SOLUTIONS MANUAL

Cryptography and Network Security: Principles and Practice Sixth Edition

Chapters 12–24

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 https://www.box.com/shared/nh8hti5167 File name is S-Crypto6e-mmyy.

W.S.

TABLE OF CONTENTS

CHAPTER 12 MESSAGE AUTHENTICATION CODES

ANSWERS TO QUESTIONS

12.1 **Masquerade:** Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or nonreceipt by someone other than the message recipient. **Content modification:** Changes to the contents of a message, including insertion, deletion, transposition, and modification. **Sequence modification:** Any modification to a sequence of messages between parties, including insertion, deletion, and reordering. **Timing modification:** Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.

12.2 At the lower level, there must be some sort of function that produces an authenticator: a value to be used to authenticate a message. This lower-level function is then used as primitive in a higher-level authentication protocol that enables a receiver to verify the authenticity of a message.

12.3 Message encryption, message authentication code, hash function.

12.4 Error control code, then encryption.

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

12.6 A hash function, by itself, does not provide message authentication. A secret key must be used in some fashion with the hash function to produce authentication. A MAC, by definition, uses a secret key to calculate a code used for authentication.

12.7 Figures 11.2 and 11.3 illustrate a variety of ways in which a hash code can be used to provide message authentication, as follows. **Figure 11.2: a.** The message plus concatenated hash code is encrypted using symmetric encryption. **b.** Only the hash code is encrypted, using symmetric encryption. **c.** Only the hash code is encrypted, using public-key encryption and using the sender's private key. **d.** If confidentiality as well as a digital signature is desired, then the message plus the public-key-encrypted hash code can be encrypted using a symmetric secret key. **Figure 11.3: a.** This technique uses a hash function but no encryption for message authentication. The technique assumes that the two communicating parties share a common secret value S. A computes the hash value over the concatenation of M and S and appends the resulting hash value to M. Because B possesses S, it can recompute the hash value to verify. **b.** Confidentiality can be added to the approach of (e) by encrypting the entire message plus the hash code.

12.8 No. Section 12.4 outlines such attacks.

12.9 To replace a given hash function in an HMAC implementation, all that is required is to remove the existing hash function module and drop in the new module.

Answers to Problems

12.1 No. If internal error control is used, error propagation in the deciphering operation introduces too many errors for the error control code to correct.

12.2 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.

12.3 We use the definition from Section 12.6. For a one-block message, the MAC using CBC-MAC is $T = \mathrm{E}(K, X)$, where K is the key and X is the message block. Now consider the two-block message in which the first block is X and the second block is $X \oplus T$. Then the MAC is $\mathrm{E}(K, [T \oplus (X \oplus T)]) = \mathrm{E}(K, X) = T$.

12.4 We use Figure 12.8a 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 $\mathrm{E}(K, \mathbf{0}) = \mathrm{CBC}(K, \mathbf{0}) = T_0 \oplus K_1$. 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.

12.5 **a.** In each case (64 bits, 128 bits) the constant is the binary representation of the irreducible polynomial defined in Section 12.6. The two constants are

$$R_{128} = 0^{120}10000111 \quad \text{and} \quad R_{64} = 0^{59}11011$$

b. Here is the algorithm from the NIST document:

```
1. Let $L$ = E($K$, $0^b$).
2. If $\text{MSB}_1(L) = 0$, then $K1 = L << 1$;
   Else $K_1 = (L << 1) \oplus R_b$;
3. If $\text{MSB}_1(K_1) = 0$, then $K_2 = K_1 << 1$;
   Else $K_2 = (K_1 << 1) \oplus Rb$.
```

12.6 **a.** MtE

b. This is basically the E&M approach, but uses one key instead of two.

12.7 As was discussed in Chapter 4, multiplication distributes over addition in a field, and for this type of field the XOR operation is the addition operation. Consider a message consisting of two blocks. Then by Figure 12.10a, the GHASH function is

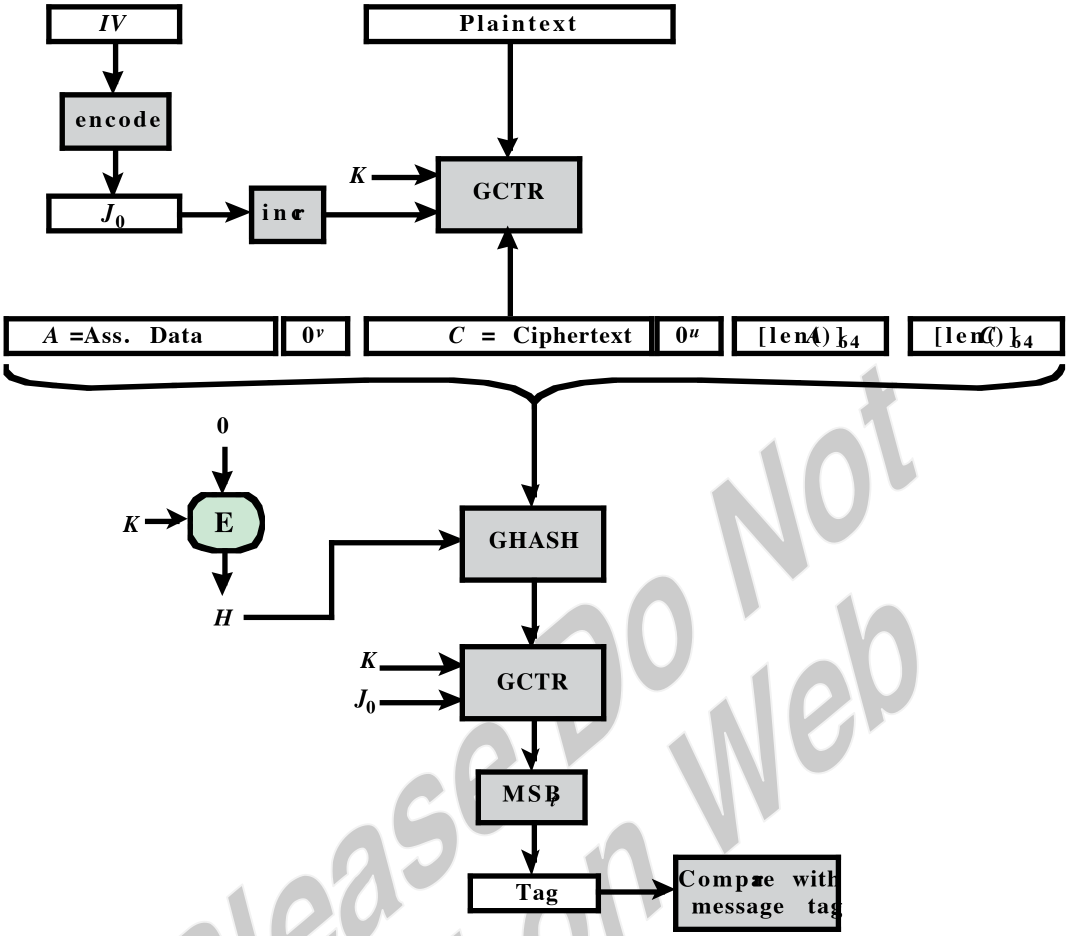
$$(X_1 \bullet H) \oplus X_2) \bullet H)$$

Multiplying through by H, we get the expression

$$(X_1 \bullet H^2) \oplus (X_2 \bullet H)$$

This calculation can be extended through more blocks, up to X_m, to get the expression shown in the statement of this problem.

12.8



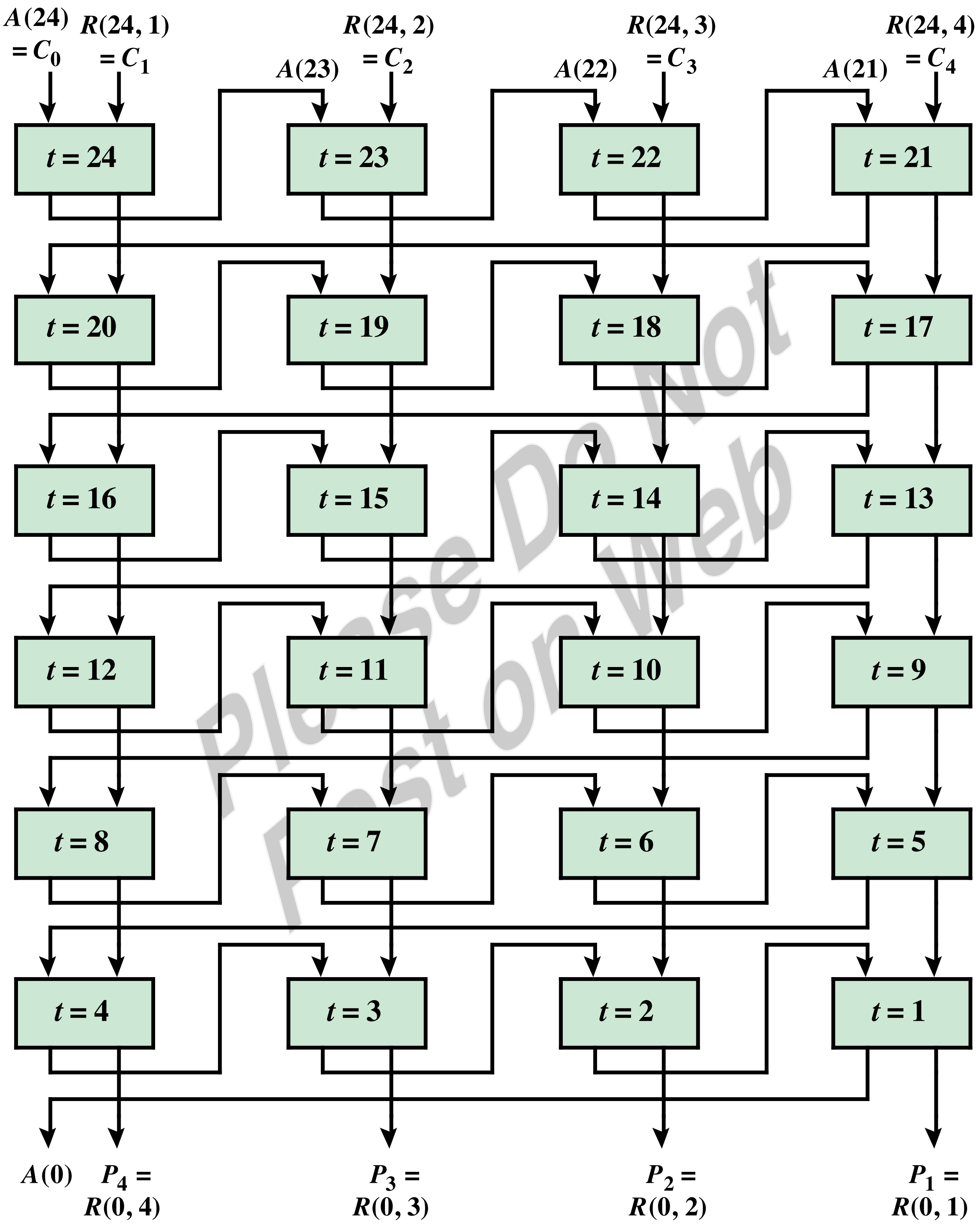
12.9 a. The following matrix shows the message for each received 2-bit word.

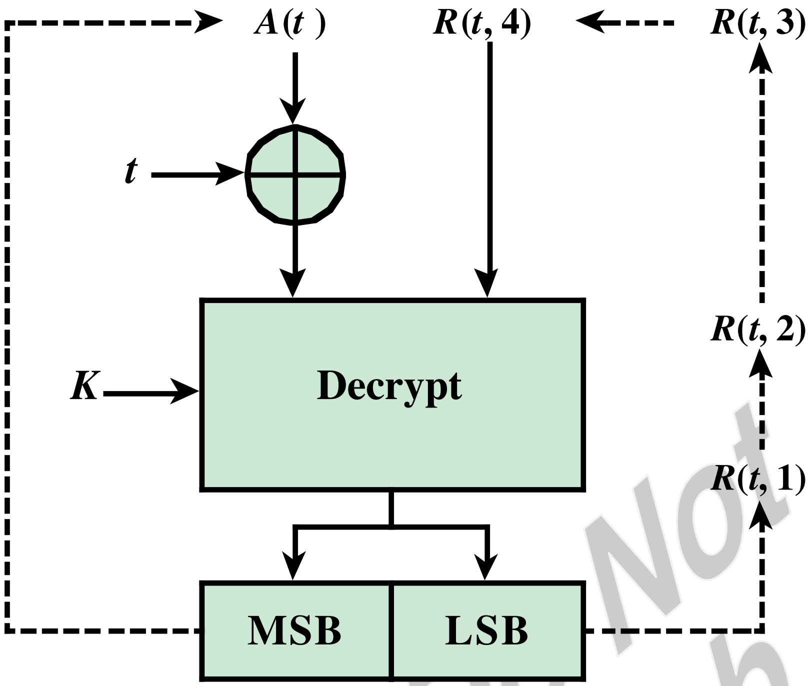
	Word			
Key	00	01	10	11
1	0	1	—	—
2	1	—	0	—
3	—	0	—	1
4	—	—	1	0

b. The probability that some one can successfully impersonate Alice is 0.5 because only two of the four words are possible as transmitted word under the joint secret key.

c. An opponent Eve who tries to replace a transmitted message by another one will know that only two keys can possibly have been used, but she doesn't know which one. So, the probability of a successful substitution is also 0.5.

12.10


A(24) = C0
R(24, 1) = C1
A(23)
R(24, 2) = C2
A(22)
R(24, 3) = C3
A(21)
R(24, 4) = C4
t = 24
t = 23
t = 22
t = 21
t = 20
t = 19
t = 18
t = 17
t = 16
t = 15
t = 14
t = 13
t = 12
t = 11
t = 10
t = 9
t = 8
t = 7
t = 6
t = 5
t = 4
t = 3
t = 2
t = 1
A(0)
P4 = R(0, 4)
P3 = R(0, 3)
P2 = R(0, 2)
P1 = R(0, 1)



12.11 a. This is functionally equivalent; that is, it produces the same result. This alternative description of the key wrap algorithm involves indexing rather than shifting. This approach allows one to calculate the wrapped key in place, avoiding the rotation in the algorithm in Section 12.8. This produces identical results and is more easily implemented in software.

b.

```
1. Initialize variables.
  $A = C_0$
  for $i = 1$ to $n$
      $R(i) = C_i$
2. Calculate intermediate values.
  for $j = 5$ to $0$
      for $i = n$ to $1$
              $t = (n \times j)+i$
              $B = \mathrm{E}(K, [(A \oplus t) \mid\mid R(i)])$
              $A = \mathrm{MSB}_{64}(B)$
              $R(i) = \mathrm{LSB}_{64}(B)$
3. Output results.
  if $A(0) = \mathrm{A6A6A6A6A6A6A6A6}$
  then
      for $i = 1$ to $n$
          $P(i) = R(i)$
  else
      return error
```

CHAPTER 13 DIGITAL SIGNATURES

ANSWERS TO QUESTIONS

13.1 Suppose that John sends an authenticated message to Mary. The following disputes that could arise: **1.** Mary may forge a different message and claim that it came from John. Mary would simply have to create a message and append an authentication code using the key that John and Mary share. **2.** John can deny sending the message. Because it is possible for Mary to forge a message, there is no way to prove that John did in fact send the message.

13.2 **1.** It must be able to verify the author and the date and time of the signature. **2.** It must be able to authenticate the contents at the time of the signature. **3.** The signature must be verifiable by third parties, to resolve disputes.

13.3 **1.** The signature must be a bit pattern that depends on the message being signed. **2.** The signature must use some information unique to the sender, to prevent both forgery and denial. **3.** It must be relatively easy to produce the digital signature.
4. It must be relatively easy to recognize and verify the digital signature. **5.** It must be computationally infeasible to forge a digital signature, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message. **6.** It must be practical to retain a copy of the digital signature in storage.

13.4 A **direct digital signature** involves only the communicating parties (source, destination). It is assumed that the destination knows the public key of the source. A digital signature may be formed by encrypting the entire message with the sender's private key or by encrypting a hash code of the message with the sender's private key. An **arbitrated digital signature** operates as follows. Every signed message from a sender X to a receiver Y goes first to an arbiter A, who subjects the message and its signature to a number of tests to check its origin and content. The message is then dated and sent to Y with an indication that it has been verified to the satisfaction of the arbiter.

13.5 It is important to perform the signature function first and then an outer confidentiality function. In case of dispute, some third party must view the message and its signature. If the signature is calculated on an encrypted message, then the third party also needs access to the decryption key to read the original message. However, if the signature is the inner operation, then the recipient can store the plaintext message and its signature for later use in dispute resolution.

13.6 **1.** The validity of the scheme depends on the security of the sender's private key. If a sender later wishes to deny sending a particular message, the sender can claim that the private key was lost or stolen and that someone else forged his or her signature. **2.** Another threat is that some private key might actually be stolen from X at time T. The opponent can then send a message signed with X's signature and stamped with a time before or equal to T.

Answers to Problems

13.1 Instead of two keys e and d we will have THREE keys u, v, and w. They must be selected in such way that uvw = 1 mod ϕ(N). (This can be done e.g. by selecting u and v randomly (but they have to be prime to ϕ(N)) and then choosing w such that the equation holds.) The key w is made public, while u and v become the first and the second signatory's key respectively. Now the first signatory signs document M by computing $S1 = M^u \bmod N$ The second signatory can verify the signature with the help of his key v and publicly known w, because $S1^{vw} \bmod N$ has to be M. He then 'adds' his signature by computing $S2 = S1^v \bmod N$ (that is $S2 = M^{uv} \bmod N$). Anyone can now verify that S2 is really the double signature of M (i.e. that M was signed by both signatories) because $S2^w \bmod N$ is equal to M only if $S2 = M^{uv} \bmod N$.

13.2 A user who produces a signature with s = 0 is inadvertently revealing his or her private key x via the relationship:

$$s = 0 = k^{-1}[H(m) + xr) \bmod q$$

$$x = \frac{-H(m)}{r} \bmod q$$

13.3 A user's private key is compromised if *k* is discovered.

13.4 **a.** Note that at the start of step 4, $z = b^{2^j m} \bmod w$ .The idea underlying this algorithm is that if (b^m mod *w*) ≠ 1 and $w = 1 + 2^a m$ is prime, the sequence of values

$b^m \bmod w, \quad b^{2m} \bmod w, \quad b^{4m} \bmod w, \ldots$

will end with 1, and the value just preceding the first appearance of 1 will be $w - 1$. Why? Because, if w is prime, then if we have $z^2 \bmod w = 1$, then we have $z^2 \equiv 1 \bmod w$. And if that is true, then $z = (w - 1)$ or $z = (w + 1)$. We cannot have $z = (w + 1)$, because on the preceding step, z was calculated mod w, so we must have $z = (w - 1)$. On the other hand, if we reach a point where $z = 1$, and z was not equal to $(w - 1)$ on the preceding step, then we know that w is not prime.

b. This algorithm is a simplified version of the Miller-Rabin algorithm. In both cases, a test variable is repeatedly squared and computed modulo the possible prime, and the possible fails if a value of 1 is encountered.

13.5 The signer must be careful to generate the values of k in an unpredictable manner, so that the scheme is not compromised.

13.6 **a.** If Algorithm 1 returns the value g, then we see that $g^q = 1 \pmod{p}$. Thus, ord(g) divides q. Because q is prime, this implies that ord(g) $\in \{1, q\}$. However, because $g \neq 1$, we have that ord(g) $\neq 1$, and so it must be that ord(g) = q.

b. If Algorithm 2 returns the value g, then we see that $g^q \equiv \left(h^{p-1/q}\right)^q \equiv h^{p-1} \equiv 1 \left(\bmod p\right)$. Thus, ord($g$) divides q. Because q is prime, this implies that ord(g) $\in \{1, q\}$. However, because $g \neq 1$, we have that ord(g) $\neq 1$, and so it must be that ord(g) = q.

c. Algorithm 1 works by choosing elements of Z_p until it finds one of order q. Since q divides $p - 1$, Z_p contains exactly $\phi(q) = q - 1$ elements of order q. Thus, the probability that $g \in Z_p$ has order q is $(q - 1)/(p - 1)$. When $p = 40193$ and $q = 157$ this probability is 156/40192 . So, we expect Algorithm 1 to make 40192/156 ≈ 258 loop iterations.

d. No. If p is 1024 bits and q is 160 bits, then we expect Algorithm 1 to require $(q - 1)/(p - 1) \approx (2^{1024})/(2^{160}) = 2^{864}$ loop iterations.

e. Algorithm 2 will fail to find a generator in its first loop iteration only if $1 \equiv h^{(p-1)/q} \pmod{p}$. This implies that ord(h) divides $(p - 1)/q$. Thus, the number of bad choices for h is the number of elements of Z_p with order dividing
$(p - 1)/q$:

$$\sum_{d|(p-1)/q} \phi(d)$$

This sum is equal to (p – 1)/q. Thus, the desired probability is:

$$1-\frac{(p-1)/q}{p-1}=1-\frac{1}{q}=\frac{q-1}{q}=\frac{156}{157}\approx 0.994$$

13.7 a. To verify the signature, the user verifies that $(g^Z)^h = g^X \bmod p$.

b. To forge the signature of a message, I find its hash h. Then I calculate Y to satisfy Yh = 1 mod (p – 1). Now $g^{Yh} = g$, so $g^{XYh} = g^X \bmod p$. Hence (h, g^{XY}) is a valid signature and the opponent can calculate g^{XY} as $(g^X)^Y$.

13.8 a. The receiver validates the digital signature by ensuring that the first 56-bit key in the signature will encipher validation parameter *u*1 into E(*k*1, *u*1) if the first bit of *M* is 0, or that it will encipher *U*1 into E(*K*1, *U*1) if the first bit of *M* is 1; the second 56-bit key in the signature will encipher validation parameter *u*2 into E(*k*2, *u*2) if the second bit of *M* is 0, or it will encipher *U*2 into E(*K*2, *U*2) if the second bit of *M* is 1,; and so on.

b. Only the sender, who knows the private values of *ki* and *Ki* and who originally creates *vi* and *Vi* from *ui* and *Ui* can disclose a key to the receiver. An opponent would have to discover the value of the secret keys from the plaintext-ciphertext pairs of the public key, which was computationally infeasible at the time that 56-bit keys were considered secure.

c. This is a one-time system, because half of the keys are revealed the first time.

d. A separate key must be included in the signature for each bit of the message resulting in a huge digital signature.

CHAPTER 14 KEY MANAGEMENT AND DISTRIBUTION

ANSWERS TO QUESTIONS

14.1 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.

14.2 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.

14.3 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.

14.4 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.

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

14.6 Public announcement. Publicly available directory. Public-key authority. Public-key certificates

14.7 **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.

14.8 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.

14.9 **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.

14.10 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.

14.11 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.

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

Answers to Problems

14.1 **a.** A sends a connection request to B, with an event marker or nonce (Na) encrypted with the key that A shares with the KDC. If B is prepared to accept the connection, it sends a request to the KDC for a session key, including A's encrypted nonce plus a nonce generated by B (Nb) and encrypted with the key that B shares with the KDC. The KDC returns two encrypted blocks to B. One block is intended for B and includes the session key, A's identifier, and B's nonce. A similar block is prepared for A and passed from the KDC to B and then to A. A and B have now securely obtained the session key and, because of the nonces, are assured that the other is authentic.

b. The proposed scheme appears to provide the same degree of security as that of Figure 14.3. One advantage of the proposed scheme is that the, in the event that B rejects a connection, the overhead of an interaction with the KDC is avoided.

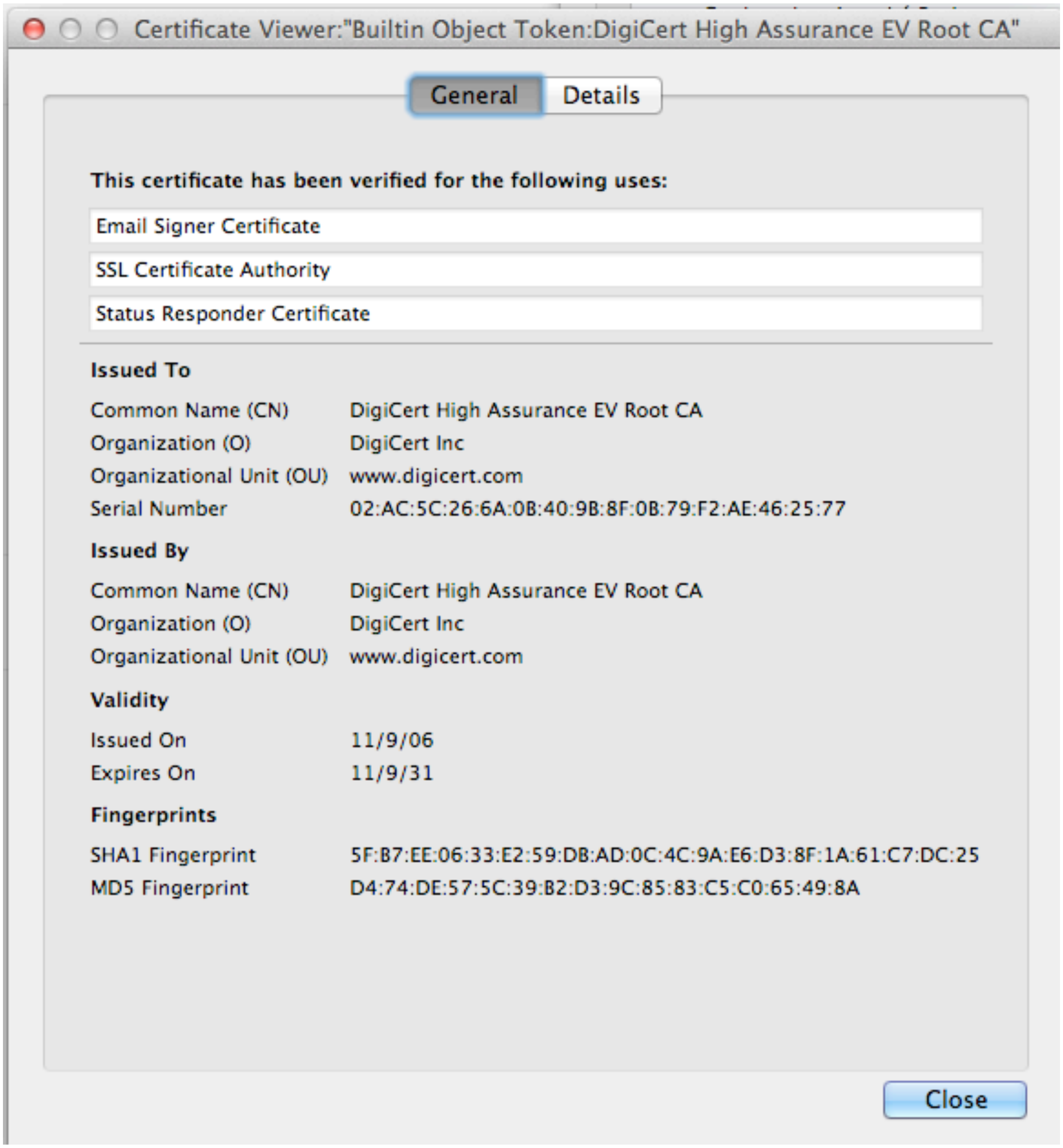
14.2 i) sending to the server the source name A, the destination name Z (his own), and E(K_a, R), as if A wanted to send him the same message encrypted under the same key R as A did it with B

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.

14.3 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.

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


Certificate Viewer:"Builtin Object Token:DigiCert High Assurance EV Root CA"
General
Details
Close

This certificate has been verified for the following uses:

Email Signer Certificate

SSL Certificate Authority

Status Responder Certificate

Issued To

Common Name (CN)	DigiCert High Assurance EV Root CA
Organization (O)	DigiCert Inc
Organizational Unit (OU)	www.digicert.com
Serial Number	02:AC:5C:26:6A:0B:40:9B:8F:0B:79:F2:AE:46:25:77

Issued By

Common Name (CN)	DigiCert High Assurance EV Root CA
Organization (O)	DigiCert Inc
Organizational Unit (OU)	www.digicert.com

Validity

Issued On	11/9/06
Expires On	11/9/31

Fingerprints

SHA1 Fingerprint	5F:B7:EE:06:33:E2:59:DB:AD:0C:4C:9A:E6:D3:8F:1A:61:C7:DC:25
MD5 Fingerprint	D4:74:DE:57:5C:39:B2:D3:9C:85:83:C5:C0:65:49:8A

14.5 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.

14.6 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 $\mathrm{E}(K_{AB}, [\mathrm{A}, \mathrm{B}, N_A, K'_{AB}])$.

14.7 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.

14.8 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.

14.9 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.

14.10 Host *i* has master key KMH_i, with variants $\mathrm{KMH}_{i,j}$, j = 0, 1, 2.

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

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

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

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

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

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

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

$$\mathrm{KEYGEN}[\mathrm{RN}, \mathrm{KMT}_i, \mathrm{KMT}_j]$$

which will interpret RN as E(KMH0, KS) and return both $\mathrm{E}(\mathrm{KMH}_i, \mathrm{KS})$ and $\mathrm{E}(\mathrm{KMH}_j, \mathrm{KS})$, which are sent to the terminals *i* and *j*, respectively. RN need not be maintained at the host.

CHAPTER 15 USER AUTHENTICATION

ANSWERS TO QUESTIONS

15.1 **Simple replay:** The opponent simply copies a message and replays it later. **Repetition that can be logged:** An opponent can replay a timestamped message within the valid time window. **Repetition that cannot be detected:** This situation could arise because the original message could have been suppressed and thus did not arrive at its destination; only the replay message arrives. **Backward replay without modification:** This is a replay back to the message sender. This attack is possible if symmetric encryption is used and the sender cannot easily recognize the difference between messages sent and messages received on the basis of content.

15.2 **1.** Attach a sequence number to each message used in an authentication exchange. A new message is accepted only if its sequence number is in the proper order. **2.** Party A accepts a message as fresh only if the message contains a timestamp that, in A's judgment, is close enough to A's knowledge of current time. This approach requires that clocks among the various participants be synchronized. **3.** Party A, expecting a fresh message from B, first sends B a nonce (challenge) and requires that the subsequent message (response) received from B contain the correct nonce value.

15.3 When a sender's clock is ahead of the intended recipient's clock., an opponent can intercept a message from the sender and replay it later when the timestamp in the message becomes current at the recipient's site. This replay could cause unexpected results.

15.4 The problem that Kerberos addresses is this: Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restrict access to authorized users and to be able to authenticate requests for service. In this environment, a workstation cannot be trusted to identify its users correctly to network services.

15.5 **1.** A user may gain access to a particular workstation and pretend to be another user operating from that workstation. **2.** A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation. **3.** A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations.

15.6 **1.** Rely on each individual client workstation to assure the identity of its user or users and rely on each server to enforce a security policy based on user identification (ID). **2.** Require that client systems authenticate themselves to servers, but trust the client system concerning the identity of its user. **3.** Require the user to prove identity for each service invoked. Also require that servers prove their identity to clients.

15.7 **Secure:** A network eavesdropper should not be able to obtain the necessary information to impersonate a user. More generally, Kerberos should be strong enough that a potential opponent does not find it to be the weak link. **Reliable:** For all services that rely on Kerberos for access control, lack of availability of the Kerberos service means lack of availability of the supported services. Hence, Kerberos should be highly reliable and should employ a distributed server architecture, with one system able to back up another. **Transparent:** Ideally, the user should not be aware that authentication is taking place, beyond the requirement to enter a password. **Scalable:** The system should be capable of supporting large numbers of clients and servers. This suggests a modular, distributed architecture.

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

15.9 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.

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

Answers to Problems

15.1 It is not so much a protection against an attack as a protection against error. Since N_a is not unique across the network, it is possible for B to mistakenly send message 6 to some other party that would accept N_a.

15.2

(1) A → B: $ID_A \| N_a$
(2) B → KDC: $ID_A \| ID_B \| N_a \| N_b$
(3) KDC → B: $\mathrm{E}(PR_{auth}, [ID_A \| PU_a]) \| \mathrm{E}(PU_b, \mathrm{E}(PR_{auth}, [N_a \| N_b \| K_s \| ID_A \| ID_B]))$
(4) B → A: $\mathrm{E}(PU_a, \mathrm{E}(PR_{auth}, [N_a \| N_b \| K_s \| ID_A \| ID_B]))$
(5) A → B: $\mathrm{E}(K_s, N_b)$

15.3 **a.** An unintentionally postdated message (message with a clock time that is in the future with respect to the recipient's clock) that requests a key is sent by a client. An adversary blocks this request message from reaching the KDC. The client gets no response and thinks that an omission or performance failure has occurred. Later, when the client is off-line, the adversary replays the suppressed message from the same workstation (with the same network address) and establishes a secure connection in the client's name.

b. An unintentionally postdated message that requests a stock purchase could be suppressed and replayed later, resulting in a stock purchase when the stock price had already changed significantly.

15.4 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.

15.5 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.

15.6 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.

15.7 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 → B: $A \{r_B, B\}$

15.8 **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.

15.9 **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).

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

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

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

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

Chapter 16 Network Access Control and Cloud Security

Answers to Questions

16.1 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).

16.2 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.

16.3 **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.

16.4 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.

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

16.6 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.

16.7 **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.

16.8 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.

16.9 **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

16.2 **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 17 TRANSPORT-LEVEL SECURITY

ANSWERS TO QUESTIONS

17.1 The advantage of using **IPSec** (Figure 17.1a) 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 17.1c) is that the service can be tailored to the specific needs of a given application.

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

17.3 **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.

17.4 **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.

17.5 **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 $2^{64} – 1$.

17.6 **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).

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

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

17.9 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.

17.10 **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

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

17.2 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.

17.3 **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.

17.4 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).

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

CHAPTER 18 WIRELESS NETWORK SECURITY

ANSWERS TO QUESTIONS

18.1 Basic service set.

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

18.3 **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.

18.4 It may or may not be.

18.5 **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.

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

18.7 **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.

18.8 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

18.1 **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.

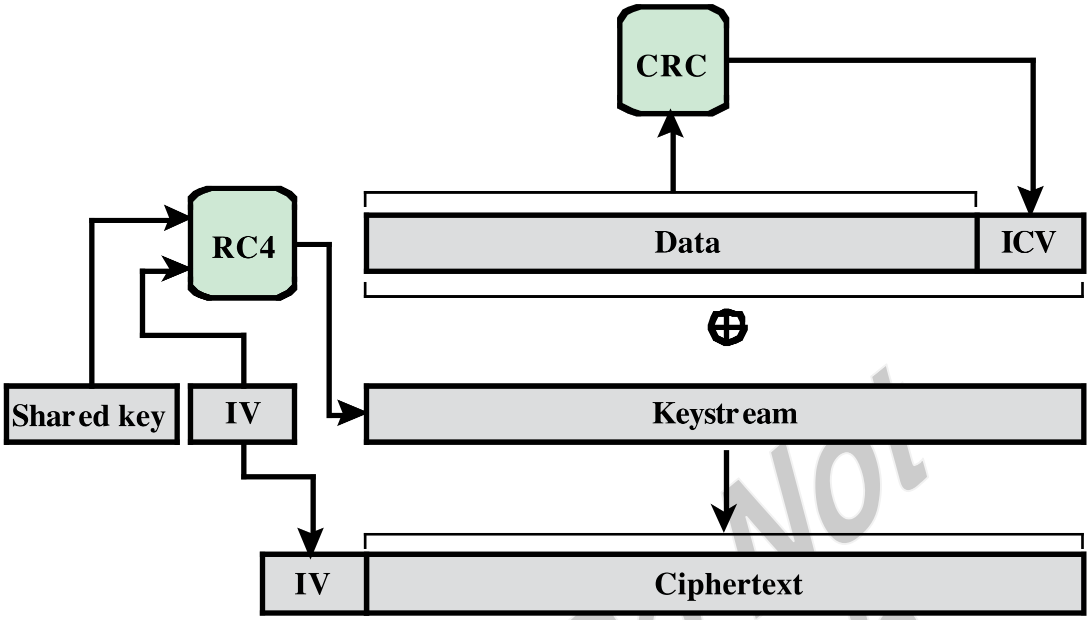
b. This scheme depends on all parties behaving honestly. The scheme does not protect against MAC address forgery.

18.2 **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.

18.3 a.

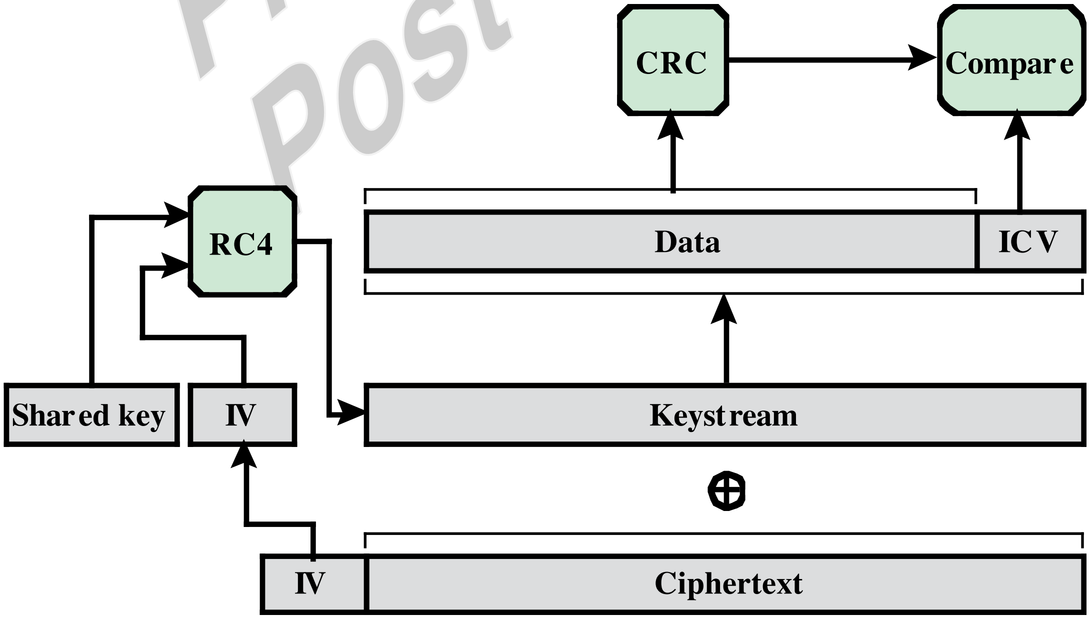


b. **1.** The IV value, which is received in plaintext, is concatenated with the WEP key shared by transmitter and receiver to form the seed, or key input, to RC4.

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.



18.4 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 19 Electronic Mail Security

Answers to Questions

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

19.2 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.

19.3 **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.

19.4 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.

19.5 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.

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

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

19.8 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.

19.9 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.

19.10 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

19.1 CFB avoids the need to add and strip padding.

19.2 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

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

19.3 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:

$$P\left(2^{64},N\right)=1-\frac{2^{64}!}{\left(2^{64}-N\right)!2^{64\times k}}$$
$$>1-e^{-[N\times(N-1)]/2^{65}}$$

19.4 **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}.

19.5 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.

19.6 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.

19.7 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.

19.8 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.

19.9 **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 20 IP SECURITY

ANSWERS TO QUESTIONS

20.1 **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.

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

20.3 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 20.3 (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 20.3 (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).

20.4 **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.

20.5 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.

20.6 **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.

20.7 **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

20.1 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.

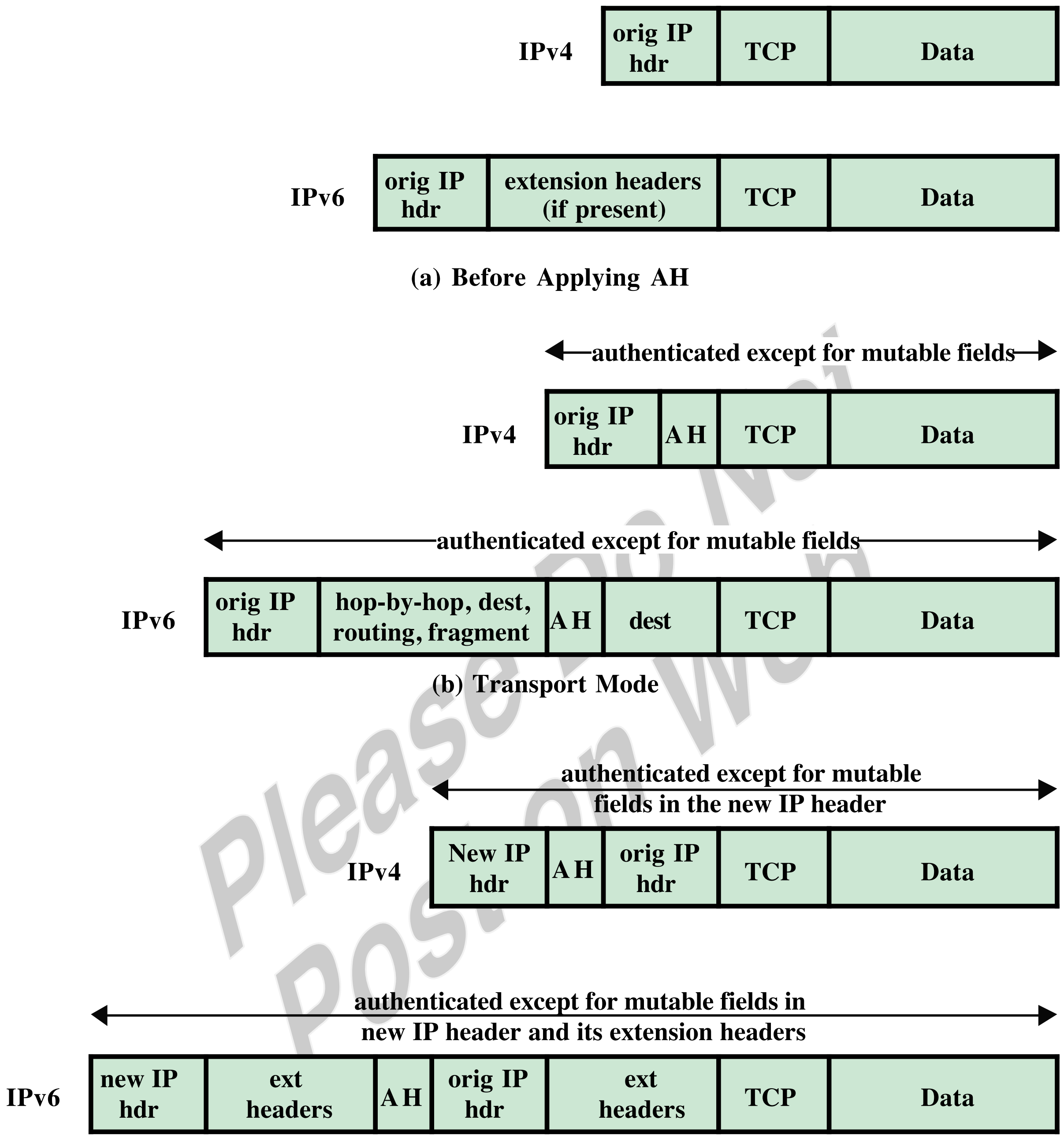
row 4: TCP traffic between this host (1.2.3.101) and the server (1.2.4.10) on server port 80 is ESP protected.

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.

20.2.


IPv4
orig IP hdr
TCP
Data
IPv6
orig IP hdr
extension headers (if present)
TCP
Data
authenticated except for mutable fields
IPv4
orig IP hdr
A H
TCP
Data
authenticated except for mutable fields
IPv6
orig IP hdr
hop-by-hop, dest, routing, fragment
A H
dest
TCP
Data
authenticated except for mutable fields in the new IP header
IPv4
New IP hdr
A H
orig IP hdr
TCP
Data
authenticated except for mutable fields in new IP header and its extension headers
IPv6
new IP hdr
ext headers
A H
orig IP hdr
ext headers
TCP
Data

(a) Before Applying AH

(b) Transport Mode

(c) Tunnel Mode

20.3 **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.

20.4 **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.

20.5 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.

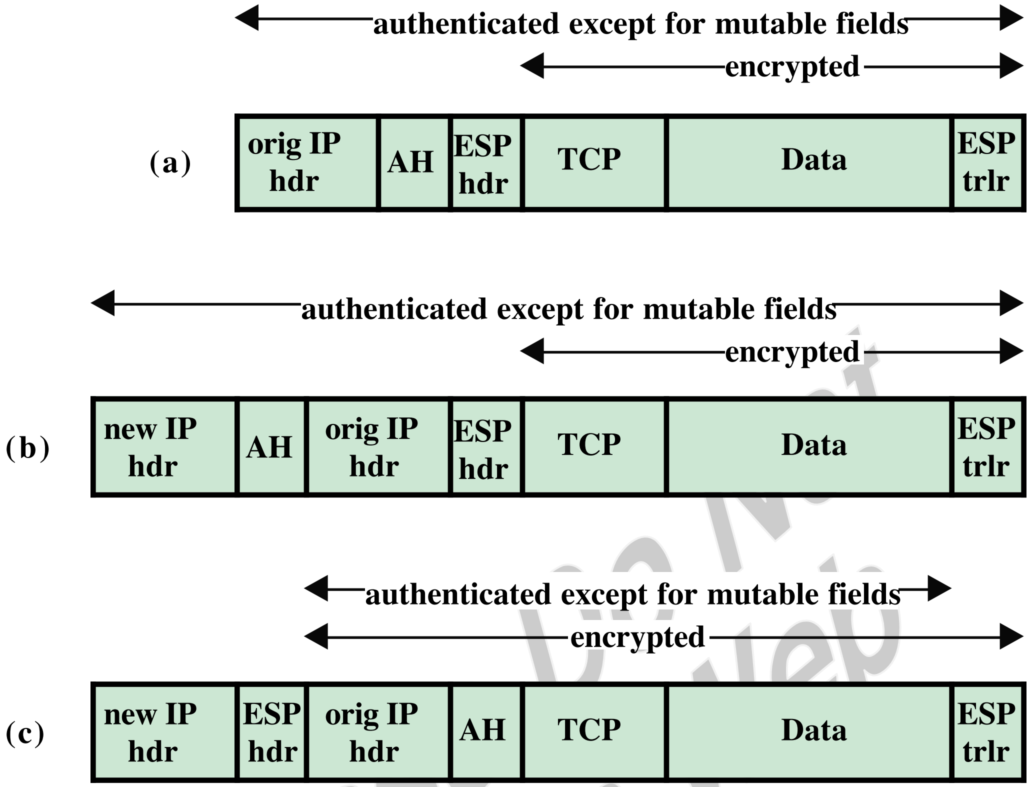
20.6 From RFC 2401

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.

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



20.8 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.

20.9 The Initial Exchanges and the CREATE_CHILD_SA Exchange

20.10 It is an addition to the IP layer.

Chapter 21 Malicious Software

Answers to Questions

21.1 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.

21.2 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.

21.3 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).

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

21.5 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.

21.6 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.

21.7 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.

21.8 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.

21.9 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.

21.10 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.

21.11 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.

21.12 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.

21.13 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.

21.14 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

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

21.2 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.

21.3 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.

21.4 **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.

21.5 Logic bomb.

21.6 Backdoor.

21.7 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.

21.8 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.

21.9 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.

21.10 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.

21.11 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.

21.12 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.

21.13 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.

21.14 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 22 Intruders

Answers to Questions

22.1 **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.

22.2 **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.

22.3 **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.

22.4 **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.

22.5 **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.

22.6 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.

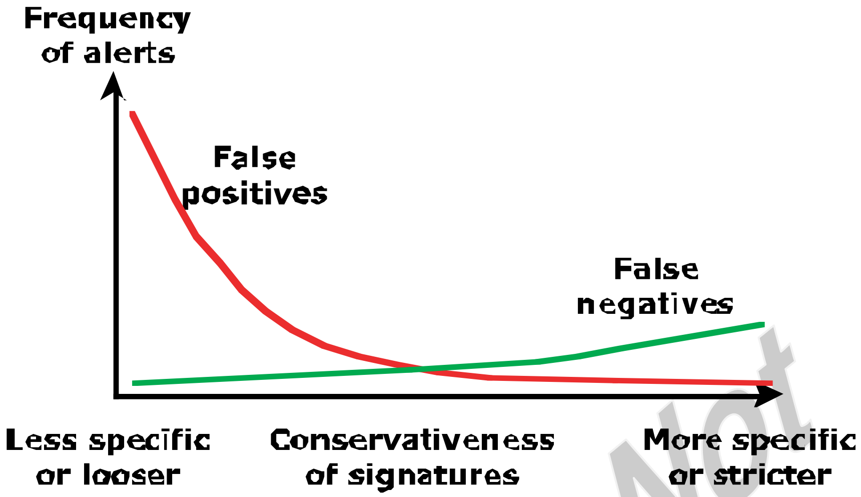
22.7 Honeypots are decoy systems that are designed to lure a potential attacker away from critical systems.

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

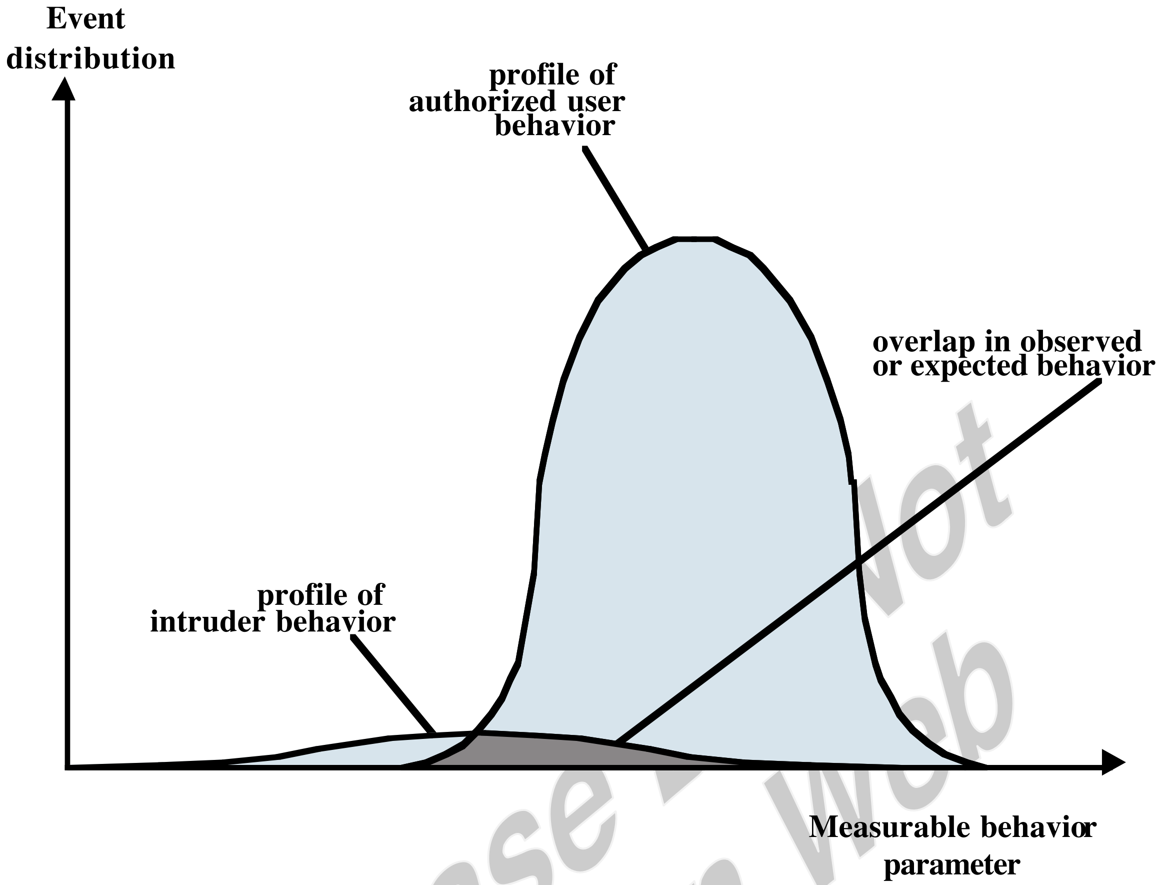
22.9 **User education:** Users can be told the importance of using hard-to-guess passwords and can be provided with guidelines for selecting strong passwords. **Computer-generated passwords:** Users are provided passwords generated by a computer algorithm. **Reactive password checking:** the system periodically 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

22.1 This is a typical example:



22.2 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 22.1, 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.


Event distribution
profile of authorized user behavior
overlap in observed or expected behavior
profile of intruder behavior
Measurable behavior parameter

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.

22.3 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).

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

$$\Pr\left[\text{Blue}/\text{WB}\right] = \frac{\Pr\left[\text{WB}/\text{Blue}\right]\Pr\left[\text{Blue}\right]}{\Pr\left[\text{WB}/\text{Blue}\right]\Pr\left[\text{Blue}\right] + \Pr\left[\text{WB}/\text{Green}\right]\Pr\left[\text{Green}\right]}$$

$$= \frac{(0.8)(0.15)}{(0.8)(0.15) + (0.2)(0.85)} = 0.41$$

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

22.5 **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.

22.6 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].

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

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

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

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

$$\frac{6 \times 10^{19} \text{passwords}}{6.4 \times 10^{6} \text{ passwords / second}} = 9.4 \times 10^{12} \text{seconds}$$

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

22.11 **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.

22.12 Yes.

22.13 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.

22.14 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.

22.15 **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:

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

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 $(1 - x) \approx e^{-x}$.

22.16 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 23 Firewalls

Answers to Questions

23.1 **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.

23.2 **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.

23.3 **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.

23.4 **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.

23.5 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 23.2. 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

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

23.7 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.

23.8 **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 23.2. 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 23.1d). 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.

23.9 • 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.

23.10 • 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.

23.11 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.

23.12 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

23.1 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.

23.2 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.

23.3 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.

23.4 **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.

23.5 **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 1023.

23.6 **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)

23.7 **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

23.8 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.

23.9 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.

23.10 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 24 Legal and Ethical Aspects

Answers to Questions

24.1
- 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.

24.2
- 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.

24.3
- 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.

24.4 (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.

24.5
- 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

24.6 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.

24.7 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.

24.8
- 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.

24.9
- 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.

24.10

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

24.1 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.

24.2 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.

24.3 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 24.2.

24.4 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.

24.5 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 24.3 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.

24.6 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.

24.7 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 24.3, and should also refer to any relevant privacy legislation that applies to the chosen organization.

24.8 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 24.3 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.

24.9 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

24.10 **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.

24.11 **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.

24.12 **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.