus P_2 = E[K, C_3] \oplus C_2 \oplus E[K, C_2] \oplus C_1 \oplus E[K, C_1] \oplus IV$

Now suppose that C_1 and C_2 arrive in the reverse order. Let us refer to the decrypted blocks as Q_i.

$Q_1 = E[K, C_2] \oplus IV$
$Q_2 = E[K, C_1] \oplus C_2 \oplus Q_1 = E[K, C_1] \oplus C_2 \oplus E[K, C_2] \oplus IV$
$Q_3 = E[K, C_3] \oplus C_1 \oplus Q_2 = E[K, C_3] \oplus C_1 \oplus E[K, C_1] \oplus C_2 \oplus E[K, C_2] \oplus IV$

The result is that $Q_1 \neq P_1$; $Q_2 \neq P_2$; but $Q_3 = P_3$. Subsequent blocks are clearly unaffected.

4.5 The problem has a simple fix, namely the inclusion of the name of B in the signed information for the third message, so that the third message now reads:

$A \rightarrow B: \quad A \{r_B, B\}$

4.6 **a.** This is a means of authenticating A to B. R_1 serves as a challenge, and only A is able to encrypt R_1 so that it can be decrypted with A's public key.

b. Someone (e.g., C) can use this mechanism to get A to sign a message. Then, C will present this signature to D along with the message, claiming it was sent by A. This is a problem if A uses its public/private key for both authentication, signatures, etc.

4.7 **a.** This is a means of authenticating A to B. Only A can decrypt the second message, to recover R_2.

b. Someone (e.g. C) can use this mechanism to get A to decrypt a message (i.e., send that message as R_2) that it has eavesdropped from the network (originally sent to A).

4.8 It contains the Alice's ID, Bob's name, and timestamp encrypted by the KDC-Bob secret key.

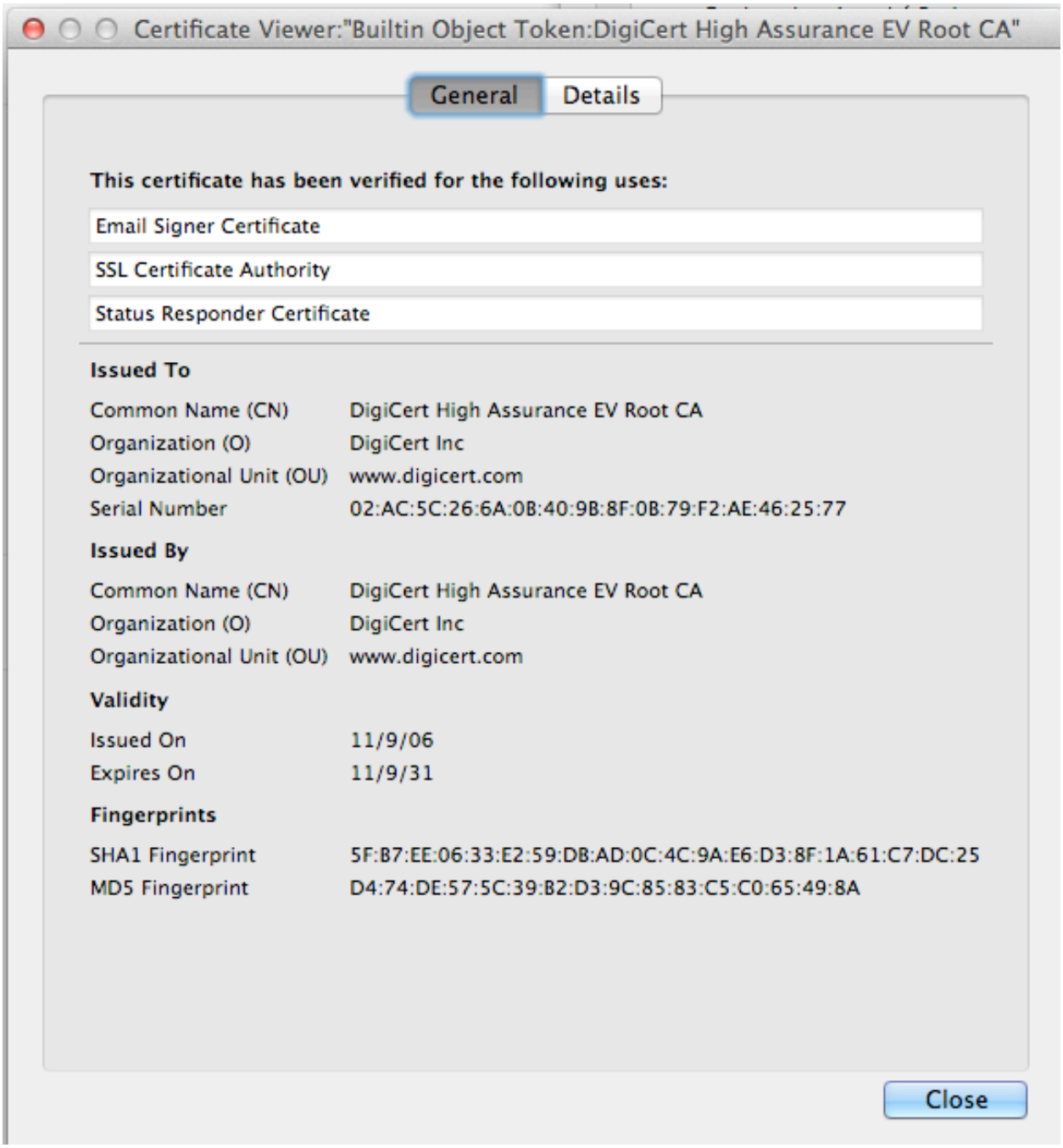
4.9 It contains Alice's name encrypted by the KDC-Bob secret key.

4.10 It has a nonce (e.g., time stamp) encrypted with the session key.

4.11 It contains the session key encrypted by the KDC-Bob secret key.

4.12 Taking the *e*th root mod *n* of a ciphertext block will always reveal the plaintext, no matter what the values of *e* and *n* are. In general this is a very difficult problem, and indeed is the reason why RSA is secure. The point is that, if *e* is too small, then taking the normal integer *e*th root will be the same as taking the *e*th root mod *n*, and taking integer *e*th roots is relatively easy.

4.13 Here is an example of a trusted root CA certificate from Firefox.



4.14 When a symmetric key is used to protect stored information, the recipient usage period may start after the beginning of the originator usage period as shown in the figure. For example, information may be encrypted before being stored on a compact disk. At some later time, the key may be distributed in order to decrypt and recover the information.

4.15 a. A believes that she shares K'_{AB} with B since her nonce came back in message 2 encrypted with a key known only to B (and A). B believes that he shares K'_{AB} with A since N_A was encrypted with K'_{AB}, which could only be retrieved from message 2 by someone who knows K'_{AB} (and this is known only by A and B). A believes that K'_{AB} is fresh since it is included in message 2 together with N_A (and hence message 2 must have been constructed after message 1 was sent). B believes (indeed, knows) that K'_{AB} is fresh since he chose it himself.

b. B. We consider the following interleaved runs of the protocol:

1. A →C(B) : A, N_A
1`. C(B) →A : B, N_A
2`. A →C(B) : $\mathrm{E}(K_{AB}, [N_A, K'_{AB}])$
2. C(B) →A : $\mathrm{E}(K_{AB}, [N_A, K'_{AB}])$
3. A →C(B) : $\mathrm{E}(K'_{AB}, N_A)$

C cannot encrypt A's nonce, so he needs to get help with message 2. He therefore starts a new run with A, letting A do the encryption and reflecting the reply back. A will accept the unprimed protocol run and believe that B is present.

c. To prevent the attack, we need to be more explicit in the messages, e.g. by changing message 2 to include the sender and receiver (in this order), i.e. to be $\mathrm{E}(K_{AB}, [\mathrm{A}, \mathrm{B}, N_A, K'_{AB}])$.

4.16 A typical PKI consists of seven core components. These are briefly described below:

1. Digital certificates (public-key certificates, X.509 certificates): A digital certificate is a signed data structure that binds one or more attributes of an entity with its corresponding public key. By being signed by a recognized and trusted authority (i.e. the Certification Authority) a digital certificate provides assurances that a particular public key belongs to a specific entity (and that the entity possesses the corresponding private key).

2. Certification Authority (CA): Certification Authorities are the people, processes and tools that are responsible for the creation, issue and management of public-key certificates that are used within a PKI.

3. Registration Authority (RA): Registration Authorities are the people, processes and tools that are responsible for authenticating the identity of new entities (users or computing devices) that require certificates from CAs. RAs additionally maintain local registration data and initiate renewal or revocation processes for old or redundant certificates. They

act as agents of CAs (and in that regard can carry out some of the functions of a CA if required).
4. Certificate repository: A database, or other store, which is accessible to all users of a PKI, within which public-key certificates, certificate revocation information and policy information can be held.
5. PKI client software: Client-side software is required to ensure PKI-entities are able to make use of the key and digital certificate management services of a PKI (e.g. key creation, automatic key update and refreshment).
6. PKI-enabled applications: Software applications must be PKI-enabled before they can be used within a PKI. Typically this involves modifying an application so that it can understand and make use of digital certificates (e.g. to authenticate a remote user and authenticate itself to a remote user).
7. Policy (Certificate Policy and Certification Practice Statement): Certificate Policies and Certification Practice Statements are policy documents that define the procedures and practices to be employed in the use, administration and management of certificates within a PKI.

4.17 The primary weakness of symmetric encryption algorithms is keeping the single key secure. Known as key management, it poses a number of significant challenges. If a user wants to send an encrypted message to another using symmetric encryption, he must be sure that she has the key to decrypt the message. How should the first user get the key to the second user? He would not want to send it electronically through the Internet, because that would make it vulnerable to eavesdroppers. Nor can he encrypt the key and send it, because the recipient would need some way to decrypt the key. And if he can even get the get securely to the user, how can be he certain that an attacker has not seen the key on that person's computer? Key management is a significant impediment to using symmetric encryption.

4.18 **a.** A requests a session key for use between A and B from the KDC. A nonce is used for challenge-response.
b. If someone manages to get an old K_s, they can replay the message from step 3 to B and communicate with B, pretending to be A.
c. Timestamps included with the message can counter this vulnerability

4. 19 Adding EMK_0 would allow users to generate personal session keys, which could be exchanged, avoiding the necessity of storing a key variable in a user-to-user session.

4.20 Host *i* has master key KMH_i, with variants $KMH_{i,j}$, j = 0, 1, 2.

$KMH_{i,0}$: used to encrypt session key *KS*
$KMH_{i,1}$: used to encrypt user master keys (at Host *i*)
$KMH_{i,2}$: used to encrypt cross domain key KMH(*i*, *j*) = KMH(*j*, *i*)
(Host *i* to Host *j*)

Host *i* stores $E[KMH_{i,2}, KMH(i, j)]$ and uses a translation instruction RFMK':

$$RFMK'[E[KMH_{i,2}, KMH(i, j)], E(KMH_{i,0}, KS)] \rightarrow E(KMH_{i,j}, K)]$$

A second translation function RTMK (at Host *j*)

$$RTMK[E[KMH_{j,2}, KMH(j, i)], E(KMH(i, j), KS)] \rightarrow E(KMH_{j,0}, KS)]$$

which may be deciphered by a user at Host *j*.

4.21 One solution is to add an instruction similar to RFMK of the form

$$KEYGEN[RN, KMT_i, KMT_j]$$

which will interpret RN as E(KMH0, KS) and return both $E(KMH_i, KS)$ and $E(KMH_j, KS)$, which are sent to the terminals *i* and *j*, respectively. RN need not be maintained at the host.

CHAPTER 5 NETWORK ACCESS CONTROL AND CLOUD SECURITY

ANSWERS TO QUESTIONS

5.1 Network access control (NAC) is an umbrella term for managing access to a network. NAC authenticates users logging into the network and determines what data they can access and actions they can perform. NAC also examines the health of the user's computer or mobile device (the endpoints).

5.2 The Extensible Authentication Protocol (EAP) acts as a framework for network access and authentication protocols. EAP provides a set of protocol messages that can encapsulate various authentication methods to be used between a client and an authentication server. EAP can operate over a variety of network and link level facilities, including point-to-point links, LANs, and other networks, and can accommodate the authentication needs of the various links and networks.

5.3 **EAP-TLS (EAP-Transport Layer Security):** EAP-TLS (RFC 5216) defines how the TLS protocol (described in Chapter 17) can be encapsulated in EAP messages. **EAP-TTLS (EAP-Tunneled TLS)** is similar to EAP-TLS except only the server has a certificate to authenticate itself to the client first. **EAP-GPSK (EAP Generalized Pre-Shared Key)** is an EAP method for mutual authentication and session key derivation using a Pre-Shared Key (PSK). EAP-GPSK specifies an EAP method based on pre-shared keys and employs secret key-based cryptographic algorithms. **EAP-IKEv2** supports mutual authentication and session key establishment using a variety of methods.

5.4 EAPOL (EAP over LAN) operates at the network layers and makes use of an IEEE 802 LAN, such as Ethernet or Wi-Fi, at the link level. EAPOL enables a supplicant to communicate with an authenticator and supports the exchange of EAP packets for authentication.

5.5 IEEE 802.1X, Port-Based Network Access Control was designed to provide access control functions for LANs.

5.6 NIST defines cloud computing as follows: A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models.

5.7 **Software as a service (SaaS):** The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a Web browser. Instead of obtaining desktop and server licenses for software products it uses, an enterprise obtains the same functions from the cloud service. **Platform as a service (PaaS):** The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages and tools supported by the provider. PaaS often provides middleware-style services such as database and component services for use by applications. In effect, PaaS is an operating system in the cloud. **Infrastructure as a service (IaaS):** The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. IaaS enables customers to combine basic computing services, such as number crunching and data storage, to build highly adaptable computer systems.

5.8 The NIST cloud computing reference architecture focuses on the requirements of "what" cloud services provide, not a "how to" design solution and implementation. The reference architecture is intended to facilitate the understanding of the operational intricacies in cloud computing. It does not represent the system architecture of a specific cloud computing system; instead it is a tool for describing, discussing, and developing a system-specific architecture using a common framework of reference.

5.9 **Abuse and nefarious use of cloud computing:** For many cloud providers (CPs), it is relatively easy to register and begin using cloud services, some even offering free limited trial periods. This enables attackers to get inside the cloud to conduct various attacks, such as spamming, malicious code attacks, and denial of service. **Insecure interfaces and APIs:** CPs expose a set of software interfaces or APIs that customers use to manage and interact with cloud services. The

security and availability of general cloud services is dependent upon the security of these basic APIs. **Malicious insiders:** Under the cloud computing paradigm, an organization relinquishes direct control over many aspects of security and, in doing so, confers an unprecedented level of trust onto the CP. One grave concern is the risk of malicious insider activity. Cloud architectures necessitate certain roles that are extremely high-risk. Examples include CP system administrators and managed security service providers. **Shared technology issues:** IaaS vendors deliver their services in a scalable way by sharing infrastructure. Often, the underlying components that make up this infrastructure (CPU caches, GPUs, etc.) were not designed to offer strong isolation properties for a multi-tenant architecture. **Data loss or leakage:** For many clients, the most devastating impact from a security breach is the loss or leakage of data. **Account or service hijacking:** With stolen credentials, attackers can often access critical areas of deployed cloud computing services, allowing them to compromise the confidentiality, integrity, and availability of those services. **Unknown risk profile:** In using cloud infrastructures, the client necessarily cedes control to the cloud provider on a number of issues that may affect security.

ANSWERS TO PROBLEMS

5.2 **Data link layer:** responsible for transmitting and receiving EAP frames between the peer and authenticator. **EAP layer:** receives and transmits EAP packets via the lower layer, implements duplicate detection and retransmission, and delivers and receives EAP messages to and from the EAP peer and authenticator layers. **EAP peer/authenticator layer:** EAP peer and authenticator layers demultiplex incoming EAP packets according to their Type, and deliver them to the EAP method corresponding to that Type. **EAP method layer:** EAP methods implement the authentication algorithms and receive and transmit EAP messages via the EAP peer and authenticator layers. Since fragmentation support is not provided by EAP itself, this is the responsibility of EAP methods.

CHAPTER 6 TRANSPORT-LEVEL SECURITY

ANSWERS TO QUESTIONS

6.1 The advantage of using **IPSec** (Figure 6.1a) is that it is transparent to end users and applications and provides a general-purpose solution. Further, IPSec includes a filtering capability so that only selected traffic need incur the overhead of IPSec processing. The advantage of using **SSL** is that it makes use of the reliability and flow control mechanisms of TCP. The advantage of **application-specific security services** (Figure 6.1c) is that the service can be tailored to the specific needs of a given application.

6.2 SSL handshake protocol; SSL change cipher spec protocol; SSL alert protocol; SSL record protocol.

6.3 **Connection:** A connection is a transport (in the OSI layering model definition) that provides a suitable type of service. For SSL, such connections are peer-to-peer relationships. The connections are transient. Every connection is associated with one session. **Session:** An SSL session is an association between a client and a server. Sessions are created by the Handshake Protocol. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection.

6.4 **Session identifier:** An arbitrary byte sequence chosen by the server to identify an active or resumable session state. **Peer certificate:** An X509.v3 certificate of the peer. **Compression method:** The algorithm used to compress data prior to encryption. **Cipher spec:** Specifies the bulk data encryption algorithm (such as null, DES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation. It also defines cryptographic attributes such as the hash_size. **Master secret:** 48-byte secret shared between the client and server. **Is resumable:** A flag indicating whether the session can be used to initiate new connections.

6.5 **Server and client random:** Byte sequences that are chosen by the server and client for each connection. **Server write MAC secret:** The secret key used in MAC operations on data sent by the server. **Client**

write MAC secret: The secret key used in MAC operations on data sent by the client. **Server write key:** The conventional encryption key for data encrypted by the server and decrypted by the client. **Client write key:** The conventional encryption key for data encrypted by the client and decrypted by the server. **Initialization vectors:** When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter the final ciphertext block from each record is preserved for use as the IV with the following record. **Sequence numbers:** Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed $2^{64} - 1$.

6.6 **Confidentiality:** The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads. **Message Integrity:** The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC).

6.7 Fragmentation; compression; add MAC; encrypt; append SSL record header.

6.8 HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server.

6.9 The initial version, SSH1 was focused on providing a secure remote logon facility to replace TELNET and other remote logon schemes that provided no security. SSH also provides a more general client/server capability and can be used for such network functions as file transfer and e-mail.

6.10 **Transport Layer Protocol**: Provides server authentication, data confidentiality, and data integrity with forward secrecy (i.e., if a key is compromised during one session, the knowledge does not affect the security of earlier sessions). The transport layer may optionally provide compression. **User Authentication Protocol**: Authenticates the user to the server. **Connection Protocol:** Multiplexes multiple logical communications channels over a single underlying SSH connection.

ANSWERS TO PROBLEMS

6.1 The change cipher spec protocol exists to signal transitions in ciphering strategies, and can be sent independent of the complete handshake protocol exchange.

6.2 To integrity protect the first set of messages where the cookies and crypto suite information is exchanged. This will prevent a man-in-the-middle attack in step 1 for instance, where someone can suppress the original message and send a weaker set of crypto suites.

6.3

a. **Brute Force Cryptanalytic Attack:** The conventional encryption algorithms use key lengths ranging from 40 to 168 bits.
b. **Known Plaintext Dictionary Attack:** SSL protects against this attack by not really using a 40-bit key, but an effective key of 128 bits. The rest of the key is constructed from data that is disclosed in the Hello messages. As a result the dictionary must be long enough to accommodate 2^{128} entries.
c. **Replay Attack:** This is prevented by the use of nonces.
d. **Man-in-the-Middle Attack:** This is prevented by the use of public-key certificates to authenticate the correspondents.
e. **Password Sniffing:** User data is encrypted.
f. **IP Spoofing:** The spoofer must be in possession of the secret key as well as the forged IP address.
g. **IP Hijacking:** Again, encryption protects against this attack.
h. **SYN Flooding:** SSL provides no protection against this attack.

6.4 SSL relies on an underlying reliable protocol to assure that bytes are not lost or inserted. There was some discussion of reengineering the future TLS protocol to work over datagram protocols such as UDP, however, most people at a recent TLS meeting felt that this was inappropriate layering (from the SSL FAQ).

6.5 This allows for the message to be authenticated before attempting decryption, which may be more efficient.

CHAPTER 7 WIRELESS NETWORK SECURITY

ANSWERS TO QUESTIONS

7.1 Basic service set.

7.2 Two or more basic service sets interconnected by a distribution system.

7.3 **Association:** Establishes an initial association between a station and an AP. **Authentication:** Used to establish the identity of stations to each other. **Deauthentication:** This service is invoked whenever an existing authentication is to be terminated. **Disassociation:** A notification from either a station or an AP that an existing association is terminated. A station should give this notification before leaving an ESS or shutting down. **Distribution:** used by stations to exchange MAC frames when the frame must traverse the DS to get from a station in one BSS to a station in another BSS. **Integration:** enables transfer of data between a station on an IEEE 802.11 LAN and a station on an integrated IEEE 802.x LAN. **MSDU delivery:** delivery of MAC service data units. **Privacy:** Used to prevent the contents of messages from being read by other than the intended recipient. **Reassocation:** Enables an established association to be transferred from one AP to another, allowing a mobile station to move from one BSS to another.

7.4 It may or may not be.

7.5 **Mobility** refers to the types of physical transitions that can be made by a mobile node within an 802.11 environment (no transition, movement from one BSS to another within an ESS, movement from one ESS to another). **Association** is a service that allows a mobile node that has made a transition to identify itself to the AP within a BSS so that the node can participate in data exchanges with other mobile nodes.

7.6 IEEE 802.11i addresses three main security areas: authentication, key management, and data transfer privacy.

7.7 **Discovery:** An AP uses messages called Beacons and Probe Responses to advertise its IEEE 802.11i security policy. The STA uses these to identify an AP for a WLAN with which it wishes to communicate. The STA associates with the AP, which it uses to select the cipher suite and authentication mechanism when the Beacons and Probe Responses present a choice. **Authentication**: During this phase, the STA and AS prove their identities to each other. The AP blocks non-authentication traffic between the STA and AS until the authentication transaction is successful. The AP does not participate in the authentication transaction other than forwarding traffic between the STA and AS. **Key generation and distribution**: The AP and the STA perform several operations that cause cryptographic keys to be generated and placed on the AP and the STA. Frames are exchanged between the AP and STA only **Protected data transfer**: Frames are exchanged between the STA and the end station through the AP. As denoted by the shading and the encryption module icon, secure data transfer occurs between the STA and the AP only; security is not provided end-to-end. **Connection termination**: The AP and STA exchange frames. During this phase, the secure connection is torn down and the connection is restored to the original state.

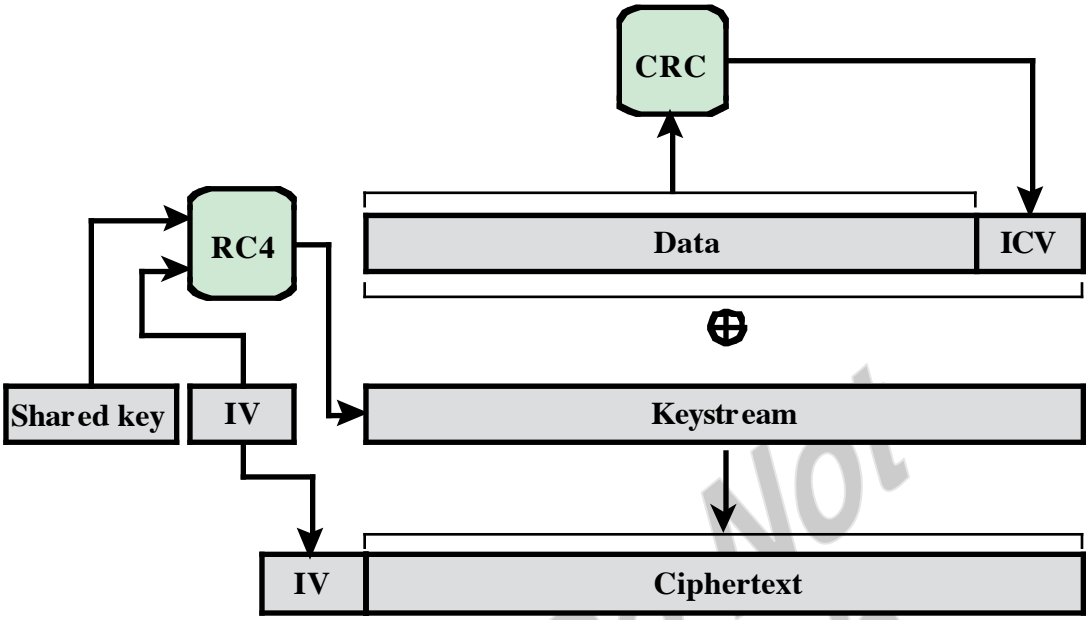
7.8 TKIP is designed to require only software changes to devices that are implemented with the older wireless LAN security approach called Wired Equivalent Privacy (WEP).

ANSWERS TO PROBLEMS

7.1 **a.** This scheme is extremely simple and easy to implement. It does protect against very simple attacks using an off-the-shelf Wi-Fi LAN card, and against accidental connection to the wrong network.
b. This scheme depends on all parties behaving honestly. The scheme does not protect against MAC address forgery.

7.2 **a.** Because the AP remembers the random number previously sent, it can check whether the result sent back was encrypted with the correct key; the STA must know the key in order to encrypt the random value successfully.
b. This scheme does nothing to prove to the STA that the AP knows the key, so authentication is only one way.
c. If an attacker is eavesdropping, this scheme provides the attacker with a plaintext-ciphertext pair to use in cryptanalysis.

7.3 a.

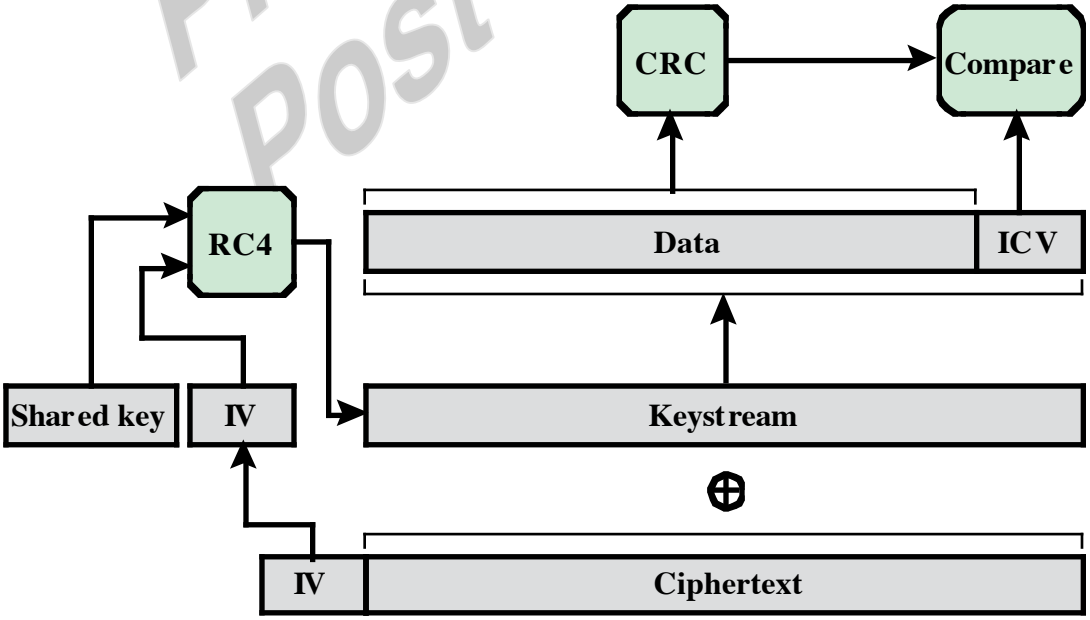


b. 1. The IV value, which is received in plaintext, is concatenated with the WEP key shared by transmitter and receiver to form the seed, or key input, to RC4.

2. The ciphertext portion of the received MPDU is decrypted using RC4 to recover the Data block and the ICV.

3. The ICV is computed over the plaintext received Data block and compared to the received plaintext ICV to authenticate the Data block.

c.



7.4 Because WEP works by XORing the data to get the ciphertext, bit flipping survives the encryption process. Flipping a bit in the plaintext always flips the same bit in the ciphertext and vice versa.

CHAPTER 8 ELECTRONIC MAIL SECURITY

ANSWERS TO QUESTIONS

8.1 Authentication, confidentiality, compression, e-mail compatibility, and segmentation

8.2 A detached signature is useful in several contexts. A user may wish to maintain a separate signature log of all messages sent or received. A detached signature of an executable program can detect subsequent virus infection. Finally, detached signatures can be used when more than one party must sign a document, such as a legal contract. Each person's signature is independent and therefore is applied only to the document. Otherwise, signatures would have to be nested, with the second signer signing both the document and the first signature, and so on.

8.3 **a.** It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification. If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress the message when verification is required. **b.** Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty. The algorithm is not deterministic; various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio and, as a result, produce different compressed forms. However, these different compression algorithms are interoperable because any version of the algorithm can correctly decompress the output of any other version. Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm.

8.4 R64 converts a raw 8-bit binary stream to a stream of printable ASCII characters. Each group of three octets of binary data is mapped into four ASCII characters.

8.5 When PGP is used, at least part of the block to be transmitted is encrypted. If only the signature service is used, then the message

digest is encrypted (with the sender's private key). If the confidentiality service is used, the message plus signature (if present) are encrypted (with a one-time symmetric key). Thus, part or all of the resulting block consists of a stream of arbitrary 8-bit octets. However, many electronic mail systems only permit the use of blocks consisting of ASCII text.

8.6 PGP includes a facility for assigning a level of trust to individual signers and to keys.

8.7 RFC 5322 defines a format for text messages that are sent using electronic mail.

8.8 MIME is an extension to the RFC 822 framework that is intended to address some of the problems and limitations of the use of SMTP (Simple Mail Transfer Protocol) or some other mail transfer protocol and RFC 822 for electronic mail.

8.9 S/MIME (Secure/Multipurpose Internet Mail Extension) is a security enhancement to the MIME Internet e-mail format standard, based on technology from RSA Data Security.

8.10 DomainKeys Identified Mail (DKIM) is a specification for cryptographically signing e-mail messages, permitting a signing domain to claim responsibility for a message in the mail stream.

ANSWERS TO PROBLEMS

8.1 CFB avoids the need to add and strip padding.

8.2 This is just another form of the birthday paradox discussed in Appendix 11A. Let us state the problem as one of determining what number of session keys must be generated so that the probability of a duplicate is greater than 0.5. From Equation (11.6) in Appendix 11A, we have the approximation:

$$k = 1.18 \times \sqrt{n}$$

For a 128-bit key, there are 2^{128} possible keys. Therefore

$$k = 1.18 \times \sqrt{2^{128}} = 1.18 \times 2^{64}$$

8.3 Again, we are dealing with a birthday-paradox phenomenon. We need to calculate the value for:

P(n, k) = Pr [at least one duplicate in k items, with each item able to take on one of n equally likely values between 1 and n]

In this case, k = N and n = 2^{64}. Using equation (11.5) of Appendix 11A:

$$P\left(2^{64}, N\right) = 1 - \frac{2^{64}!}{\left(2^{64} - N\right)! 2^{64 \times k}}$$

$$> 1 - e^{-[N \times (N-1)]/2^{65}}$$

8.4 **a.** Not at all. The message digest is encrypted with the sender's private key. Therefore, anyone in possession of the public key can decrypt it and recover the entire message digest.

b. The probability that a message digest decrypted with the wrong key would have an exact match in the first 16 bits with the original message digest is 2^{-16}.

8.5 We trust this owner, but that does not necessarily mean that we can trust that we are in possession of that owner's public key.

8.6 In X.509 there is a hierarchy of Certificate Authorities. Another difference is that in X.509 users will only trust Certificate Authorities while in PGP users can trust other users.

8.7 DES is unsuitable because of its short key size. Two-key triple DES, which has a key length of 112 bits, is suitable. AES is also suitable.

8.8 It certainly provides more security than a monoalphabetic substitution. Because we are treating the plaintext as a string of bits and encrypting 6 bits at a time, we are not encrypting individual characters. Therefore, the frequency information is lost, or at least significantly obscured.

8.9 **a.** The first step is to convert the characters into 8-bit ASCII with zero parity. Consulting the table in Appendix Q, we have the following correspondence:

p 01110000
l 01101100
a 01100001
i 01101001
n 01101110
t 01110100
e 01100101
x 01111000
t 01110100

Next, we block these off into groups of 6 bits, show the 6-bit decimal value, and do the encoding.

011100 000110 110001 100001 011010 010110 111001 110100
28 6 49 33 26 22 57 52
c G x h a W 5 0
011001 010111 100001 110100
25 23 33 52
Z X h 0

So the radix-64 encoding is cGxhaW50ZXh0

b. All of the characters are "safe", so the quoted-printable encoding is simply plaintext

CHAPTER 9 IP SECURITY

ANSWERS TO QUESTIONS

9.1 **Secure branch office connectivity over the Internet:** A company can build a secure virtual private network over the Internet or over a public WAN. This enables a business to rely heavily on the Internet and reduce its need for private networks, saving costs and network management overhead. **Secure remote access over the Internet:** An end user whose system is equipped with IP security protocols can make a local call to an Internet service provider (ISP) and gain secure access to a company network. This reduces the cost of toll charges for traveling employees and telecommuters. **Establishing extranet and intranet connectivity with partners:** IPSec can be used to secure communication with other organizations, ensuring authentication and confidentiality and providing a key exchange mechanism. **Enhancing electronic commerce security:** Even though some Web and electronic commerce applications have built-in security protocols, the use of IPSec enhances that security.

9.2 Access control; connectionless integrity; data origin authentication; rejection of replayed packets (a form of partial sequence integrity); confidentiality (encryption); and limited traffic flow confidentiality

9.3 A security association is uniquely identified by three parameters: **Security Parameters Index (SPI):** A bit string assigned to this SA and having local significance only. The SPI is carried in AH and ESP headers to enable the receiving system to select the SA under which a received packet will be processed. **IP Destination Address:** Currently, only unicast addresses are allowed; this is the address of the destination endpoint of the SA, which may be an end user system or a network system such as a firewall or router. **Security Protocol Identifier:** This indicates whether the association is an AH or ESP security association.

A security association is normally defined by the following parameters:

Sequence Number Counter: A 32-bit value used to generate the Sequence Number field in AH or ESP headers, described in Section 9.3 (required for all implementations). **Sequence Counter Overflow:** A flag indicating whether overflow of the Sequence Number Counter

should generate an auditable event and prevent further transmission of packets on this SA (required for all implementations). **Anti-Replay Window:** Used to determine whether an inbound AH or ESP packet is a replay, described in Section 9.3 (required for all implementations). **AH Information:** Authentication algorithm, keys, key lifetimes, and related parameters being used with AH (required for AH implementations). **ESP Information:** Encryption and authentication algorithm, keys, initialization values, key lifetimes, and related parameters being used with ESP (required for ESP implementations). **Lifetime of this Security Association:** A time interval or byte count after which an SA must be replaced with a new SA (and new SPI) or terminated, plus an indication of which of these actions should occur (required for all implementations). **IPSec Protocol Mode:** Tunnel, transport, or wildcard (required for all implementations). These modes are discussed later in this section. **Path MTU:** Any observed path maximum transmission unit (maximum size of a packet that can be transmitted without fragmentation) and aging variables (required for all implementations).

9.4 **Transport mode** provides protection primarily for upper-layer protocols. That is, transport mode protection extends to the payload of an IP packet. **Tunnel mode** provides protection to the entire IP packet.

9.5 A replay attack is one in which an attacker obtains a copy of an authenticated packet and later transmits it to the intended destination. The receipt of duplicate, authenticated IP packets may disrupt service in some way or may have some other undesired consequence.

9.6 **1.** If an encryption algorithm requires the plaintext to be a multiple of some number of bytes (e.g., the multiple of a single block for a block cipher), the Padding field is used to expand the plaintext (consisting of the Payload Data, Padding, Pad Length, and Next Header fields) to the required length. **2.** The ESP format requires that the Pad Length and Next Header fields be right aligned within a 32-bit word. Equivalently, the ciphertext must be an integer multiple of 32 bits. The Padding field is used to assure this alignment. **3.** Additional padding may be added to provide partial traffic flow confidentiality by concealing the actual length of the payload.

9.7 **Transport adjacency:** Refers to applying more than one security protocol to the same IP packet, without invoking tunneling. This approach to combining AH and ESP allows for only one level of combination; further nesting yields no added benefit since the processing is performed at one IPSec instance: the (ultimate) destination. **Iterated tunneling:** Refers to the application of multiple

layers of security protocols effected through IP tunneling. This approach allows for multiple levels of nesting, since each tunnel can originate or terminate at a different IPSec site along the path.

ANSWERS TO PROBLEMS

9.1

row 1: Traffic between this host and any other host, both using port 500, and using UDP, bypasses IPsec. This is used for IKE traffic.

row 2: ICMP message to or from any remote address are error messages, and bypass IPsec.

row 3: Traffic between 1.2.3.101 and 1.2.3.0/24 is intranet traffic and must be protected by ESP, with the exception of traffic defined in earlier rows.

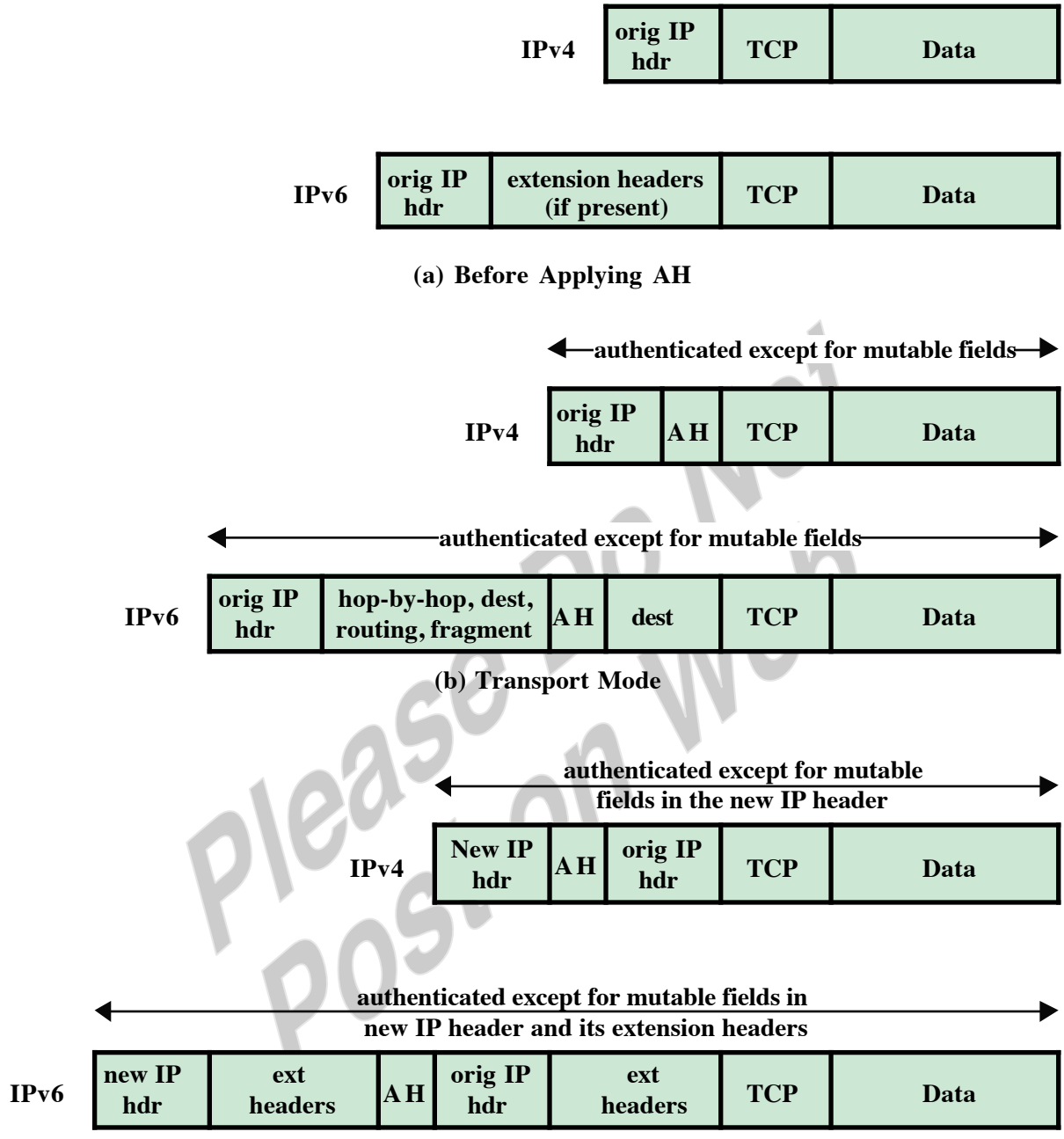
row 4: TCP traffic between this host (1.2.3.101) and the server (1.2.4.10) on server port 80 is ESP protected.

row 5: TCP traffic between this host (1.2.3.101) and the server (1.2.4.10) on server port 80 is protected by TLS and so can bypass IPsec.

row 6: Any other traffic between 1.2.3.101 and 1.2.3.0/24 is prohibited and is discarded.

row 7: Any other traffic between 1.2.3.101 goes to the Internet and bypasses IPsec.

9.2.



9.3 **AH** provides access control, connectionless integrity, data origin authentication, and rejection of replayed packets. **ESP** provides all of these plus confidentiality and limited traffic flow confidentiality.

9.4 **a. Immutable**: Version, Internet Header Length, Total Length, Identification, Protocol (This should be the value for AH.), Source Address, Destination Address (without loose or strict source routing). None of these are changed by routers in transit.

Mutable but predictable: Destination Address (with loose or strict source routing). At each intermediate router designated in the source routing list, the Destination Address field is changed to indicate the next designated address. However, the source routing field contains the information needed for doing the MAC calculation.
Mutable (zeroed prior to ICV calculation): Type of Service (TOS), Flags, Fragment Offset, Time to Live (TTL), Header Checksum. TOS may be altered by a router to reflect a reduced service. Flags and Fragment offset are altered if an router performs fragmentation. TTL is decreased at each router. The Header Checksum changes if any of these other fields change.

b. **Immutable**: Version, Payload Length, Next Header (This should be the value for AH.), Source Address, Destination Address (without Routing Extension Header)
Mutable but predictable: Destination Address (with Routing Extension Header)
Mutable (zeroed prior to ICV calculation): Class, Flow Label, Hop Limit

c. IPv6 options in the Hop-by-Hop and Destination Extension Headers contain a bit that indicates whether the option might change (unpredictably) during transit.
Mutable but predictable: Routing
Not Applicable: Fragmentation occurs after outbound IPSec processing and reassembly occur before inbound IPSec processing , so the Fragmentation Extension Header, if it exists, is not seen by IPSec.

9.5 **a.** The received packet is to the left of the window, so the packet is discarded; this is an auditable event. No change is made to window parameters.

b. The received packet falls within the window. If it is new, the MAC is checked. If the packet is authenticated, the corresponding slot in the window is marked. If it is not new, the packet is discarded. In either case, no change is made to window parameters.

c. The received packet is to the right of the window and is new, so the MAC is checked. If the packet is authenticated, the window is advanced so that this sequence number is the right edge of the window, and the corresponding slot in the window is marked. In this case, the window now spans from 120 to 540.

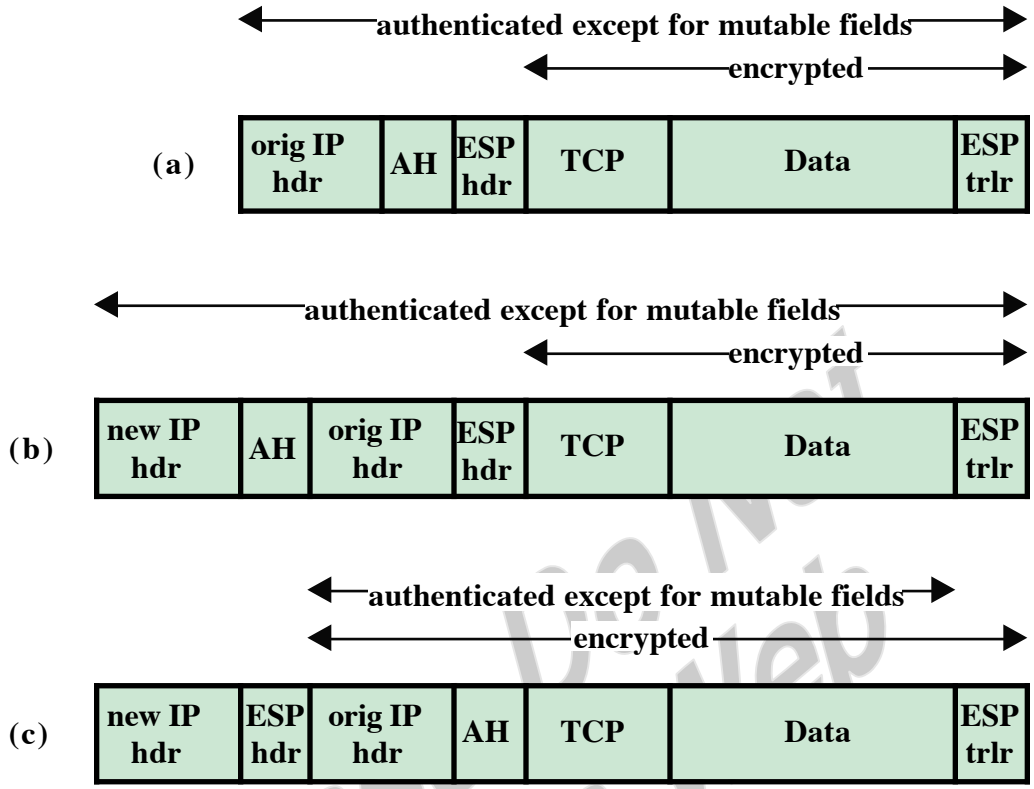
9.6 From RFC 2401

IPv4 Header Fields	Outer Header at Encapsulator	Inner Header at Decapsulator
version	4 (1)	no change
header length	constructed	no change
TOS	copied from inner header (5)	no change
total length	constructed	no change
ID	constructed	no change
Flags	constructed, DF (4)	no change
Fragment offset	constructed	no change
TTL	constructed	decrement (2)
protocol	AH, ESP, routing header	no change
checksum	constructed	no change
source address	constructed (3)	no change
destination address	constructed (3)	no change
options	never copied	no change

IPv6 Header Fields	Outer Header at Encapsulator	Inner Header at Decapsulator
version	6 (1)	no change
class	copied or configured (6)	no change
flow id	copied or configured	no change
length	constructed	no change
next header	AH, ESP, routing header	no change
hop count	constructed (2)	decrement (2)
source address	constructed (3)	no change
dest address	constructed (3)	no change
extension headers	never copied	no change

1. The IP version in the encapsulating header can be different from the value in the inner header.
2. The TTL in the inner header is decremented by the encapsulator prior to forwarding and by the decapsulator if it forwards the packet.
3. src and dest addresses depend on the SA, which is used to determine the dest address, which in turn determines which src address (net interface) is used to forward the packet.
4. configuration determines whether to copy from the inner header (IPv4 only), clear or set the DF.
5. If Inner Hdr is IPv4, copy the TOS. If Inner Hdr is IPv6, map the Class to TOS.
6. If Inner Hdr is IPv6, copy the Class. If Inner Hdr IPv4, map the TOS to Class.

9.7 We show the results for IPv4; IPv6 is similar.



9.8 This order of processing facilitates rapid detection and rejection of replayed or bogus packets by the receiver, prior to decrypting the packet, hence potentially reducing the impact of denial of service attacks. It also allows for the possibility of parallel processing of packets at the receiver, i.e., decryption can take place in parallel with authentication.

9.9 The Initial Exchanges and the CREATE_CHILD_SA Exchange

9.10 It is an addition to the IP layer.

CHAPTER 10 MALICIOUS SOFTWARE

ANSWERS TO QUESTIONS

10.1 The three broad mechanisms malware can use to propagate are: infection of existing executable or interpreted content by viruses that is subsequently spread to other systems; exploit of software vulnerabilities either locally or over a network by worms or drive-by-downloads to allow the malware to replicate; and social engineering attacks that convince users to bypass security mechanisms to install trojans, or to respond to phishing attacks.

10.2 Four broad categories of payloads that malware may carry are: corruption of system or data files; theft of service in order to make the system a zombie agent of attack as part of a botnet; theft of information from the system, especially of logins, passwords or other personal details by keylogging or spyware programs; and stealthing where the malware hides it presence on the system from attempts to detect and block it.

10.3 The typical phases of operation of a virus or worm are: a dormant phase (when the virus is idle), a propagation phase (where it makes copies of itself elsewhere), a triggering phase (when activated), and an execution phase (to perform some target function).

10.4 Some mechanisms a virus can use to conceal itself include: encryption, stealth, polymorphism, metamorphism.

10.5 Machine executable viruses infect executable program files to carry out their work in a manner that is specific to a particular operating system and, in some cases, specific to a particular hardware platform. Macro viruses infect files with macro or scripting code that is used to support active content in a variety of user document types, and is interpreted by an application.

10.6 A worm may access remote systems to propagate using: an electronic mail or instant messenger facility, file sharing, remote execution capability, remote file access or transfer capability, or a remote login capability.

10.7 A "drive-by-download" exploits browser vulnerabilities so that when the user views a web page controlled by the attacker, it contains code that exploits some browser bug to download and install malware on the system without the user's knowledge or consent. It differs from a worm since it does not actively propagate as a worm does, but rather waits for unsuspecting users to visit the malicious web page in order to spread to their systems.

10.8 A **logic bomb** is code embedded in the malware that is set to "explode" when certain conditions are met, such as the presence or absence of certain files or devices on the system, a particular day of the week or date, a particular version or configuration of some software, or a particular user running the application. When triggered, the bomb executes some payload carried by the malware.

10.9 A **backdoor** is a secret entry point into a program or system that allows someone who is aware of the backdoor to gain access without going through the usual security access procedures. A **bot** subverts the computational and network resources of the infected system for use by the attacker. A **keylogger** captures keystrokes on the infected machine, to allow an attacker to monitor sensitive information including login and password credentials. **Spyware** subverts the compromised machine to allow monitoring of a wide range of activity on the system, including monitoring the history and content of browsing activity, redirecting certain web page requests to fake sites controlled by the attacker, dynamically modifying data exchanged between the browser and certain web sites of interest; which can result in significant compromise of the user's personal information. A **rootkit** is a set of programs installed on a system to maintain covert access to that system with administrator (or root) privileges, whilst hiding evidence of its presence to the greatest extent possible. These **can** all be present in the same malware.

10.10 A rootkit may be placed in: user mode where it can intercept calls to APIs and modify results; in kernel mode where it can intercept kernel API calls and hide it presence in kernel tables; in a virtual machine hypervisor where it can then transparently intercept and modify states and events occurring in the virtualized system; or in some other external mode such as BIOS or in BIOS or system management mode, where it can directly access hardware.

10.11 Malware countermeasure elements include **prevention** in not allowing malware to get into the system in the first place, or blocking its ability to modify the system, via policy, awareness, vulnerability mitigation and threat mitigation; **detection** to determine that it has occurred and locate the malware;

identification to identify the specific malware that has infected the system; and **removal** to remove all traces of malware virus from all infected systems so that it cannot spread further.

10.12 Three places malware mitigation mechanisms may be located, are: on the infected system, where some host-based "anti-virus" program is running, monitoring data imported into the system, and the execution and behavior of programs running on the system; as part of the perimeter security mechanisms used in an organizations firewall and intrusion detection systems; or it may use distributed mechanisms that gather data from both host-based and perimeter sensors, potentially over a large number of networks and organizations, in order to obtain the largest scale view of the movement of malware.

10.13 The four generations of anti-virus software are:
First generation: simple scanners that require a malware signature to identify it
Second generation: heuristic scanners use heuristic rules to search for probable malware instances, or uses integrity checking to identify changed files
Third generation: activity traps that identify malware by its actions rather than its structure in an infected program
Fourth generation: full-featured protection uses packages of a variety of anti-virus techniques used in conjunction, including scanning and activity trap components.

10.14 A denial of service (DoS) attack is an attempt to prevent legitimate users of a service from using that service. When this attack comes from a single host or network node, then it is simply referred to as a DoS attack. A more serious threat is posed by a DDoS attack. In a DDoS attack, an attacker is able to recruit a number of hosts throughout the Internet to simultaneously or in a coordinated fashion launch an attack upon the target.

ANSWERS TO PROBLEMS

10.1 The program will loop indefinitely once all of the executable files in the system are infected.

10.2 D is supposed to examine a program P and return TRUE if P is a computer virus and FALSE if it is not. But CV calls D. If D says that CV is a virus, then CV will not infect an executable. But if D says that CV is not a virus, it infects an executable. D always returns the wrong answer.

10.3 The original code has been altered to disrupt the signature without affecting the semantics of the code. The ineffective instructions in the metamorphic code are the second, third, fifth, sixth, and eighth.

10.4 **a.** The following is from Spafford, E. " The Internet Worm Program: An Analysis." Purdue Technical Report CSD-TR-823.
Common choices for passwords usually include fantasy characters, but this list contains none of the likely choices (e.g., "hobbit", "dwarf", "gandalf", "skywalker", "conan"). Names of relatives and friends are often used, and we see women's names like "jessica", "caroline", and "edwina", but no instance of the common names "jennifer" or "kathy". Further, there are almost no men's names such as "thomas" or either of "stephen" or "steven" (or "eugene"!). Additionally, none of these have the initial letters capitalized, although that is often how they are used in passwords. Also of interest, there are no obscene words in this dictionary, yet many reports of concerted password cracking experiments have revealed that there are a significant number of users who use such words (or phrases) as passwords. The list contains at least one incorrect spelling: "commrades" instead of "comrades"; I also believe that "markus" is a misspelling of "marcus". Some of the words do not appear in standard dictionaries and are non-English names: "jixian", "vasant", "puneet", etc. There are also some unusual words in this list that I would not expect to be considered common: "anthropogenic", "imbroglio", "umesh", "rochester", "fungible", "cerulean", etc.

b. Again, from Spafford:
I imagine that this list was derived from some data gathering with a limited set of passwords, probably in some known (to the author) computing environment. That is, some dictionary-based or brute-force attack was used to crack a selection of a few hundred passwords taken from a small set of machines. Other approaches to gathering passwords could also have been used: Ethernet monitors, Trojan Horse login programs, etc. However they may have been cracked, the ones that were broken would then have been added to this dictionary. Interestingly enough, many of these words are not in the standard on-line dictionary (in /usr/dict/words). As such, these words are useful as a supplement to the main dictionary-based attack the worm used as strategy #4, but I would suspect them to be of limited use before that time.

10.5 Logic bomb.

10.6 Backdoor.

10.7 The found USB memory stick may pose a range of threats to the confidentiality, integrity and availability of the work system. Each of the malware propagation mechanisms we discuss could use such a memory stick for transport. It may carry a program infected with an executable virus, or document infected with a macro virus, which if run or opened can allow the virus to run and spread. It could carry a malicious worm that may be run automatically using the autorun capability, or by exploiting some vulnerability when the USB stick is viewed. Or it could contain a trojan horse program or file that would threaten the system if installed or allowed to run. You can mitigate these threats, and try to safely determine the contents of the memory stick, by scanning the memory stick with suitable, up-to-date anti-virus software for any signs of malware – though this will not detect unknown, zero-day exploits. You could examine the memory stick in a controlled environment, such as a live-boot linux or other system, or in some emulation environment, which cannot be changed even if the malware does manage to run.

10.8 Observations of your home PC is responding very slowly, with high levels of network activity, may indicate the presence of malware, likely including bot code, on the system. The slow response and net traffic could be caused by it participating in a botnet, perhaps distributing spam emails, performing DDoS attacks, or other malicious activities. This malware could have gained access to the system as a result of installing some trojan program perhaps advertised in spam email or on a compromised website, from a drive-by-download, or from exploit of some vulnerability on the system by a worm. Possible steps to check whether this has occurred include examining the process/task list for unknown programs executing, looking at logs of network traffic kept by a host firewall program to see which programs are generating traffic, or scanning the system with suitable, up-to-date anti-virus software for any signs of malware – though this will not detect unknown, zero-day exploits. If you do identify malware on your PC, you may be able to restore it to safe operation using suitable, up-to-date anti-virus software, provided the malware is known. Otherwise you may have to erase all storage and rebuild the system from scratch.

10.9 If a user installs some custom codec claimed needed to view some videos, they may actually be installing trojan horse code. It may indeed allow viewing of the video, or may just be an excuse to compromise the system. Such code may pose a range of threats to the confidentiality, integrity and availability of the system. It may include backdoor, bot, keylogger, spyware, rootkit or indeed any other malware payloads.

10.10 If when you download and start to install some game app, you are asked to approve the access permissions "Send SMS messages" and to "Access your address-book", you should indeed be suspicious that a game wants these types of permissions, as it would not seem needed just for a game. Rather it could be malware that wants to collect details of all your contacts, and either return them to the attacker via SMS, or allow the code to send SMS messages to your contacts, perhaps enticing them to also download and install this malware. Such code is a trojan horse, since it contains covert functions as well as the advertised functionality.

10.11 If you should open the PDF attachment, then it could contain malicious scripting code that could run should you indeed select the 'Open' button. This may be either worm (specifically exploiting a client-side vulnerability), or trojan horse code. You could you check your suspicions without threatening your system by using the scroll bar to examine all the code about to be executed should you select the 'Open' button, and see if it looks suspicious. You could also scan the PDF document with suitable, up-to-date anti-virus software for any signs of malware – though this will not detect unknown, zero-day exploits. This type of message is associated with a spear-phishing attack, given that the email was clearly crafted to suit the recipient. That particular e-mail would only have been sent to one or a few people for whom the details would seem plausible.

10.12 This email is attempting a general phishing attack, being sent to very large numbers of people, in the hope that a sufficient number both use the named bank, and are fooled into divulging their sensitive login credentials to the attacker. The most likely mechanism used to distribute this e-mail is via a botnet using large numbers of compromised systems to generate the necessary high volumes of spam emails. You should never ever follow such a link in an email and supply the requested details. You should only ever access sensitive sites by directly entering their known URL into your browser. It may be appropriate to forward a copy of such emails to a relevant contact at the bank if they ask for this. Otherwise it should just be deleted.

10.13 Such a letter strongly suggests that an attacker has collected sufficient personal details about you in order to satisfy the finance company that they are you for the purpose of establishing such a loan. Having taken the money, they have then left you responsible for the repayments. This was most likely done using either a phishing attack, perhaps persuading you to complete and return some form with the needed personal details; or by using spyware installed on your personal computer system by a worm or trojan horse malware,

that then collected the necessary details from files on the system, or by monitoring your access to sensitive sites, such as banking sites.

10.14 One approach is to send out false alerts. This would cause alerted systems to shut down traffic incorrectly. If the spoofed alerts come from an external (to the network) source, the firewall can filter them. Also, authentication schemes can prevent the attack. Alternatively, the attacker can first compromise an internal host and then forge an alert. If an authentication scheme is used, this attack can only succeed if the spoofer has access to keys. This creates a higher hurdle for the attacker. Another approach: if an attacker is aware of the use of PWC, the worm could be designed to try to thwart the timing analysis of the PWC agents. This appears to be very difficult because you have multiple cooperating agents and if the worm is to propagate in a reasonable time, sooner or later, worm propagation attempts must be made.

CHAPTER 11 INTRUDERS

ANSWERS TO QUESTIONS

11.1 **Masquerader:** An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account. **Misfeasor:** A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges. **Clandestine user:** An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.

11.2 **One-way encryption:** The system stores only an encrypted form of the user's password. When the user presents a password, the system encrypts that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible) in which the password is used to generate a key for the encryption function and in which a fixed-length output is produced. **Access control:** Access to the password file is limited to one or a very few accounts.

11.3 **1.** If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised. Even if the detection is not sufficiently timely to preempt the intruder, the sooner that the intrusion is detected, the less the amount of damage and the more quickly that recovery can be achieved. **2.** An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions. **3.** Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.

11.4 **Statistical anomaly detection** involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior. **Rule-Based Detection** involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.

11.5 **Counter:** A nonnegative integer that may be incremented but not decremented until it is reset by management action. Typically, a count of certain event types is kept over a particular period of time. **Gauge:** A nonnegative integer that may be incremented or decremented. Typically, a gauge is used to measure the current value of some entity. **Interval timer:** The length of time between two related events. **Resource utilization:** Quantity of resources consumed during a specified period.

11.6 With **rule-based anomaly detection**, historical audit records are analyzed to identify usage patterns and to generate automatically rules that describe those patterns. Rules may represent past behavior patterns of users, programs, privileges, time slots, terminals, and so on. Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior. **Rule-based penetration identification** uses rules for identifying known penetrations or penetrations that would exploit known weaknesses. Rules can also be defined that identify suspicious behavior, even when the behavior is within the bounds of established patterns of usage. Typically, the rules used in these systems are specific to the machine and operating system. Also, such rules are generated by "experts" rather than by means of an automated analysis of audit records.

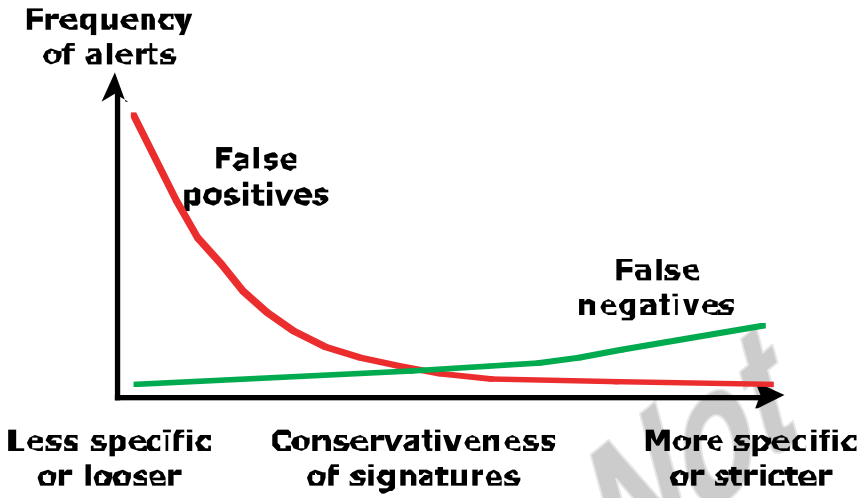
11.7 Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems.

11.8 The salt is combined with the password at the input to the one-way encryption routine.

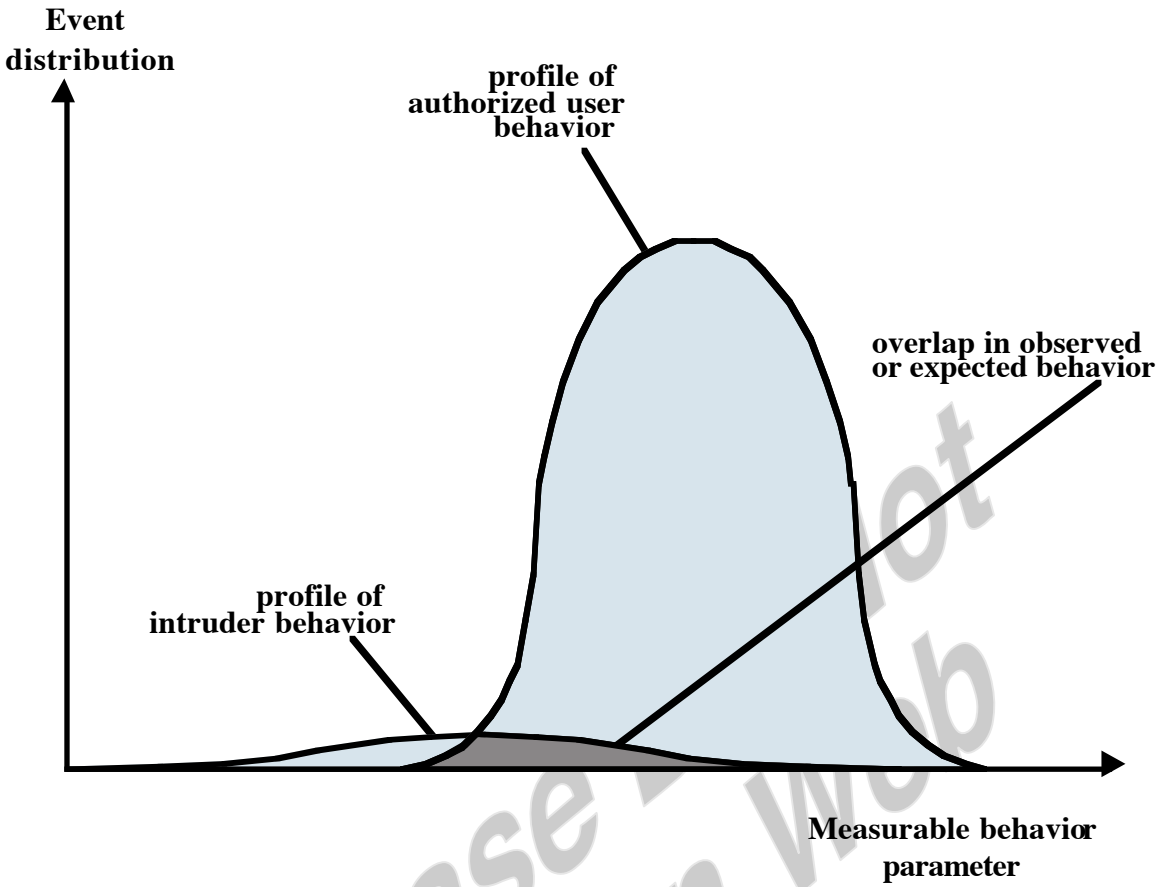
11.9 **User education:** Users can be told the importance of using hard-to-guess passwords and can be provided with guidelines for selecting strong passwords. **Computer-generated passwords:** Users are provided passwords generated by a computer algorithm. **Reactive password checking:** the system periodically runs its own password cracker to find guessable passwords. The system cancels any passwords that are guessed and notifies the user. **Proactive password checking:** a user is allowed to select his or her own password. However, at the time of selection, the system checks to see if the password is allowable and, if not, rejects it.

ANSWERS TO PROBLEMS

11.1 This is a typical example:



11.2 a. The graph below doesn't look like a correct probability distribution and is instead labeled as *event distribution*. The point here is that even if you have nice, mostly non-overlapping probability distributions for distinguishing intruders and authorized users like Figure 11.1, the problem is for most systems we hope the actual numbers of intruders is dwarfed by the number of authorized users. This means that the long tail of the authorized user's distribution that overlaps with the intruder's distribution would generate lots of false positives (relative to the number of real intruders detected) even if it is only a few percent of the authorized users.



b. A randomly selected event that in the overlap region is (roughly) 95% likely to be an authorized user, even though the region covers 50% of the intruder's probability distribution.

11.3 A file integrity checking tool such as tripwire can be very useful in identifying changed files or directories on a system, particularly when those change should not have occurred. However most computer systems are not static, and significant numbers of files do change constantly. Hence it is necessary to configure tripwire with a list of files and directories to monitor, since otherwise reports to the administrator would be filled with lists of files that are changing as a matter of normal operation of the system. It is not too difficult to monitor a small list of critical system programs, daemons and configuration files. Doing this means attempts to alter these files will likely be detected. However the large areas of the system not being monitored means an attacker changing or adding files in these areas will not be detected. The more of the system that is to be monitored, the more care is needed to identify only files not expected to change. Even then, it is likely that user's home areas, and other shared document areas, cannot be monitored, since they are likely to be creating and changing files in there regularly. As well, there needs to be a process to manage the update of monitored files (as a result of installing patches, upgrades, new services, configuration changes etc). This process has

to verify that the changed files are correct, and then update the cryptographic checksums of these files. Lastly the database of cryptographic checksums must be protected from any attempt by an attacker to corrupt it, ideally by locating on read-only media (except when controlled updates are occurring).

11.4 Let WB equal the event {witness reports Blue cab}. Then:

$$\Pr[\text{Blue}/\text{WB}] = \frac{\Pr[\text{WB}/\text{Blue}]\Pr[\text{Blue}]}{\Pr[\text{WB}/\text{Blue}]\Pr[\text{Blue}] + \Pr[\text{WB}/\text{Green}]\Pr[\text{Green}]}$$
$$= \frac{(0.8)(0.15)}{(0.8)(0.15)+(0.2)(0.85)} = 0.41$$

This example, or something similar, is referred to as "the juror's fallacy."

11.5 **a.** If this is a license plate number, that is easily guessable.
b. suitable
c. easily guessable
d. easily guessable
e. easily guessable
f. suitable
g. very unsuitable
h. This is bigbird in reverse; not suitable.

11.6 The number of possible character strings of length 8 using a 36-character alphabet is $36^8 \approx 2^{41}$. However, only 2^{15} of them need be looked at, because that is the number of possible outputs of the random number generator. This scheme is discussed in [MORR79].

11.7 **a.** T = $\frac{26^4}{2}$ seconds = 63.5 hours
b. Expect 13 tries for each digit. T = 13 × 4 = 52 seconds.

11.8 **a.** $p = r^k$
b. $p = \frac{r^k - r^p}{r^{k+p}}$
c. $p = r^p$

11.9 **a.** T = $(21 \times 5 \times 21)^2 = 4{,}862{,}025$
b. p = 1/T ≈ 2×10^{-7}

11.10 There are $95^{10} \approx 6 \times 10^{19}$ possible passwords. The time required is:

$$\frac{6\times10^{19}\text{ passwords}}{6.4\times10^{6}\text{ passwords/second}} = 9.4\times10^{12}\text{ seconds}$$

$$= 300{,}000\text{ years}$$

11.11 **a.** Since PU_a and PR_a are inverses, the value PR_a can be checked to validate that P_a was correctly supplied: Simply take some arbitrary block X and verify that X = D(PRa, E[PUa, X]).

b. Since the file /etc/publickey is publicly readable, an attacker can guess P (say P') and compute $PR_{a'}$ = D(P', E[P, PR_a]). now he can choose an arbitrary block Y and check to see if Y = D(PR_a, E[PU_a, Y]). If so, it is highly probable that P' = P. Additional blocks can be used to verify the equality.

11.12 Yes.

11.13 Without the salt, the attacker can guess a password and encrypt it. If ANY of the users on a system use that password, then there will be a match. With the salt, the attacker must guess a password and then encrypt it once for each user, using the particular salt for each user.

11.14 It depends on the size of the user population, not the size of the salt, since the attacker presumably has access to the salt for each user. The benefit of larger salts is that the larger the salt, the less likely it is that two users will have the same salt. If multiple users have the same salt, then the attacker can do one encryption per password guess to test all of those users.

11.15 **a.** If there is only one hash function (k = 1), which produces one of N possible hash values, and there is only one word in the dictionary, then the probability that an arbitrary bit b_i is set to 1 is just 1/N. If there are k hash functions, let us assume for simplicity that they produce k distinct hash functions for a given word. This assumption only introduces a small margin of error. Then, the probability that an arbitrary bit b_i is set to 1 is k/N. Therefore, the probability that b_i is equal to 0 is 1 – k/N. The probability that a bit is left unset after D dictionary words are processed is just the probability that each of the D transformations set other bits:

$$\Pr[b_i = 0] = \left(1 - \frac{k}{N}\right)^D$$

This can also be interpreted as the expected fraction of bits that are equal to 0.

b. A word not in the dictionary will be falsely accepted if all k bits tested are equal to 1. Now, from part (a), we can say that the expected fraction of bits in the hash table that are equal to one is $1 - \phi$. The probability that a random word will be mapped by a single hash function onto a bit that is already set is the probability that the bit generated by the hash function is in the set of bits equal to one, which is just $1 - \phi$. Therefore, the probability that the k hash functions applied to the word will produce k bits all of which are in the set of bits equal to one is $(1 - \phi)^k$.

c. We use the approximation $(1 - x) \approx e^{-x}$.

11.16 The system enciphers files with a master system key KM, which is stored in some secure fashion. When User i attempts to read file F, the header of F is decrypted using KM and User i's read privilege is checked. If the user has read access, the file is decrypted using KM and the reencrypted using User i's key for transmission to User i. Write is handled in a similar fashion.

CHAPTER 12 FIREWALLS

ANSWERS TO QUESTIONS

12.1 **1.** All traffic from inside to outside, and vice versa, must pass through the firewall. This is achieved by physically blocking all access to the local network except via the firewall. Various configurations are possible, as explained later in this section. **2.** Only authorized traffic, as defined by the local security policy, will be allowed to pass. Various types of firewalls are used, which implement various types of security policies, as explained later in this section. **3.** The firewall itself is immune to penetration. This implies that use of a trusted system with a secure operating system.

12.2 **Service control:** Determines the types of Internet services that can be accessed, inbound or outbound. The firewall may filter traffic on the basis of IP address and TCP port number; may provide proxy software that receives and interprets each service request before passing it on; or may host the server software itself, such as a Web or mail service. **Direction control:** Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall. **User control:** Controls access to a service according to which user is attempting to access it. This feature is typically applied to users inside the firewall perimeter (local users). It may also be applied to incoming traffic from external users; the latter requires some form of secure authentication technology, such as is provided in IPSec. **Behavior control:** Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server.

12.3 **Source IP address:** The IP address of the system that originated the IP packet. **Destination IP address:** The IP address of the system the IP packet is trying to reach. **Source and destination transport-level address:** The transport level (e.g., TCP or UDP) port number, which defines applications such as SNMP or TELNET. **IP protocol field:** Defines the transport protocol. **Interface:** For a router with three or more ports, which interface of the router the packet came from or which interface of the router the packet is destined for.

12.4 **1.** Because packet filter firewalls do not examine upper-layer data, they cannot prevent attacks that employ application-specific vulnerabilities or functions. For example, a packet filter firewall cannot block specific application commands; if a packet filter firewall allows a given application, all functions available within that application will be permitted. **2.** Because of the limited information available to the firewall, the logging functionality present in packet filter firewalls is limited. Packet filter logs normally contain the same information used to make access control decisions (source address, destination address, and traffic type). **3.** Most packet filter firewalls do not support advanced user authentication schemes. Once again, this limitation is mostly due to the lack of upper-layer functionality by the firewall. **4.** They are generally vulnerable to attacks and exploits that take advantage of problems within the TCP/IP specification and protocol stack, such as *network layer address spoofing*. Many packet filter firewalls cannot detect a network packet in which the OSI Layer 3 addressing information has been altered. Spoofing attacks are generally employed by intruders to bypass the security controls implemented in a firewall platform. **5.** Finally, due to the small number of variables used in access control decisions, packet filter firewalls are susceptible to security breaches caused by improper configurations. In other words, it is easy to accidentally configure a packet filter firewall to allow traffic types, sources, and destinations that should be denied based on an organization's information security policy.

12.5 A **traditional packet filter** makes filtering decisions on an individual packet basis and does not take into consideration any higher layer context. A **stateful inspection packet filter** tightens up the rules for TCP traffic by creating a directory of outbound TCP connections, as shown in Table 12.2. There is an entry for each currently established connection. The packet filter will now allow incoming traffic to high-numbered ports only for those packets that fit the profile of one of the entries in this directory

12.6 An application-level gateway, also called a proxy server, acts as a relay of application-level traffic.

12.7 A circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.

12.8 **Packet filtering firewall:** Applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet.
Stateful inspection firewall: Tightens up the rules for TCP traffic by creating a directory of outbound TCP connections, as shown in Table 12.2. There is an entry for each currently established connection. The packet filter will now allow incoming traffic to high-numbered ports only for those packets that fit the profile of one of the entries in this directory.
Application proxy firewall: Acts as a relay of application-level traffic (Figure 12.1d). The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. Further, the gateway can be configured to support only specific features of an application that the network administrator considers acceptable while denying all other features
Circuit-level proxy firewall: As with an application gateway, a circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.

12.9 • The bastion host hardware platform executes a secure version of its operating system, making it a hardened system.
• Only the services that the network administrator considers essential are installed on the bastion host. These could include proxy applications for DNS, FTP, HTTP, and SMTP.
• The bastion host may require additional authentication before a user is allowed access to the proxy services. In addition, each proxy service may require its own authentication before granting user access.
• Each proxy is configured to support only a subset of the standard application's command set.
• Each proxy is configured to allow access only to specific host systems. This means that the limited command/feature set may be applied only to a subset of systems on the protected network.
• Each proxy maintains detailed audit information by logging all traffic, each connection, and the duration of each connection. The audit log is an essential tool for discovering and terminating intruder attacks.

• Each proxy module is a very small software package specifically designed for network security. Because of its relative simplicity, it is easier to check such modules for security flaws. For example, a typical UNIX mail application may contain over 20,000 lines of code, while a mail proxy may contain fewer than 1000.
• Each proxy is independent of other proxies on the bastion host. If there is a problem with the operation of any proxy, or if a future vulnerability is discovered, it can be uninstalled without affecting the operation of the other proxy applications. Also, if the user population requires support for a new service, the network administrator can easily install the required proxy on the bastion host.
• A proxy generally performs no disk access other than to read its initial configuration file. Hence, the portions of the file system containing executable code can be made read only. This makes it difficult for an intruder to install Trojan horse sniffers or other dangerous files on the bastion host.
• Each proxy runs as a nonprivileged user in a private and secured directory on the bastion host.

12.10 • Filtering rules can be tailored to the host environment. Specific corporate security policies for servers can be implemented, with different filters for servers used for different application.
• Protection is provided independent of topology. Thus both internal and external attacks must pass through the firewall.
• Used in conjunction with stand-alone firewalls, the host-based firewall provides an additional layer of protection. A new type of server can be added to the network, with its own firewall, without the necessity of altering the network firewall configuration.

12.11 Between internal and external firewalls are one or more networked devices in a region referred to as a DMZ (demilitarized zone) network. Systems that are externally accessible but need some protections are usually located on DMZ networks. Typically, the systems in the DMZ require or foster external connectivity, such as a corporate Web site, an e-mail server, or a DNS (domain name system) server.

12.12 An **external firewall** is placed at the edge of a local or enterprise network, just inside the boundary router that connects to the Internet or some wide area network (WAN). One or more **internal firewalls** protect the bulk of the enterprise network.

ANSWERS TO PROBLEMS

12.1 It will be impossible for the destination host to complete reassembly of the packet if the first fragment is missing, and therefore the entire packet will be discarded by the destination after a time-out.

12.2 When a TCP packet is fragmented so as to force interesting header fields out of the zero-offset fragment, there must exist a fragment with FO equal to 1. If a packet with FO = 1 is seen, conversely, it could indicate the presence, in the fragment set, of a zero-offset fragment with a transport header length of eight octets Discarding this one-offset fragment will block reassembly at the receiving host and be as effective as the direct method described above.

12.3 If the router's filtering module enforces a minimum fragment offset for fragments that have non-zero offsets, it can prevent overlaps in filter parameter regions of the transport headers.

12.4
1. Allow return TCP Connections to internal subnet.
2. Prevent Firewall system itself from directly connecting to anything.
3. Prevent External users from directly accessing the Firewall system.
4. Internal Users can access External servers,
5. Allow External Users to send email in.
6. Allow External Users to access WWW server.
7. Everything not previously allowed is explicitly denied.

12.5
a. Rules A and B allow inbound SMTP connections (incoming email) Rules C and D allow outbound SMTP connections (outgoing email) Rule E is the default rule that applies if the other rules do not apply.
b. Packet 1: Permit (A); Packet 2: Permit (B): Packet 3: Permit (C) Packet 4: Permit (D)
c. The attack could succeed because in the original filter set, rules B and D allow all connections where both ends are using ports above 1012.

12.6
a. A source port is added to the rule set.
b. Packet 1: Permit (A); Packet 2: Permit (B): Packet 3: Permit (C) Packet 4: Permit (D); Packet 5: Deny (E); Packet 6: Deny (E)

12.7
a. Packet 7 is admitted under rule D. Packet 8 is admitted under rule C.
b. Add a column called ACK Set, with the following values for each rule: A = Yes; B = Yes; C = Any; D = Yes; E = Any

12.8 A requirement like "all external Web traffic must flow via the organization's Web proxy." is easier stated than implemented. This is because identifying what actually constitutes “web traffic” is highly problematical. Although the standard port for HTTP web servers is port

80, servers are found on a large number of other ports (including servers belonging to large, well-known and widely used organizations). This means it is very difficult to block direct access to all possible web servers just using port filters. Whilst it is easy enough to configure web browser programs to always use a proxy, this will not stop direct access by other programs. It also means that the proxy server must have access to a very large number of external ports, since otherwise access to some servers would be limited. As well as HTTP access, other protocols are used on the web. All of these should also be directed via the proxy in order to implement the desired policy. But this may impact the operation of other programs using these protocols. In particular, the HTTPS protocol is used for secure web access that encrypts all traffic flowing between the client and the server. Since the traffic is encrypted, it means the proxy cannot inspect its contents in order to apply malware, SPAM or other desired filtering. Whilst there are some mechanisms for terminating the encrypted connections at the proxy, they have limitations and require the use of suitable browsers and proxy servers.

12.9 A possible requirement to manage information leakage requires all external e-mail to be given a sensitivity tag (or classification) in its subject and for external e-mail to have the lowest sensitivity tag. At its simplest a policy can just require user's to always include such a tag in email messages. Alternatively with suitable email agent programs it may be possible to enforce the prompting for and inclusion of such a tag on message creation. Then, when external email is being relayed through the firewall, the mail relay server must check that the correct tag value is present in the Subject header, and refuse to forward the email outside the organization if not, and notify the user of its rejection.

12.10 Suitable packet filter rulesets FOR the "External Firewall" and the "Internal Firewall" respectively, to satisfy the stated "informal firewall policy", could be:

action	src	port	dest	port	flags	comment
permit	DMZ mail gateway	any	any	SMTP (25)		header sanitize
permit	any	any	DMZ mail gateway	SMTP (25)		content filtered
permit	any	any	DMZ mail gateway	POP3S (995)		user auth
permit	DMZ web proxy	any	any	HTTP/S (80,443)		content filtered, user auth
permit	DMZ DNS server	DNS (53)	any	DNS (53)		TCP & UDP
permit	any	DNS (53)	DMZ DNS server	DNS (53)		TCP & UDP
permit	any	any	any DMZ server	any	estab-lished	return traffic flow
deny	any	any	any	any		block all else

action	src	port	dest	port	flags	comment
permit	any internal	any	DMZ mail gateway	SMTP (25)		
permit	any internal	any	DMZ mail gateway	POP3/S (110,995)		user auth
permit	any internal	any	DMZ web proxy	HTTP/S (80,443)		content filtered, user auth
permit	any internal	DNS (53)	DMZ DNS server	DNS (53)		UDP lookup
permit	DMZ DNS server	DNS (53)	any internal	DNS (53)		UDP lookup
permit	any internal	any	any DMZ server	SSH (22)		user auth on server
permit	mgmt user hosts	any	any DMZ server	SNMP (161)		
permit	any DMZ server	any	mgmt user hosts	SNMP TRAP (162)		
permit	any DMZ server	any	any internal	any	estab-lished	return traffic flow
deny	any	any	any	any		block all else

CHAPTER 13 NETWORK MANAGEMENT SECURITY

ANSWERS TO QUESTIONS

13.1 **1.** A single operator interface with a powerful but user-friendly set of commands for performing most or all network management tasks. **2.** A minimal amount of separate equipment. That is, most of the hardware and software required for network management is incorporated into the existing user equipment.

13.2 Management station, management agent, management information base, network management protocol.

13.3 To manage resources in the network, each resource is represented as an object. An object is, essentially, a data variable that represents one aspect of the managed agent. The collection of objects is referred to as a **management information base** (MIB).

13.4 **Get:** enables the management station to retrieve the value of objects at the agent. **Set:** enables the management station to set the value of objects at the agent.

Notify: enables an agent to notify the management station of significant events.

13.5 To accommodate devices that do not implement SNMP, the concept of proxy was developed. In this scheme an SNMP agent acts as a proxy for one or more other devices; that is, the SNMP agent acts on behalf of the proxied devices.

13.6 An **SNMP community** is a relationship between an SNMP agent and a set of SNMP managers that defines authentication, access control, and proxy characteristics. The community concept is a local one, defined at the agent. The agent establishes one community for each desired combination of authentication, access control, and proxy characteristics. Each community is given a unique (within this agent) community name, and the managers within that community are provided with and must employ the community name in all get and set

operations. The agent may establish a number of communities, with overlapping manager membership.

13.7 SNMPv1 is the original standard version of SNMP. SNMPv2 added functional capabilities to those of SNMPv1 and changed some formats. SNMPv3 is a security facility that can work with either SNMPv1 or SNMPv2.

13.8 **Modification of information:** An entity could alter an in-transit message generated by an authorized entity in such a way as to cause unauthorized management operations, including the setting of object values. The essence of this threat is that an unauthorized entity could change any management parameter, including those related to configuration, operations, and accounting. **Masquerade:** Management operations that are not authorized for some entity may be attempted by that entity by assuming the identity of an authorized entity. **Message stream modification:** SNMP is designed to operate over a connectionless transport protocol. There is a threat that SNMP messages could be reordered, delayed, or replayed (duplicated) to cause unauthorized management operations. **Disclosure:** An entity could observe exchanges between a manager and an agent and thereby learn the values of managed objects and learn of notifiable events.

13.9 In any message transmission, one of the two entities, transmitter or receiver, is designated as the authoritative SNMP engine, according to the following rules:

1. When an SNMP message contains a payload that expects a response (for example, a Get, GetNext, GetBulk, Set, or Inform PDU), then the receiver of such messages is authoritative. **2.** When an SNMP message contains a payload that does not expect a response (for example, an SNMPv2-Trap, Response, or Report PDU), then the sender of such a message is authoritative.

13.10 A localized key is defined in RFC 2574 as a secret key shared between a user and one authoritative SNMP engine. The objective is that the user need only maintain a single key (or two keys if both authentication and privacy are required) and therefore need only remember one password (or two). The actual secrets shared between a particular user and each authoritative SNMP engine are different. The process by which a single user key is converted into multiple unique keys, one for each remote SNMP engine, is referred to as key localization.

13.11 **Group:** a set of zero or more <securityModel, securityName> tuples on whose behalf SNMP management objects can be accessed.

Security level: Determines access rights for a group. For example, an agent may allow read-only access for a request communicated in an unauthenticated message but may require authentication for write access. **Context:** a named subset of the object instances in the local MIB. Contexts provide a useful way of aggregating objects into collections with different access policies. **MIB view:** a specific set of managed objects (and optionally specific object instances). **Access policy:** a particular set of access rights.

ANSWERS TO PROBLEMS

13.1 **a.** The value of a Gauge has its maximum value whenever the information being modeled is greater than or equal to that maximum value; if the information being modeled subsequently decreases below the maximum value, the Gauge remains at the maximum value. The gauge can only be released from this maximum value by subsequent management action.

b. The SNMPv2 interpretation provides a realistic representation of the underlying value at all times, subject to the limitation of the gauge. However, a manager may want to know that some maximum value has been reached or exceeded. By "sticking" the gauge at its maximum value until it is noticed and released by a manager, this information is preserved.

13.2

MIB Access Category	SNMP Access Mode	
	READ-ONLY	**READ-WRITE**
read-only	Available for Get and Trap operations	—
read-write	Available for Get and Trap operations	Available for Get, Set, and Trap operations
write-only	Available for Get and Trap Operations, but the value is implementation-specific.	Available for Get, Set, and Trap Operations, but the value is implementation-specific for Get and Set.
not-accessible	Unavailable	

13.3 **a.** This restriction makes sense because the authoritative receiver will only check those fields if the message is to be authenticated.

b. Keep in mind that in the case of the authoritative sender, these values represent the "official" local values of snmpEngineBoots and snmpEngineTime. When the Response message is received by the non-authorititave engine, it may only use these values for

synchronization if the message is authenticated. However, an implementation might perhaps use these values for a "reality check" even on non-authenticated Response messages.

13.4 If we had done the time window check first, we would have declared the message untimely because MAET < (SET – 150). That doesn't seem good, but that is what the RFC says.

13.5 In the example just given, the authoritative SNMP engine is more than 150 seconds behind the non-authoritative engine (because msgAuthoritativeEngineTime < (snmpEngineTime – 150)) but time synchronization occurs (because latestReceivedEngineTime < msgAuthoritativeEngineTime).

13.6 The first method is straightforward but has the drawback that it requires the use of encryption even in systems that support only message authentication. A related drawback is that export restrictions from the United States and possibly other countries could complicate the use of an encryption-based approach.

13.7 Note that the value protocolKeyChange is just the concatenation of random and delta. So the receiver can compute:
digest = Hash(keyOld || random)
keyNew = digest ⊕ delta

13.8 **(1)** The one-way function puts an impenetrable barrier between the old and new keys, so that if the new key is discovered, it is still infeasible to recover the older key. **(2)** Deducing relationship between the bits of the old and new keys is intractable in both directions (forward and backward), so that the cryptanalyst only has the traffic protected by a given key to use in attempting to determine that key.

CHAPTER 14 LEGAL AND ETHICAL ASPECTS

ANSWERS TO QUESTIONS

14.1
- Computers as targets: This form of crime targets a computer system, to acquire information stored on that computer system, to control the target system without authorization or payment (theft of service), or to alter the integrity of data or interfere with the availability of the computer or server. Using the terminology of Chapter 1, this form of crime involves an attack on data integrity, system integrity, data confidentiality, privacy, or availability.
- Computers as storage devices: Computers can be used to further unlawful activity by using a computer or a computer device as a passive storage medium. For example, the computer can be used to store stolen password lists, credit card or calling card numbers, proprietary corporate information, pornographic image files, or "warez" (pirated commercial software).
- Computers as communications tools: Many of the crimes falling within this category are simply traditional crimes that are committed online. Examples include the illegal sale of prescription drugs, controlled substances, alcohol, and guns; fraud; gambling; and child pornography.

14.2
- Real property: Land and things permanently attached to the land, such as trees, buildings, and stationary mobile homes.
- Personal property: Personal effects, moveable property and goods, such as cars, bank accounts, wages, securities, a small business, furniture, insurance policies, jewelry, patents, pets, and season baseball tickets.
- Intellectual property: Any intangible asset that consists of human knowledge and ideas. Examples include software, data, novels, sound recordings, the design of a new type of mousetrap, or a cure for a disease.

14.3
- Copyrights: Copyright law protects the tangible or fixed expression of an idea, not the idea itself.

- Trademarks: A trademark is a word, name, symbol, or device that is used in trade with goods to indicate the source of the goods and to distinguish them from the goods of others.
- Patents: A patent for an invention is the grant of a property right to the inventor.

14.4 (1) The proposed work is original. (2) The creator has put this original idea into a concrete form, such as hard copy (paper), software, or multimedia form.

14.5
- Reproduction right: Lets the owner make copies of a work
- Modification right: Also known as the derivative-works right, concerns modifying a work to create a new or derivative work
- Distribution right: Lets the owner publicly sell, rent, lease, or lend copies of the work.
- Public-performance right: Applies mainly to live performances
- Public-display right: Lets the owner publicly show a copy of the work directly or by means of a film, slide, or television image

14.6 The DMCA, signed into law in 1998, is designed to implement World Intellectual Property Organization (WIPO) treaties, signed in 1996. In essence, DMCA strengthens the protection of copyrighted materials in digital format.

14.7 Digital Rights Management (DRM) refers to systems and procedures that ensure that holders of digital rights are clearly identified and receive the stipulated payment for their works.

14.8
- Content provider: Holds the digital rights of the content and wants to protect these rights. Examples are a music record label and a movie studio.
- Distributor: Provides distribution channels, such as an online shop or a Web retailer. For example, an online distributor receives the digital content from the content provider and creates a Web catalog presenting the content and rights metadata for the content promotion.
- Consumer: Uses the system to access the digital content by retrieving downloadable or streaming content through the distribution channel and then paying for the digital license. The player/viewer application used by the consumer takes charge of initiating license request to the clearinghouse and enforcing the content usage rights.
- Clearinghouse: Handles the financial transaction for issuing the digital license to the consumer and pays royalty fees to the content provider and distribution fees to the distributor accordingly. The

clearinghouse is also responsible for logging license consumptions for every consumer.

14.9
- Notice: Organizations must notify individuals what personal information they are collecting, the uses of that information, and what choices the individual may have.
- Consent: Individuals must be able to choose whether and how their personal information is used by, or disclosed to, third parties. They have the right not to have any sensitive information collected or used without express permission, including race, religion, health, union membership, beliefs, and sex life.
- Consistency: Organizations may use personal information only in accordance with the terms of the notice given the data subject and any choices with respect to its use exercised by the subject.
- Access: Individuals must have the right and ability to access their information and correct, modify, or delete any portion of it.
- Security: Organizations must provide adequate security, using technical and other means, to protect the integrity and confidentiality of personal information.
- Onward transfer: Third parties receiving personal information must provide the same level of privacy protection as the organization from whom the information is obtained.
- Enforcement: The Directive grants a private right of action to data subjects when organizations do not follow the law. In addition, each EU member has a regulatory enforcement agency concerned with privacy rights enforcement.

14.10
1. A code can serve two inspirational functions: as a positive stimulus for ethical conduct on the part of the professional, and to instill confidence in the customer or user of an IS product or service. However, a code that stops at just providing inspirational language is likely to be vague and open to an abundance of interpretations.
2. A code can be educational. It informs professionals about what should be their commitment to undertake a certain level of quality of work and their responsibility for the well being of users of their product and the public, to the extent the product may affect nonusers. The code also serves to educate managers on their responsibility to encourage and support employee ethical behavior and on their own ethical responsibilities.
3. A code provides a measure of support for a professional whose decision to act ethically in a situation may create conflict with an employer or customer.
4. A code can be a means of deterrence and discipline. A professional society can use a code as a justification for revoking membership

or even a professional license. An employee can use a code as a basis for a disciplinary action.

5. A code can enhance the profession's public image, if it is seen to be widely honored.

Answers to Problems

14.1 **Article 2 Illegal access**: This is a general threat the could fall into any of the three categories, depending on what use is made of the access.
Article 3 Illegal interception: Computer as target, attack on data confidentiality.
Article 4 Data interference: Computer as target, attack on data integrity.
Article 5 System interference: Computer as target, various attack types.
Article 6 Misuse of devices: Primarily computer as communications tool.
Article 7 Computer-related forgery: Computer as target, data integrity or privacy.
Article 8 Computer-related fraud: Computer as communications tool
Article 9 Offenses related to child pornography: Computer as communications tool.
Article 10 Infringements of copyright and related rights: Computer as communications tool.
Article 11 Attempt and aiding or abetting: Computer as communications tool.

14.2 **Theft of intellectual property:** Computer as target, attack on data confidentiality.
Theft of other (proprietary) info including customer records, financial records, etc.: Computer as target, attack on privacy.
Denial of service attacks: Computer as target, attack on availability.
Virus, worms or other malicious code: This is a general threat the could fall into any of the three categories, depending on what use is made of the attack.
Fraud (credit card fraud, etc.): Computer as communications tool.
Identity theft of customer: Computer as communications tool.
Illegal generation of spam e-mail. Computer as communications tool.
Phishing: Computer as target, attack on privacy.
Unauthorized access to/use of information, systems or networks: This is a general threat the could fall into any of the three categories, depending on what use is made of the attack.

Sabotage: deliberate disruption, deletion, or destruction of information, systems, or networks: Computer as target, attack on availability.
Extortion: Computer as communications tool.
Web site defacement: Computer as target, attack on data integrity.
Zombie machines on organization's network/bots/use of network by BotNets: Computer as communications tool.
Intentional exposure of private or sensitive information: Computer as target, attack on privacy.
Spyware (not including adware): Computer as communications tool.

14.3 There is no simple answer to this problem, as it depends on which survey is reviewed, given that the details do change from year to year and region to region. Any answer should note significant changes in the types of crime reported, and differences between the survey results and those shown in Table 14.2.

14.4 There is no single answer to this problem. However a web search on 'DeCSS' should be done. Two key current sites are the Gallery of CSS Descramblers at CMU and the Wikipedia DeCSS page which both provide many details and further links on the case. Given the very large number of items in the Gallery of CSS Descramblers it is fair to conclude that the MPAA failed to suppressing details of the DeCSS descrambling algorithm.

14.5 If a person purchases a track from the iTunes store, protected by Apple's FairPlay DRM, by an EMI artist, then the DRM component roles shown in Figure 14.3 in this case are: Content Provider is EMI, Distributor and Clearinghouse are both handled by the iTunes Store, and the Consumer is the person purchasing the track.

14.6 EU calls out the need for notice. This proactive measure is worthwhile. OECD mentions collection limitation, not explicitly called out in the EU list. Again, a worthwhile principle.

14.7 There is no simple answer to this problem, as it depends on the relevant organization's Privacy Policy. However any answer should consider all the principles listed in section 14.3, and should also refer to any relevant privacy legislation that applies to the chosen organization.

14.8 In this scenario, the administrator has very likely broken the law (though it depends on the jurisdiction applying), and breached company policy (provided they actually had one), even if for potentially altruistic reasons. The actions likely violated several of the

potential ethical dilemmas listed in Table 14.3 including employee monitoring (in checking their passwords), hacking (in accessing the password files from other sections), and even internal privacy (knowing other user's passwords gives access to their data that you otherwise do not have authorization for). You might defend yourself by arguing that as a systems administrator you were authorized to access the password file. Unfortunately you are not the administrator for the section whose password file was cracked, and it will be difficult to argue that you had authority to do so. You would also have to argue that you had no intent to use that data to break any law, that your motives were not malicious and that they were in the interests of the organization and its employees. You might support these arguments by referring to item 2.5 (analysis of risks) in the ACM code, and item 7 (correct errors) in the IEEE code. The counter argument is that you failed to obey for example item 2.8 (authorized access) in the ACM code. Clearly the outcome would have been more satisfactory if the administrator had raised the issue of password security with senior management, and been granted permission to conduct the survey of current password security in a manner consistent with the law and company policy.

14.9 Assume appropriate section and subsection numbering for AITP.

	ACM	IEEE	AITP
dignity and worth of people	1.2	8, 9	—
personal integrity	Section 2	2, 3, 4	2.1, 3.6
responsibility for work	Section 2	1	1.3
confidentiality of information	1.7, 1.8	—	3.1, 4.5
public safety, health, and welfare	1.1, 1.2	1	3.3
participation in professional societies	—	—	—
knowledge about technology related to social power.	2.7	5	4.8

14.10 a. EC1.2, EC2.2, and EC4.1 seem designed more to protect ACM's reputation than to focus on the professionals ethical responsibility and so can reasonably be excluded. EC 2.3 and EC 3.1 are not explicit in the 1997 Code and perhaps should be. They are covered implicitly however.

b. In a number of areas, the 1997 Code is more detailed and more explicit, which provides better guidance to the professional. For

example, the 1997 Code includes references to being aware of the legal responsibilities of professionals and managerial obligations.

14.11 **a.** I.3 refers to adequate compensation; this does not seem to be on target for an ethics code. II.b refers to disseminating information. Even though this is qualified with respect to legal and proprietary restraints, it seems better not to include this in the Code. II.e seems designed more for IEEE's benefit than the individual's. Section III, on responsibilities to employers and clients, is not explicit in the 2006 Code and perhaps should be.

b. Nothing new in the 2006 Code not covered in the older Code.

14.12 **a.** ACM Code. The Software Engineering Code (SEC) specifically calls out responsibilities to client and employer. Perhaps ACM Code should as well. In general SEC is more detailed; this has the benefit of covering more ground in more detail but the disadvantage of discouraging professionals from reading the whole code.

b. IEEE Code. SEC specifically calls out responsibilities to client and employer. SEC specifically addresses confidentiality. Both should probably be addressed in IEEE code.

c. AITP Code. SEC refers to the quality of the products of the professional. AITP does not specifically call this out.

CHAPTER 15 SHA-3

ANSWERS TO QUESTIONS

15.1 The criteria fall into three categories:
•Security: The evaluation considered the relative security of the candidates compared to each other and to SHA-2. In addition, specific security requirements related to various applications and resistance to attacks are included in this category.
•Cost: NIST intends SHA-3 to be practical in a wide range of applications. Accordingly, SHA-3 must have high computational efficiency, so as to be usable in high-speed applications, such as broadband links, and low memory requirements.
•Algorithm and implementation characteristics: This category includes a variety of considerations, including flexibility; suitability for a variety of hardware and software implementations; and simplicity, which will make an analysis of security more straightforward.

15.2 The sponge construction has the same general structure as other iterated hash functions. The sponge function 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. Unlike a typical hash function, the sponge construction allows for a variable-length output.

15.3 The function *f* is executed once for each input block of the message to be hashed. The function takes as input the 1600-bit state variable and converts it into a 5 × 5 matrix of 64-bit lanes. This matrix then passes through 24 rounds of processing. Each round consists of five steps, and each step updates the state matrix by permutation or substitution operations. The rounds are identical with the exception of the final step in each round, which is modified by a round constant that differs for each round.

15.4 **Theta:** 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.
Rho: The bits of each word are permuted using a circular bit shift. W[0, 0] is not affected.
Pi: Words are permuted in the 5×5 matrix. W[0, 0] is not affected.

Chi: 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.
Theta: W[0, 0] is updated by XOR with a round constant.

ANSWERS TO PROBLEMS

15.1 c = 448: 448/64 = 7 lanes of all zeros. This includes all 5 lanes in row y = 0, plus two lanes in row y = 1, namely L[0, 1], L[1, 1].
c = 512: 512/64 = 8 lanes of all zeros. This includes all the lanes in row y = 0, plus three lanes in row y = 1, namely L[0, 1], L[1, 1], L[2, 1].
c = 768: 768/64 = 12 lanes of all zeros. This includes all the lanes in rows y = 0 and y = 1, plus two lanes in row y = 2, namely L[0, 2], L[1, 2].
c = 1024: 1024/64 = 16 lanes of all zeros. This includes all the lanes in rows y = 0, y = 1, and y = 2, plus L[0, 3].

15.2 Potentially, all of the lanes will have at least one 1 bit after the theta step function in Round 0. This is because every column has at least one nonzero lane, and every lane is updated by the XOR of itself and all of the lanes in the preceding and following columns (with a bit position shift in the following column). It is possible that the XOR would result in a zero result for all 64 bits of a lane and so that lane would remain zero. In that case the chi step function is the next possible function to achieve the result we are looking for. In this case, a lane is updated as a function of the next two lanes in its row. If we assume that a given lane is all zeros, then the calculation is (see Equation 15.4)

$$\text{NOT}(a[x+1]) \text{ AND } a[x+2]$$

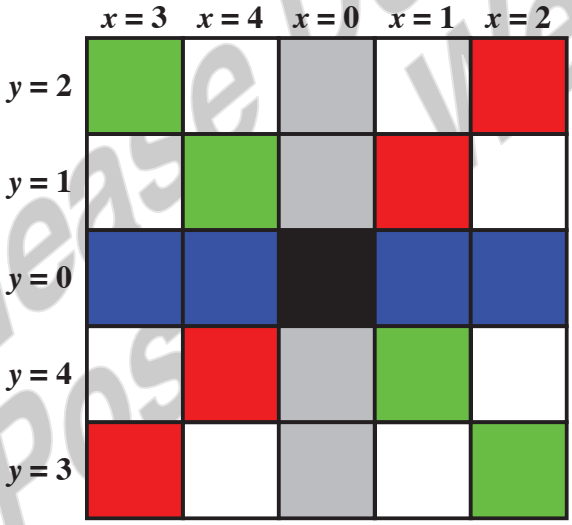
This will yield a zero result unless there is at least one bit position out of 64 bit positions such that that bit is 0 in the (x+1) lane and 1 in the (x+2). If at the end of Round 0 there is still one or more of the initial zero words that are still zero, then there is a high probability that it will pick up at least one 1 bit in the theta or chi step of Round 1.

15.3 It is perhaps easier to visualize the permutation in this orientation. We have:

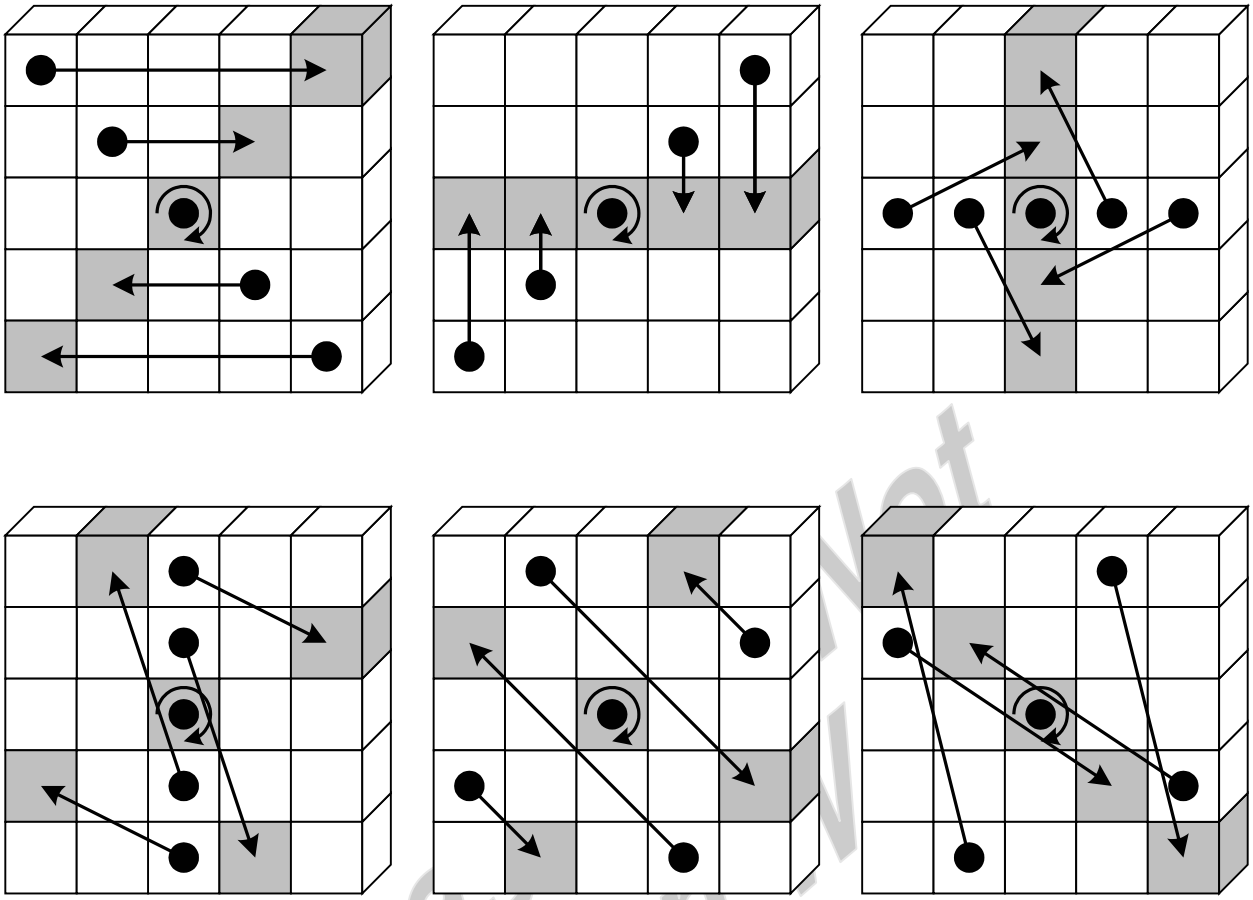
	$x=3$	$x=4$	$x=0$	$x=1$	$x=2$
$y=2$	L[3,2]	L[4,2]	L[0,2]	L[1,2]	L[2,2]
$y=1$	L[3,1]	L[4,1]	L[0,1]	L[1,1]	L[2,1]
$y=0$	L[3,0]	L[4,0]	L[0,0]	L[1,0]	L[2,0]
$y=4$	L[3,4]	L[4,4]	L[0,4]	L[1,4]	L[2,4]
$y=3$	L[3,3]	L[4,3]	L[0,3]	L[1,3]	L[2,3]

→

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



Now visualize the matrix as having 4 straight lines through five squares: vertical, horizontal, and the two diagonals. The center square of each line is L[0,0], which does not change position. Now consider this figure from the Keccak documentation show the pi permutation when the rows and columns are organized as indicated above.



We see that the falling diagonal is mapped to the rising diagonal, the rising diagonal is mapped to the central row, and the central row is mapped into the central column. The remaining three mappings are less succinctly described. Still, this orientation is useful for getting a feel for what the permutation accomplishes.

15.4 Let us say we are concerned with the execution of the iota function in round *i*.

a. During the theta step function in round *i*+1, every lane in column *x* = 1 and column *x* = 4 is updated with *L*[0, 0] as one of the inputs to the calculation. For a moment, ignore the pi step and assume that we were to go immediately to the chi step. Every lane in column x = 2 and x = 3 is affected by the corresponding lane in x = 4, which has already been updated in the theta step to incorporate *L*[0, 0]. Similarly, every lane in x = 0 is affected by the corresponding lane in x = 1, which has already been updated in the theta step to incorporate *L*[0, 0]. Thus, during round *i*+1 all lanes are affected by the changes to *L*[0, 0] during round *i*, via the theta and step functions. It can be shown that the permutation pi does not affect this reasoning. This is left as a further exercise to the student.

b. Keep in mind that only a few bit positions in $L[0, 0]$ are affected by the iota function (at most 6). Thus during round $i+1$, only a few bit positions in each lane are affected. By the same reasoning as that of the answer to Problem 11.14, we can expect that there is high probability that all bit will be affected by the end of round $i+2$, and even higher probability by the end of round $i+3$.