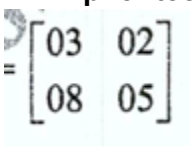
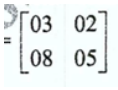
CMR INSTITUTE OF TECHNOLOGY
DEPARTMENT OF CSE
VTU QP SOLUTION
BCS702-CRYPTOGRAPHY & NETWORK SECURITY

MODULE - 1					
1	a.	**Obtain Ciphertext for the given plain text "HILLCIPHER" by applying the Hill Cipher technique using key K**  **SOLUTION** Hill Cipher is a classical polygraphic substitution cipher based on linear algebra. It encrypts blocks of letters using matrix multiplication modulo 26 (since there are 26 letters in the English alphabet). Given Plaintext: HILLCIPHER Key Matrix (K):  Alphabet Mapping: A=0, B=1, …, Z=25 Split Plaintext into Pairs (H,I) (L,L) (C,I), (P,H), (E,R) These letters are converted into numbers: (7,8), (11,11), (2,8), (15,7), (4,17) The alphabet are encrypted using the Encryption Formula $C=K\times P \bmod 26$ Pair Calculations HI → (7,8) $C1=(3\times7+2\times8)=21+16=37 \bmod 26=11$	7	L3	CO1

		C2=(8×7+5×8)=56+40=96mod 26=18 (11,18) → L S LL → (11,11) C1=(33+22)=55mod26=3 C2=(88+55)=143mod 26=13 (3,13) → D N CI → (2,8) C1=(6+16)=22mod 26=22 C_1 = (6+16)=22 \mod 26 = 22 C2=(16+40)=56mod 26=4 (22,4) → W E PH → (15,7) C1=(45+14)=59mod 26=7 C2=(120+35)=155mod 26=25 (7,25) → H Z ER → (4,17) C1=(12+34)=46mod26=20 C2=(32+85)=117mod 26=13 (20,13) → U N Final Ciphertext: LSDNWEHZUN			
	b.	**Write a short note on Steganography and its advantages and disadvantages.** **SOLUTION** Steganography is the art and science of hiding secret information inside another ordinary medium so that the existence of the message itself is concealed. Unlike cryptography, which scrambles the content, steganography hides the message within	**6**	**L2**	**CO1**

		files such as images, audio, video, or text. **Example**: Hiding a secret message inside the pixels of an image. A typical steganography system involves: 1. Cover medium – the original file (image/audio/video). 2. Secret message – the data to be hidden. 3. Stego key (optional) – a key used for embedding. 4. Stego object – the output file containing the hidden message. Types of Steganography ● Image steganography – hides data in image pixels. ● Audio steganography – embeds data in sound files. ● Video steganography – hides data in video frames. ● Text steganography – uses spacing, formatting, or character patterns. Advantages 1. High secrecy – hides the existence of communication. 2. Less suspicion – looks like a normal file. 3. Large storage capacity – multimedia files can hold significant hidden data. 4. Can be combined with cryptography for extra security. Disadvantages 1. Vulnerable to detection using steganalysis tools. 2. Data loss risk if the file is compressed or edited. 3. Limited capacity depending on the cover medium. 4. Requires careful implementation to avoid noticeable changes.			
	c.	**Write a neat diagram, explain the model for network security** **SOLUTION**	**7**	**L3**	**CO1**

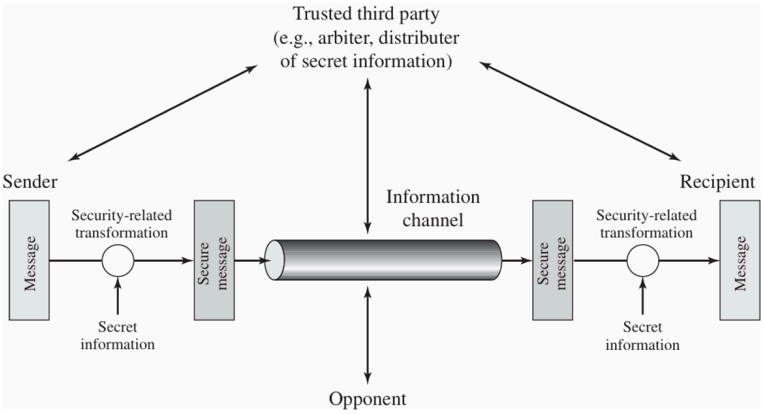


Figure 1.4 Model for Network Security

A model for much of what we will be discussing is captured, in very general terms, in Figure 1.5. A message is to be transferred from one party to another across some sort of Internet service. The two parties, who are the *principals* in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the Internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.

Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components:

- A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender.
- Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.[6]

This general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.
2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.
4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

OR

2	**a.**	**State the rules used for encryption in PLAYFAIR cipher and encrypt the message "COMPUTER" using the keyword "ENGINEERING" using PLAYFAIR CIPHER** **SOLUTION**	**7**	**L3**	**CO1**

Rules used for Encryption

•The matrix is constructed by filling in the letters of keyword (No duplicates).

•From left to right and from top to bottom.

•Then fill the remainder of the matrix with remaining letters in alphabetical order.

•Letter I/J are counted as one letter.

Plain text is encrypted two letters at a time with the following rules:

•Repeating letters in the plain text are separated with filler such as X.

•If both letters of plain text are in same row, then it is replaced by letter to right.

•If both letters of plain text are in same column, then it is replaced by letter beneath it.

•Otherwise, each plain text letter in a pair is replaced by the letter's row and the column of other letter.

Matrix Construction

E	N	G	I	R
A	B	C	D	F
H	K	L	M	O
P	Q	S	T	U
V	W	X	Y	Z

prepare Plaintext by Split into pairs:

CO | MP | UT | ER

(No repeated letters in a pair, no padding needed)

Encrypt Each Pair
CO

C → (Row2, Col3)
O → (Row3, Col5)

Rectangle rule →
C → F
O → L

→ FL

MP

		M → (Row3, Col4) P → (Row4, Col1) Rectangle rule → M → H P → T → HT UT U → (Row4, Col5) T → (Row4, Col4) Same row → move right U → P T → U → PU ER E → (Row1, Col1) R → (Row1, Col5) Same row → move right E → N R → E → NE Final Ciphertext FLHTPUNE			
	b.	**Describe simple XOR and one-time pad encryption techniques with an example and their difficulties** **SOLUTION** XOR works like: $C = P \oplus K$ $P = C \oplus K$	**7**	**L2**	**CO1**

Where:

- P = plaintext bits
- K = key bits
- C = cipher bits

Example

P = 1010
K = 1100
C = P XOR K = 0110

Improvement of Verman cipher, proposed by Army Signal Corp Officer Joseph Mauborgne. Suggested to use a key that is as long as the plain text. Key is not repeated. Key is used to encrypt and decrypt the message for one time and then it is discarded. Each message requires a new key of the same length as the new message. Works same as the Vigenère cipher with long key [modulo 26]. If cryptanalyst finds two keys, the two possible plain text can be produced.

```
ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
```

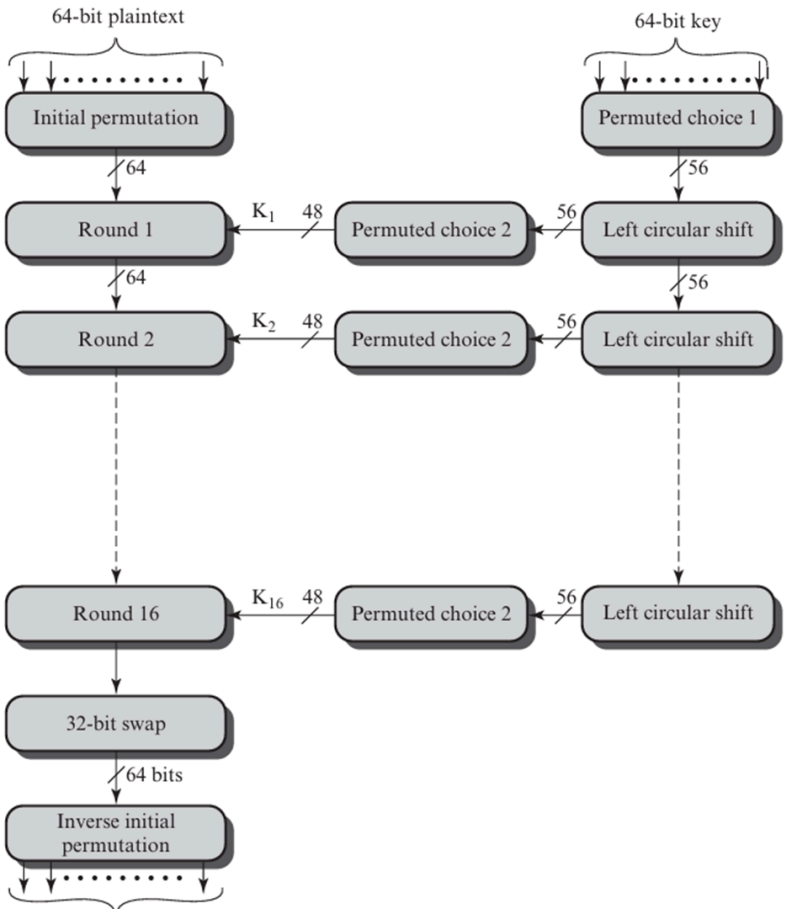
We now show two different decryptions using two different keys:

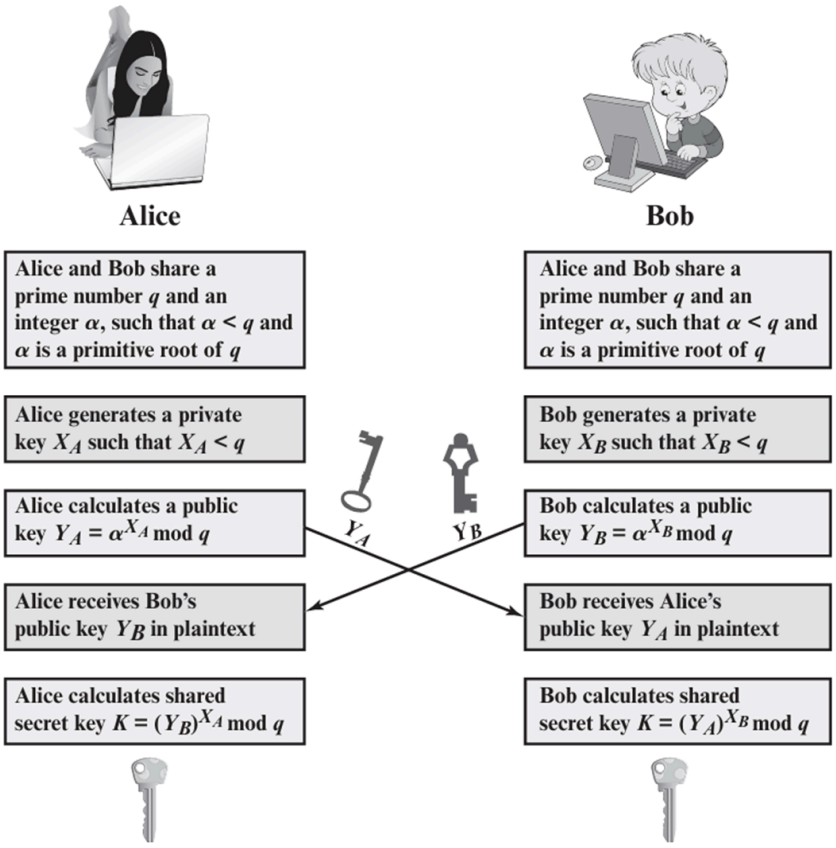
```
ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
key:        pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih
plaintext:  mr mustard with the candlestick in the hall

ciphertext: ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
key:        pftgpmiydgaxgoufhkllmhsqdqogtewbqfgyovuhwt
plaintext:  miss scarlet with the knife in the library
```

Producing large quantities of random keys can be difficult. Key distribution and protecting is also difficult.

	c.	**With a block diagram, explain the various steps involved in encryption and key generation of the DES algorithm.** **SOLUTION** Data Encryption Standard (DES) was the most widely used encryption scheme. DES was issued in 1977 by the National Bureau of Standards, now the	**6**	**L2**	**CO1**

		National Institute of Standards and Technology (NIST), as Federal Information Processing Standard 46 (FIPS PUB 46). The algorithm itself is referred to as the Data Encryption Algorithm (DEA). For DEA, data are encrypted in 64-bit blocks using a 56-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption. Over the years, DES became the dominant symmetric encryption algorithm, especially in financial applications. In 1994, NIST reaffirmed DES for federal use for another five years. NIST recommended the use of DES for applications other than the protection of classified information. There are two inputs to the encryption function: The plaintext to be encrypted . The key. The plaintext must be 64 bits in length and the key is 56 bits in length.  			
MODULE -2					
3	**a.**	**Demonstrate the Diffie-Hellman key exchange algorithm** **SOLUTION**	**8**	**L2**	**CO2**



	b.	**Perform encryption and decryption using the RSA algorithm given public key is 6 for two prime numbers 17 and 31 with message 3** **SOLUTION** Prime numbers: $p = 17, q = 31$ Public exponent: $e = 6$ Message: $M = 3$	**7**	**L3**	**CO2**

		$n = p \times q = 17 \times 31 = 527$ $\phi(n) = (p-1)(q-1) = 16 \times 30 = 480$ For RSA, e must be **coprime** with $\phi(n)$. $\gcd(6, 480) = 6 \neq 1$ Since gcd ≠ 1, **e = 6 is not valid** for RSA.			
	c.	**Describe the fundamental requirements that a public key cryptosystem must meet to ensure security** **SOLUTION** A public-key encryption scheme has six ingredients: Plaintext: The message or data that is fed into the algorithm as input. Encryption Algorithm: The algorithm that performs various transformations on the plain text. Public and private keys: Pair of keys that have been selected for encryption and decryption. Cipher text: The encrypted message produced as output. Depends on plain text and key. Decryption Algorithm: Accepts the cipher text and key to produce the plain text	**5**	**L2**	**CO2**
OR					
4	**a.**	**Explain briefly the elliptic curve cryptography and mention two applications** **SOLUTION** ECC makes use of Elliptic curves for performing encryption and decryption. Variables and coefficients are all restricted to the elements of the finite field. Offers equal security for smaller key size when compared to RSA there by reducing the processing overhead.	**6**	**L2**	**CO2**

Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A \times G$

User B Key Generation

Select private n_B	$n_B < n$
Calculate public P_B	$P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$

Figure 10.7 ECC Diffie–Hellman Key Exchange

	b.	**Let q=719 and g=5, Xa = 157, Xb = 293. Use the Diffie Hellman Key exchange algorithm to find Ya, Yb and Secret Key K.**	**8**	**L2**	**CO2**

SOLUTION

Given $q = 719$, $g = 5$, $X_a = 157$, and $X_b = 293$.

First compute the public keys.

$$Y_a = g^{X_a} \mod q = 5^{157} \mod 719 = 28$$

$$Y_b = g^{X_b} \mod q = 5^{293} \mod 719 = 279$$

Now compute the shared secret key.

Secret key computed by A:

$$K = Y_b^{X_a} \mod q = 279^{157} \mod 719 = 618$$

		Secret key computed by B: $K = Y_a^{X_b} \mod q = 28^{293} \mod 719 = 618$ Since both values are equal, the shared secret key is: $K = 618$ Thus, $Y_a = 28$, $Y_b = 279$, and the Secret Key $K = 618$.			
	c.	**Briefly explain the security aspects of the RSA algorithm** **SOLUTION** Five possible approaches to attacking the RSA Brute force: This involves trying all possible private keys. Mathematical attacks: Factoring the product of two primes. Timing attacks: Depends on the running time of the decryption algorithm. Hardware fault-based attack: This involves inducing hardware faults in the processor that is generating digital signatures. Chosen ciphertext attacks: This type of attack exploits properties of the RSA algorithm.	**5**	**L2**	**CO2**
		MODULE - 03			
5	**a.**	**Explain the symmetric key distribution using Asymmetric Encryption** **SOLUTION** 1. A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a message to B consisting of PU_a and an identifier of A, ID_A. 2. B generates a secret key, K_s, and transmits it to A, which is encrypted with A's public key. 3. A computes D(PR_a, E(PU_a, K_s)) to recover the secret key. Because only A can decrypt the message, only A and B will know the identity of K_s. 4. A discards PU_a and PR_a and B discards PU_a.	**7**	**L2**	**CO3**

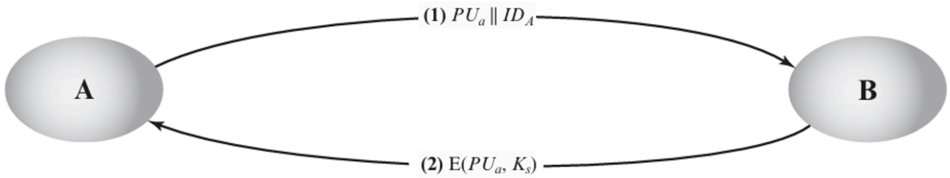


Figure 14.7 Simple Use of Public-Key Encryption to Establish a Session Key

	b.	**Explain the role of cryptographic hash functions in message authentication with a neat diagram** **SOLUTION**	**8**	**L2**	**CO3**

a. The message plus concatenated hash code is encrypted using symmetric encryption. Because only A and B share the secret key, the message must have come from A and has not been altered. The hash code provides the structure or redundancy required to achieve authentication. Because encryption is applied to the entire message plus hash code, confidentiality is also provided.

b. Only the hash code is encrypted, using symmetric encryption. This reduces the processing burden for those applications that do not require confidentiality.

c. It is possible to use 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. Because the secret value itself is not sent, an opponent cannot modify an intercepted message and cannot generate a false message.

d. Confidentiality can be added to the approach of method (c) by encrypting the entire message plus the hash code.

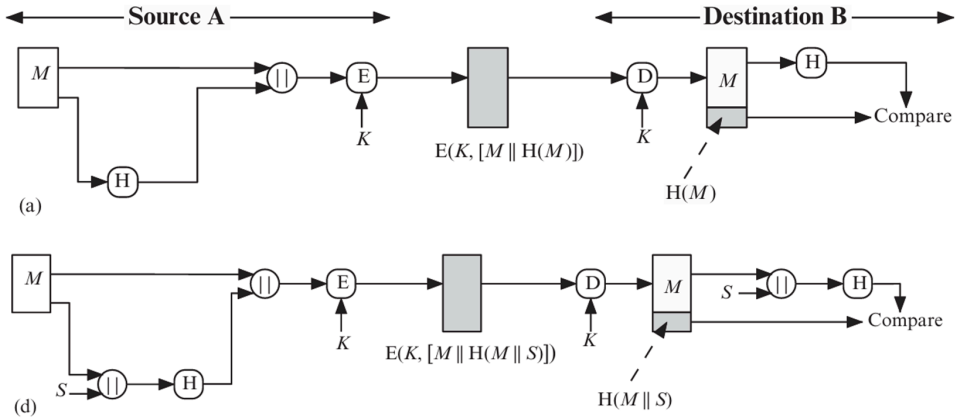
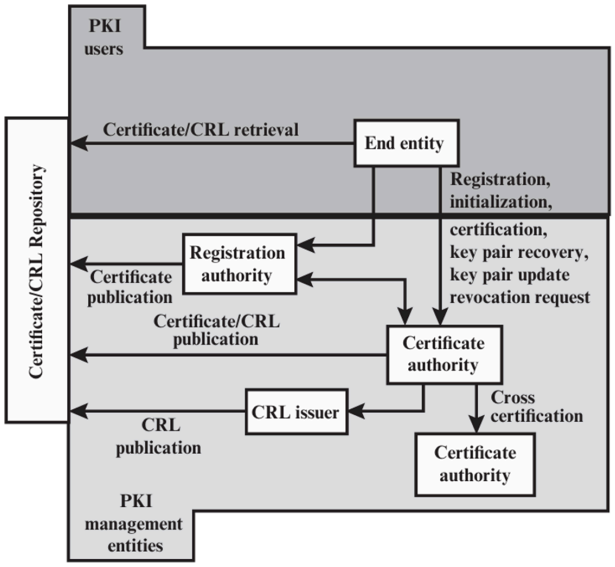


Figure 11.3 Simplified Examples of the Use of a Hash Function for Message Authentication

	c.	**Discuss the general elements of an X.509 certificate** **SOLUTION** ■ **Version:** Differentiates among successive versions of the certificate format; the default is version 1. If the *issuer unique identifier* or *subject unique identifier* are present, the value must be version 2. If one or more extensions are present, the version must be version 3. Although the X.509 specification is currently at version 7, no changes have been made to the fields that make up the certificate since version 3. ■ **Serial number:** An integer value unique within the issuing CA that is unambiguously associated with this certificate. ■ **Signature algorithm identifier:** The algorithm used to sign the certificate together with any associated parameters. Because this information is repeated in the signature field at the end of the certificate, this field has little, if any, utility. ■ **Issuer name:** X.500 name of the CA that created and signed this certificate. ■ **Period of validity:** Consists of two dates: the first and last on which the certificate is valid. ■ **Subject name:** The name of the user to whom this certificate refers. That is, this certificate certifies the public key of the subject who holds the corresponding private key. ■ **Subject's public-key information:** The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters. ■ **Issuer unique identifier:** An optional-bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities. ■ **Subject unique identifier:** An optional-bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities. ■ **Extensions:** A set of one or more extension fields. Extensions were added in version 3 and are discussed later in this section. ■ **Signature:** Covers all of the other fields of the certificate. One component of this field is the digital signature applied to the other fields of the certificate. This field includes the signature algorithm identifier.	**5**	**L2**	**CO3**
		OR			
6	**a.**	**What is Key Management? Explain with a neat diagram, how key usage can be controlled in encryption and decryption using control vectors.** **SOLUTION** **K**ey Management is the process of securely handling cryptographic keys throughout their entire lifecycle, including their generation, distribution, storage, usage, backup, and destruction. The strength of any encryption system depends not only on the algorithm used but also on how securely the	**6**	**L2**	**CO2**

		keys are managed. If keys are exposed, misused, or improperly stored, the entire security system can fail even if strong encryption algorithms are used. A key aspect of key management is controlling how a key is used. This is achieved through the use of control vectors. A control vector is a predefined bit pattern associated with a cryptographic key that specifies the permitted operations for that key. It defines whether the key can be used for encryption or decryption, whether it is meant for protecting data or encrypting other keys, whether it can be exported outside the system, and which algorithms it supports. The control vector is securely bound to the key so that the key cannot be used for any purpose other than what is authorized. When a key is generated, it is combined with its control vector and encrypted using a master key before being stored. This ensures that both the key and its usage restrictions remain protected. When a user requests an encryption or decryption operation, the system first decrypts the stored key using the master key, separates the key from its control vector, and verifies whether the requested operation is allowed. If the control vector permits the operation, the system proceeds with encryption or decryption; otherwise, the request is rejected. In this way, control vectors enforce strict usage policies and prevent unauthorized or unintended use of cryptographic keys.			
	b.	**Describe the architecture of public key infrastructure X.509 (PKIX) model with a neat diagram** **SOLUTION** The principal objective for developing a PKI is to enable secure, convenient, and efficient acquisition of public keys. The Internet Engineering Task Force (IETF) Public Key Infrastructure X.509 (PKIX) set up the model based on X.509. End entity: A generic term used to denote end users, devices. Certification authority (CA): The issuer of certificates and certificate revocation lists (CRLs). Registration authority (RA): Often associated with the end entity registration process but can assist in a number of other areas as well. CRL issuer: An optional component that a CA can delegate to publish CRLs. Repository: A generic term used to denote any method for storing certificates and CRLs so that they can be retrieved by end entities. Registration: This is the process whereby a user first makes itself known to a CA. Initialization: Before a client system can operate securely, it is necessary to install key materials. Certification: CA issues a certificate for a user's public key, returns that certificate to the user's client system, and/or posts that certificate in a repository. Key pair recovery: Key pair recovery allows end entities to restore their encryption/decryption key pair from an authorized key backup facility.	**8**	**L2**	**CO3**

		Key Pair Update: All key pairs need to be updated regularly and new certificates issued. Revocation request: An authorized person advises a CA of an abnormal situation requiring certificate revocation. Cross certification: Two CAs exchange information used in establishing a cross-certificate.  Figure 14.17 PKIX Architectural Model			
	c.	**Write a short note on the various schemes of public key distribution** **SOLUTION** Several techniques have been proposed for distribution of keys which are generalized as follows: Public announcement Publicly available directory Public-key authority Public-Key Certificates  Figure 14.10 Uncontrolled Public-Key Distribution	**5**	**L2**	**CO3**

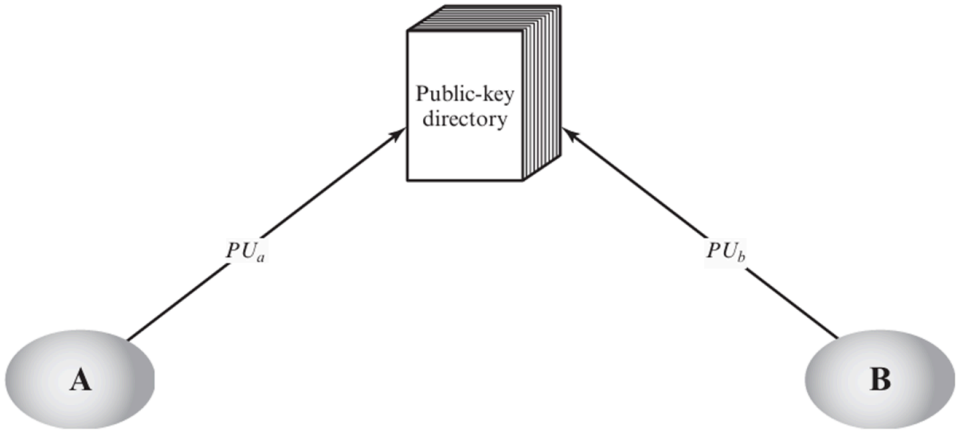


Figure 14.11 Public-Key Publication

Maintenance and distribution of the public directory is the responsibility of some trusted entity or organization .
Directory with a {name, public key} entry for each participant.
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.
A participant may replace the existing key with a new one at any time.
Participants could also access the directory electronically.

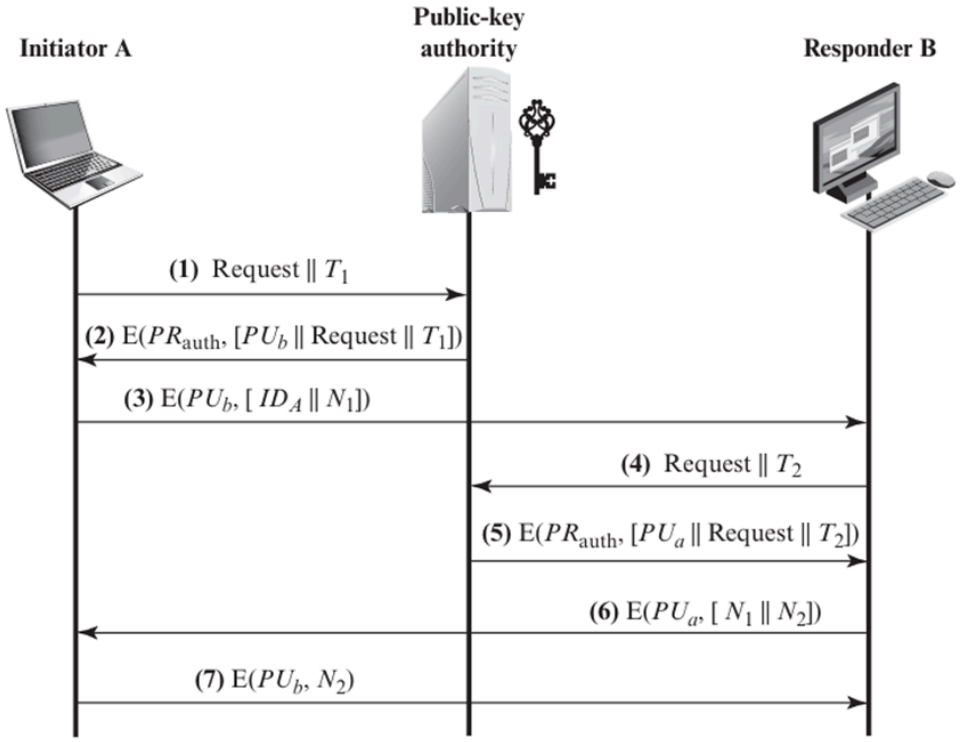
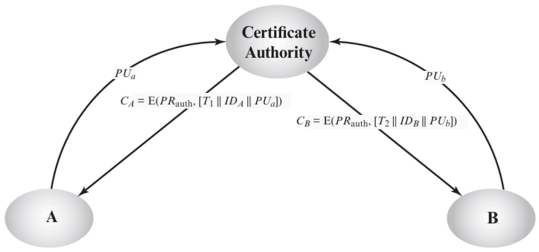
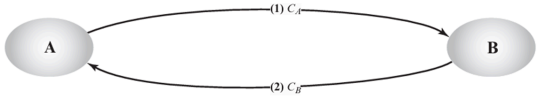


Figure 14.12 Public-Key Distribution Scenario



(a) Obtaining certificates from CA



(b) Exchanging certificates

Figure 14.13 Exchange of Public-Key Certificates

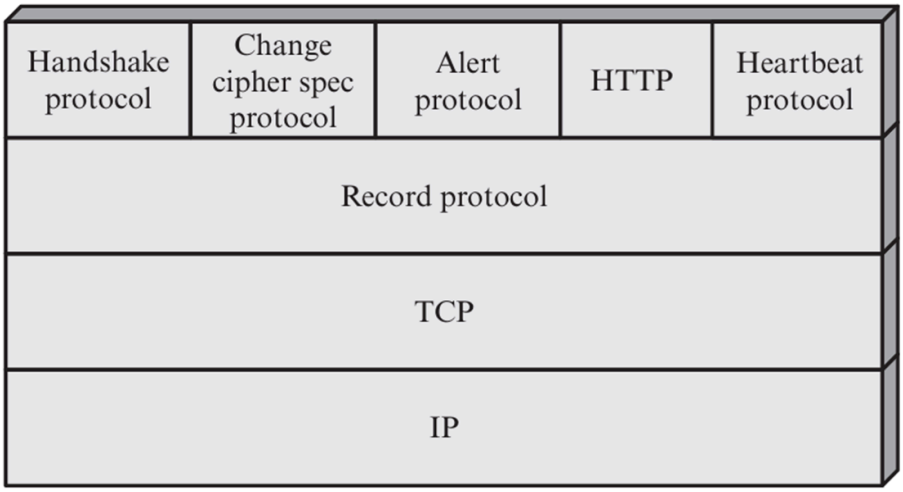
Suggested by Kohnfelder, to exchange keys without contacting a public-key authority.
A certificate consists of a public key, an identifier of the key owner, and the whole block signed by a trusted third party.
Typically, the third party is a certificate authority, such as a government agency or a financial institution.
A user can present his or her public key to the authority in a secure manner and obtain a certificate.

MODULE 4

7	**a.**	**Explain functions and cryptographic algorithms used in S/MIME functionality** **SOLUTION**	**8**	**L2**	**CO4**

Table 19.5 Cryptographic Algorithms Used in S/MIME

Function	Requirement
Create a message digest to be used in forming a digital signature.	MUST support SHA-256 SHOULD support SHA-1 Receiver SHOULD support MD5 for backward compatibility
Use message digest to form a digital signature.	MUST support RSA with SHA-256 SHOULD support —DSA with SHA-256 —RSASSA-PSS with SHA-256 —RSA with SHA-1 —DSA with SHA-1 —RSA with MD5
Encrypt session key for transmission with a message.	MUST support RSA encryption SHOULD support —RSAES-OAEP —Diffie–Hellman ephemeral-static mode
Encrypt message for transmission with a one-time session key.	MUST support AES-128 with CBC SHOULD support —AES-192 CBC and AES-256 CBC —Triple DES CBC

	b.	**Define TLS and explain its architecture with a neat diagram.** **SOLUTION** TLS is Transport layer Security. One of the most widely used security services is Transport Layer Security (TSL). (RFC 5246) TLS is an Internet standard that evolved from a commercial protocol known as Secure Sockets Layer (SSL). TLS is a general-purpose service implemented as a set of protocols that rely on TCP. Most browsers come equipped with TLS, and most Web servers have implemented the protocol.  **Figure 17.2** TLS Protocol Stack Two important TLS concepts: TLS Session : Association between a client and a server. Sessions are created by the Handshake Protocol. Define cryptographic security parameters, which can be shared among multiple connections. TLS Connection: A connection is a transport that provides a suitable type of service. Every connection is associated with one session.	**9**	**L2**	**CO4**
	c.	**Bring out the differences between Kerberos version 4 and Version 5** **SOLUTION** Encryption System Dependence: Version 4 requires the use of DES. The strength of the DES is a concern. (Ticket, Message, Key) Version 5 was redesigned to be encryption independent. It works with any symmetric encryption. Encryption keys are tagged with a type and a length, allowing the same key to be used in different algorithms.	**5**	**L2**	**CO4**

		Internet Protocol Dependence: Version 4 requires the use of Internet Protocol (IPV4). Version 5 was redesigned to be accommodate IPV6 and OSI addresses. Network addresses are tagged with type and length allowing any network address type to be used. Message byte Ordering: Version 4 sender choses a byte ordering of his/her own, which is ambiguous. Version 5 all the message structures are defined using ASN.1 (Abstract Syntax Notation One) and Basic Encoding Rules (BER). Ticket Life time: Version 4 encoded in an 8-bit quantity in units of 5 minutes. Version 5 includes an explicit start time and end time, allowing tickets with arbitrary lifetimes. Authentication Forwarding: Version 4 tickets are tied to the client's machine. It cannot be forwarded to another server. Version 5 allows forwardable tickets. The client can allow the server to temporarily act on their behalf. Inter-realm Authentication: Version 4 interoperability among N realms require N2 Kerberos-to-Kerberos relationship. Version 5 requires a fewer relationships. Double Encryption: Version 4 tickets provided are encrypted twice- once with secret key of Server and again with the secret key known to the client. It is computationally expensive. PCBC Encryption: Version 4 makes use of non-standard mode of DES known as Propagating cipher block chaining (PCBC). It is vulnerable to attacks. Version 5 provides cipher block chaining (CBC) mode for encryption. Session Keys: Version 4 singe session key is used for all types of communication. Version 5 provides a sub session key, which is used only for that one connection. A new access by the client would result in the use of a new sub session key. Password attacks:			

		Both versions are vulnerable to a password attack. Version 5 provides a mechanism called pre-authentication, making the password attacks difficult, but it does not prevent them.			
		OR			
8	**a.**	**Describe remote user authentication using asymmetric encryption** **SOLUTION** A protocol using timestamps is provided in [DENN81]: 1. A → AS: $ID_A \| ID_B$ 2. AS → A: $E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T])$ 3. A → B: $E(PR_{as}, [ID_A \| PU_a \| T]) \| E(PR_{as}, [ID_B \| PU_b \| T]) \| E(PU_b, E(PR_a, [K_s \| T]))$ In this case, the central system is referred to as an authentication server (AS), because it is not actually responsible for secret-key distribution. Rather, the AS provides public-key certificates. The session key is chosen and encrypted by A; hence, there is no risk of exposure by the AS. The timestamps protect against replays of compromised keys. This protocol is compact but, as before, requires the synchronization of clocks. Another approach, proposed by Woo and Lam [WOO92a], makes use of nonces. The protocol consists of the following steps. 1. A → KDC: $ID_A \| ID_B$ 2. KDC → A: $E(PR_{auth}, [ID_B \| PU_b])$ 3. A → B: $E(PU_b, [N_a \| ID_A])$ 4. B → KDC: $ID_A \| ID_B \| E(PU_{auth}, N_a)$ 5. KDC → B: $E(PR_{auth}, [ID_A \| PU_a]) \| E(PU_b, E(PR_{auth}, [N_a \| K_s \| ID_B]))$ 6. B → A: $E(PU_a, [E(PR_{auth}, [(N_a \| K_s \| ID_B)]) \| N_b])$ 7. A → B: $E(K_s, N_b)$	**8**	**L2**	**CO4**
	b.	**Explain Pretty Good Privacy (PGP) message transmission and reception with a neat diagram.** **SOLUTION**	**4**	**L2**	**CO3**

An alternative email security protocol is Pretty Good Privacy (PGP), which has essentially the same functionality as S/MIME. PGP was created by Phil Zimmerman and implemented as a product first released in 1991. It was made available free of charge and became quite popular for personal use. The initial PGP protocol was proprietary and used some encryption algorithms with intellectual property restrictions. In 1996, version 5.x of PGP was defined in IETF RFC 1991, *PGP Message Exchange Formats*. Subsequently, OpenPGP was developed as a new standard protocol based on PGP version 5.x. OpenPGP is defined in RFC 4880 (*OpenPGP Message Format*, November 2007) and RFC 3156 (*MIME Security with OpenPGP*, August 2001).

There are two significant differences between S/MIME and OpenPGP:

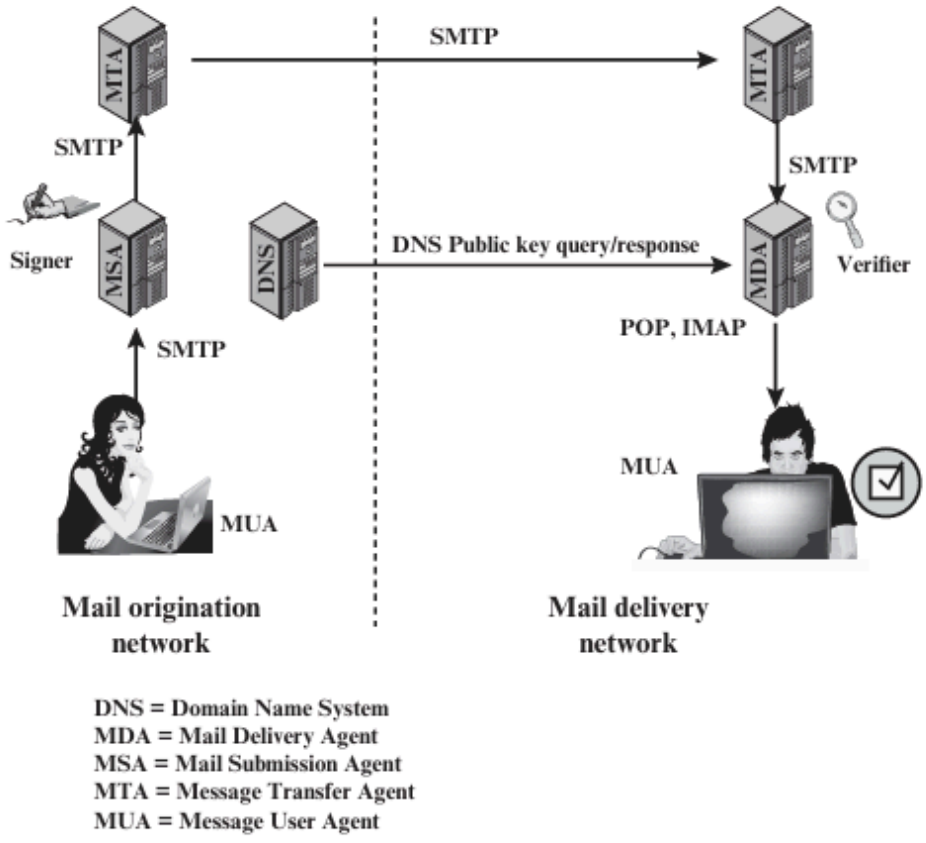
- **Key Certification:** S/MIME uses X.509 certificates that are issued by Certificate Authorities (or local agencies that have been delegated authority by a CA to issue certificates). In OpenPGP, users generate their own OpenPGP public and private keys and then solicit signatures for their public keys from individuals or organizations to which they are known. Whereas X.509 certificates are trusted if there is a valid PKIX chain to a trusted root, an OpenPGP public key is trusted if it is signed by another OpenPGP public key that is trusted by the recipient. This is called the *Web-of-Trust*.
- **Key Distribution:** OpenPGP does not include the sender's public key with each message, so it is necessary for recipients of OpenPGP messages to separately obtain the sender's public key in order to verify the message. Many organizations post OpenPGP keys on TLS-protected websites: People who wish to verify digital signatures or send these organizations encrypted mail

	c.	**Elaborate on the various security approaches that address web security threats.**	**8**	**L2**	**CO3**

SOLUTION

Table 17.1 A Comparison of Threats on the Web

	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Trojan horse browser • Modification of memory • Modification of message traffic in transit	• Loss of information • Compromise of machine • Vulnerability to all other threats	Cryptographic checksums
Confidentiality	• Eavesdropping on the net • Theft of info from server • Theft of data from client • Info about network configuration • Info about which client talks to server	• Loss of information • Loss of privacy	Encryption, Web proxies
Denial of Service	• Killing of user threads • Flooding machine with bogus requests • Filling up disk or memory • Isolating machine by DNS attacks	• Disruptive • Annoying • Prevent user from getting work done	Difficult to prevent
Authentication	• Impersonation of legitimate users • Data forgery	• Misrepresentation of user • Belief that false information is valid	Cryptographic techniques

MODULE - 5					
9	a.	**How does Domain Keys Identified Mail (DKIM) address the threats posed by email attackers and what is its strategy for email authentication?** **SOLUTION** 1. S/MIME depends on both the sending and receiving users employing S/MIME. For almost all users, the bulk of incoming mail does not use S/MIME, and the bulk of the mail the user wants to send is to recipients not using S/MIME. 2. S/MIME signs only the message content. Thus, RFC 5322 header information concerning origin can be compromised. 3. DKIM is not implemented in client programs (MUAs) and is therefore transparent to the user; the user need not take any action. 4. DKIM applies to all mail from cooperating domains. 5. DKIM allows good senders to prove that they did send a particular message and to prevent forgers from masquerading as good senders.  DNS = Domain Name System MDA = Mail Delivery Agent MSA = Mail Submission Agent MTA = Message Transfer Agent MUA = Message User Agent Figure 19.10 Simple Example of DKIM Deployment	8	L2	CO5
	b.	**Explain Internet Key Exchange (IKE) key determination features** **SOLUTION**	7	L2	CO5

Key Determination Protocol

IKE key determination is a refinement of the Diffie–Hellman key exchange algorithm. Recall that Diffie–Hellman involves the following interaction between users A and B. There is prior agreement on two global parameters: q, a large prime number; and α, a primitive root of q. A selects a random integer X_A as its private key and transmits to B its public key $Y_A = \alpha^{X_A} \bmod q$. Similarly, B selects a random integer X_B as its private key and transmits to A its public key $Y_B = \alpha^{X_B} \bmod q$. Each side can now compute the secret session key:

$$K = (Y_B)^{X_A} \bmod q = (Y_A)^{X_B} \bmod q = \alpha^{X_A X_B} \bmod q$$

The Diffie–Hellman algorithm has two attractive features:

- Secret keys are created only when needed. There is no need to store secret keys for a long period of time, exposing them to increased vulnerability.
- The exchange requires no pre-existing infrastructure other than an agreement on the global parameters.

- It does not provide any information about the identities of the parties.
- It is subject to a man-in-the-middle attack, in which a third party C impersonates B while communicating with A and impersonates A while communicating with B. Both A and B end up negotiating a key with C, which can then listen to and pass on traffic. The man-in-the-middle attack proceeds as
  1. B sends his public key Y_B in a message addressed to A (see Figure 10.2).
  2. The enemy (E) intercepts this message. E saves B's public key and sends a message to A that has B's User ID but E's public key Y_E. This message is sent in such a way that it appears as though it was sent from B's host system. A receives E's message and stores E's public key with B's User ID. Similarly, E sends a message to B with E's public key, purporting to come from A.
  3. B computes a secret key K_1 based on B's private key and Y_E. A computes a secret key K_2 based on A's private key and Y_E. E computes K_1 using E's secret key X_E and Y_B and computers K_2 using X_E and Y_A.
  4. From now on, E is able to relay messages from A to B and from B to A, appropriately changing their encipherment en route in such a way that neither A nor B will know that they share their communication with E.
- It is computationally intensive. As a result, it is vulnerable to a clogging attack, in which an opponent requests a high number of keys. The victim spends considerable computing resources doing useless modular exponentiation rather than real work.

	c.	**Explain Basic combinations of Security Association** **SOLUTION** A Security Association bundle is used when a single SA is not sufficient to provide all the required IPsec services for a traffic flow. Since one SA can implement either AH or ESP but not both simultaneously, multiple SAs may be combined to achieve both authentication and confidentiality. In transport adjacency, two transport-mode SAs are bundled together without tunneling. Typically, an inner ESP SA performs encryption on the IP payload, and an outer AH SA provides authentication. In this case, ESP is used without its	**5**	**L2**	**CO5**

		authentication option. After encrypting the payload, AH is applied to authenticate the ESP header, encrypted data, and most parts of the original IP header except mutable fields. The advantage of this method over using ESP with its authentication option is that authentication also covers the source and destination IP addresses. However, it introduces additional overhead because two SAs are required instead of one. Another important combination is the transport–tunnel bundle, where authentication is applied before encryption. In this approach, an inner AH transport SA first authenticates the original IP packet, including the IP header (except mutable fields) and payload. Then, an outer ESP tunnel SA encrypts the entire authenticated packet and adds a new outer IP header. This method ensures that authentication data is protected by encryption, preventing attackers from modifying authentication information. It also allows the receiver to store authentication data along with the original message for later verification without needing to re-encrypt the data. Applying authentication before encryption has practical benefits. Since the authentication information is encrypted, an attacker cannot tamper with it without being detected. Additionally, if authentication applies to the original plaintext message, it becomes easier to verify integrity later. If authentication were applied after encryption, the message might need to be decrypted and reprocessed for verification. Therefore, the order of applying authentication and encryption affects both security properties and operational convenience. The IPsec architecture defines certain basic combinations of SAs that must be supported by compliant IPsec hosts and security gateways. In host-to-host communication, security can be provided directly between end systems using either AH or ESP in transport mode, or a combination where ESP is applied first and AH is applied afterward in transport mode. These combinations allow systems to support authentication only, confidentiality only, or both authentication and confidentiality. Additionally, any of these transport-mode combinations can be encapsulated inside a tunnel-mode SA for added protection across untrusted networks. In more complex network scenarios, especially when security gateways such as firewalls or routers are involved, tunnel mode is required for gateway-to-gateway protection. Logical connectivity through nested SAs may differ from physical connectivity, meaning traffic may pass through multiple security layers before reaching its destination. Each SA in the bundle can use either AH or ESP depending on the required services. These combinations ensure flexibility in providing authentication, encryption, authentication before encryption, or authentication after encryption, depending on organizational security policies and network architecture.			
		OR			
10	**a.**	**Illustrate the key components of the Internet mail architecture with a clear diagram**	**8**	**L2**	**CO5**

SOLUTION

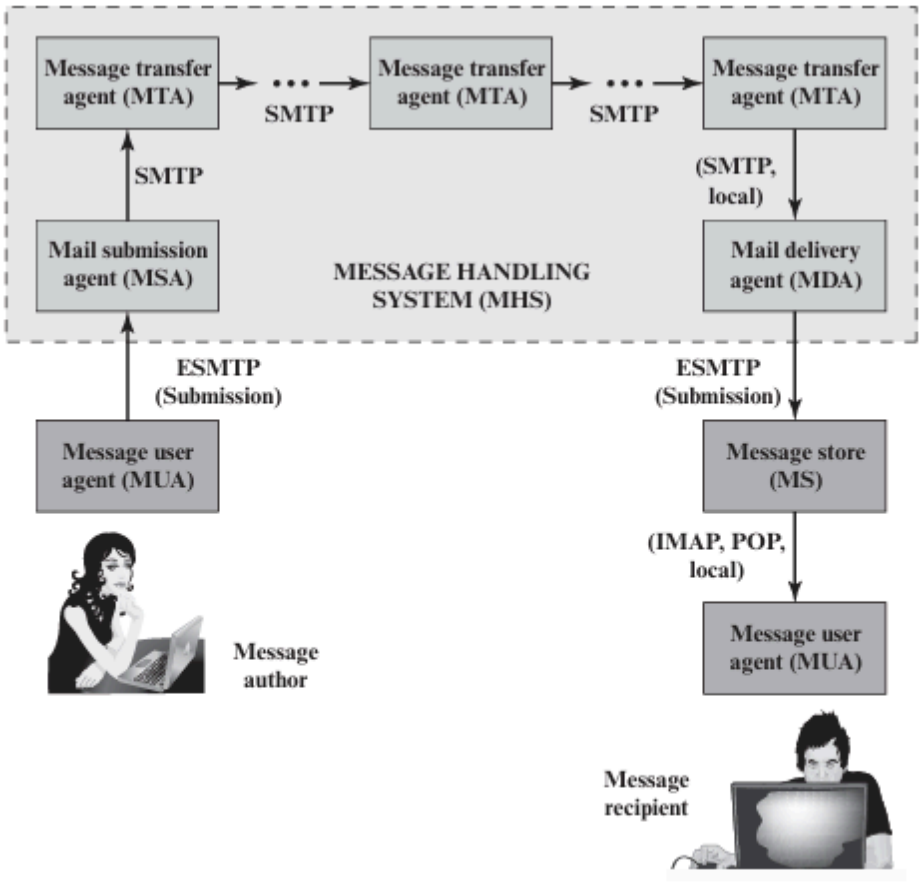


Figure 19.1 Function Modules and Standardized Protocols Used between them in the Internet Mail Architecture

At its most fundamental level, the Internet mail architecture consists of two main environments: the user world and the transfer world. The user world is represented by the Message User Agent (MUA), which operates on behalf of users and applications. It is typically an email client running on a user's computer or mobile device that allows users to compose, send, receive, and read messages. The transfer world is represented by the Message Handling Service (MHS), which is responsible for transporting messages between users. The MHS is composed of Message Transfer Agents (MTAs) that relay messages across networks, creating a virtual MUA-to-MUA communication environment. This architecture requires interoperability between MUAs for proper message formatting and display, between MUAs and the MHS during message submission and delivery, and among MTAs to ensure reliable message routing.

When a user sends an email, the author's MUA formats the message and submits it to a Mail Submission Agent (MSA), which enforces domain policies and ensures compliance with Internet standards. The MSA may reside on the

		same system as the MUA or on a separate mail server. Communication between the MUA and MSA commonly uses the Simple Mail Transfer Protocol (SMTP). After submission, the message is handled by one or more Message Transfer Agents (MTAs). Each MTA functions like a router, making routing decisions and forwarding the message toward its destination. During this process, MTAs add trace information to the message header, allowing the path of the email to be tracked. SMTP is used for communication between MTAs and also between an MTA and an MSA or Mail Delivery Agent. At the destination, a Mail Delivery Agent (MDA) transfers the message from the MHS to the recipient's Message Store (MS), where it is kept until retrieved. The Message Store may reside on a remote mail server or locally on the user's device. The recipient's MUA then accesses the stored messages for display and management. Retrieval from a remote message store typically uses Post Office Protocol (POP) or Internet Message Access Protocol (IMAP), which allow users to download or synchronize their emails. Together, these components ensure reliable message submission, transfer, storage, and retrieval within the Internet email system.			
	b.	**Explain the Encapsulating IP Security Payload** **SOLUTION** The Encapsulating Security Payload (ESP) is one of the core protocols of IPsec that provides confidentiality, data integrity, authentication, and optional anti-replay protection for IP packets. Unlike the Authentication Header (AH), which only provides integrity and authentication, ESP primarily ensures confidentiality by encrypting the payload of an IP packet. ESP can operate in both transport mode and tunnel mode, making it suitable for host-to-host as well as gateway-to-gateway secure communication. In transport mode, ESP encrypts and optionally authenticates only the payload of the IP packet, leaving the original IP header unchanged. This mode is typically used for end-to-end communication between two hosts. The ESP header is inserted after the IP header and before the payload, and the ESP trailer and authentication data are appended at the end. While the payload is encrypted, the IP header remains visible, which allows routers to forward the packet normally but exposes addressing information. In tunnel mode, ESP encrypts the entire original IP packet, including its header and payload, and encapsulates it within a new IP packet with a new outer header. This mode is commonly used in virtual private networks (VPNs) and gateway-to-gateway security, where the internal network structure must be hidden. By encrypting the entire inner packet, ESP provides stronger confidentiality and protects internal IP addresses from exposure over untrusted networks. The ESP packet format consists of several fields. The ESP header includes the Security Parameters Index (SPI), which identifies the security association, and a sequence number used for anti-replay protection. The encrypted	**7**	**L2**	**CO5**

		portion contains the payload data, padding, pad length, and next header field. After encryption, optional authentication data may be appended to ensure integrity and authenticity. The combination of encryption and authentication helps protect against eavesdropping, tampering, and replay attacks. ESP is widely used in IPsec-based VPNs because it balances security and flexibility. It allows organizations to protect sensitive data across public networks while maintaining compatibility with existing IP infrastructure. By supporting multiple encryption and authentication algorithms and operating in different modes, ESP provides a versatile mechanism for securing IP communications.			
	c.	**Describe the functional flow of Domain Keys Identified Mail (DKIM)** **SOLUTION**	**5**	**L2**	**CO5**

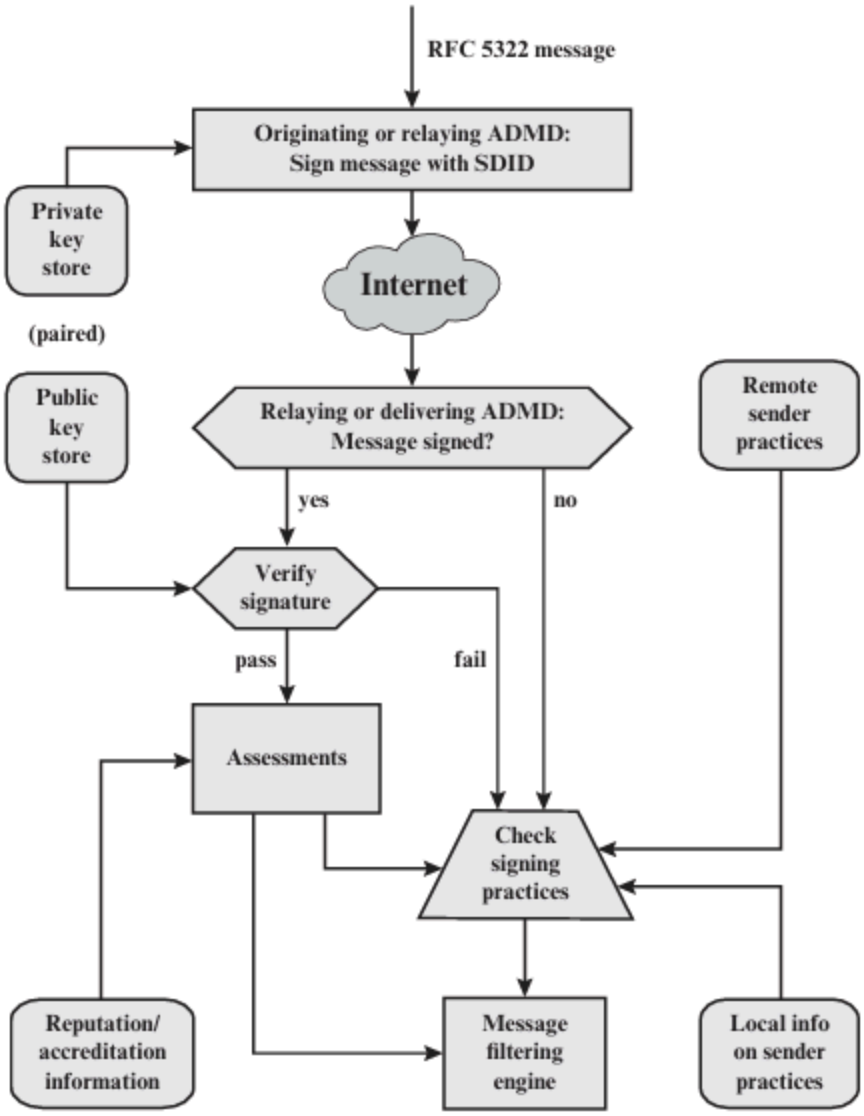


Figure 19.11 DKIM Functional Flow

Basic message processing in secure email systems using domain-based authentication mechanisms is divided between a signing Administrative Management Domain (ADMD) and a verifying ADMD. In the simplest case, the originating ADMD signs the message while the delivering ADMD verifies it, although intermediate ADMDs may also participate in the handling path. Signing is performed by an authorized module within the signing ADMD, which may reside in the MUA, MSA, or an MTA, and it uses private key information from a key store to generate a digital signature for the message. Verification is performed by an authorized module within the verifying ADMD, which may be located in an MTA, MDA, or MUA. This module checks the signature using public key information retrieved from a key store and determines whether a signature was required. If the signature is valid, reputation information about the signer is evaluated and passed to the

		message filtering system to aid in trust decisions. If the signature fails or is missing when the sender's domain is known to use signing (for example, a domain that uses DKIM but the message lacks a DKIM signature), the system may treat the message as suspicious or potentially fraudulent, and information about the domain's signing practices may be retrieved locally or remotely and supplied to the filtering system for further action.			