CBCS SCHEME

Seventh Semester B.E./B.Tech Degree Examination
Dec.2025/Jan.2026

Course: BAD703 – Data Security and Privacy

Max Marks: 100
Time: 3 Hours

Instructions:

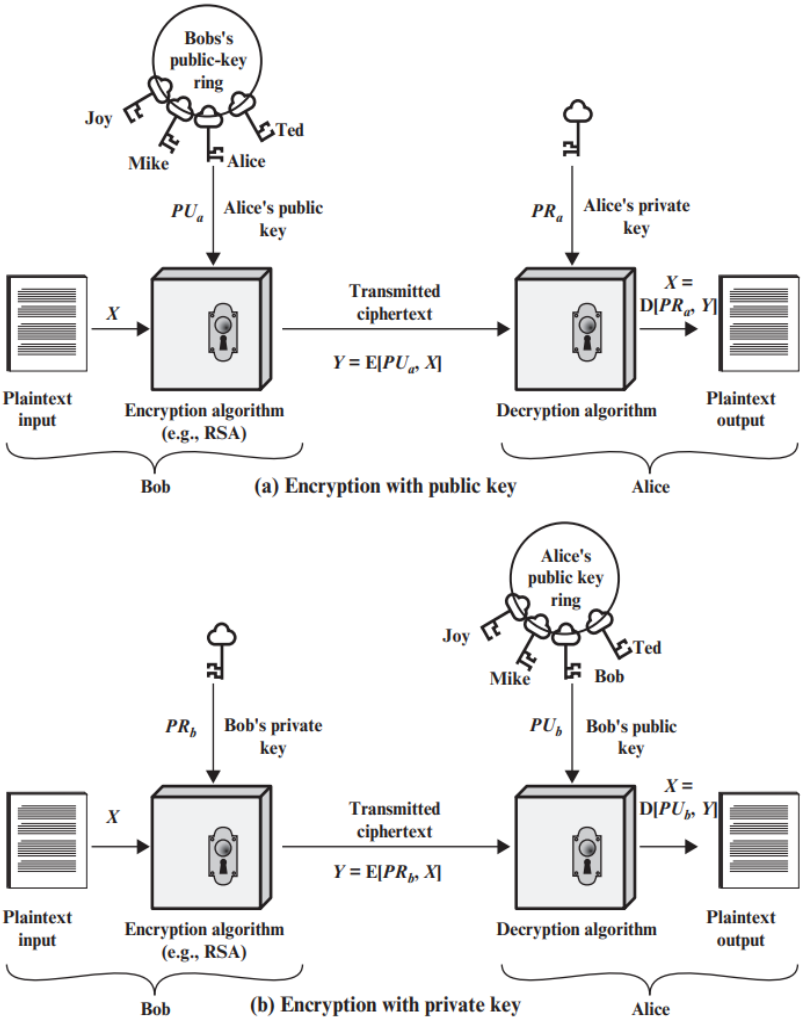
1. Answer any FIVE full questions, choosing ONE full question from each module.
2. Marks are indicated against each question.
3. L: Bloom's level, C: Course outcomes.

Module – 1

Q1. a) Explain the model for network security with a neat diagram. (10 Marks)

Network Security Model

1. **Sender** – Entity that sends the data
2. **Receiver** – Intended recipient
3. **Message** – Data to be protected
4. **Security Transformation** – Encryption or decryption algorithm
5. **Secret Key** – Used for encryption/decryption
6. **Trusted Third Party** – Assists in key management
7. **Opponent (Attacker)** – Attempts to compromise security



Working

- The sender encrypts the plaintext using a secret key.
- Encrypted data (ciphertext) is transmitted over an insecure channel.
- The receiver decrypts the ciphertext using the corresponding key.
- A trusted third party ensures secure key distribution.
- An attacker may attempt eavesdropping, modification, or fabrication.

b) What is Steganography? Describe different methods with examples. (10 Marks)

Definition

Steganography is the art and science of hiding secret information within an innocent-looking cover medium such that the **existence of the message is concealed**.

Difference from Cryptography

- Cryptography hides **content**
- Steganography hides **existence**

Methods of Steganography

1. Image Steganography

- Uses digital images as cover
- Common technique: **LSB substitution**

Example:

Original pixel : 10110110

Secret bit : 1

Modified pixel : 10110111

2. Text Steganography

- Uses formatting or linguistic features
- Methods: spacing, synonyms, innocuous text

3. Audio Steganography

- Embeds data in audio signals
- Techniques: phase coding, echo hiding

4. Video Steganography

- Hides data in video frames

Applications

- Covert communication
- Copyright protection

OR

Q2. a) Explain the Playfair cipher encryption and decryption process with an example. (10 Marks)

What is Playfair Cipher?

- It's a **cipher that encrypts letters in pairs (digrams)** instead of one letter at a time.
- Makes it **harder to break** than a simple substitution cipher because there are 676 possible diagrams.
- Uses a **5×5 matrix** of letters built from a keyword.
- **I and J share one cell** in the matrix.

Rules for Encryption

For each pair of letters in the message:

1. **Same row:** replace each letter with the one to its **right** (wrap around).
2. **Same column:** replace each letter with the one **below** (wrap around).
3. **Different row & column:** form a rectangle and replace each letter with the one in **its row and the column of the other letter**.
4. **Repeating letters:** separate them with **X**.
5. **Odd letters:** add **X** at the end.

Encrypting "KEEP DATA SAFE" with key "NETWORK"

Step 1: Build 5×5 matrix using the keyword "NETWORK"

1. Write the keyword first (remove duplicates): N E T W O R K

2. Fill the remaining letters (I/J together): A B C D F G H I/J L M P Q S U V X Y Z

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

Step 2: Prepare plaintext

- Remove spaces → **KEEPDATASAFE**
- Split into digraphs → KE | EP | DA | TA | SA | FE

Step 3: Encrypt each digraph using rules

1. **KE** → K and E are in the **same column** → take letters **below**:
    - K → F, E → K → Cipher: **FK**
2. **EP** → E and P are in **different rows and columns** → rectangle rule:
    - E → T, P → M → Cipher: **TM**
3. **DA** → D and A → rectangle → D→G, A→D → Cipher: **GD**
4. **TA** → T and A are in **same column** → T→A, A→G → Cipher: **AG**
5. **SA** → S and A → rectangle → S→Q, A→C → Cipher: **QC**
6. **FE** → F and E are in **same column** → F→K, E→E → Cipher: **KE**

Step 4: Combine ciphertext

FK TM GD AG QC KE

b) Discuss various substitution techniques used in classical encryption. (10 Marks)

Substitution techniques are classical encryption methods in which plaintext symbols are replaced by ciphertext symbols according to a predefined rule. These techniques provide confidentiality by obscuring the original message.

Types of Substitution Techniques

1. Caesar Cipher
The Caesar cipher is the simplest substitution cipher, where each letter in the plaintext is replaced by a letter a fixed number of positions down the alphabet.
Example (Shift = 3):
Plaintext : A B C D
Ciphertext: D E F G
Merits
Simple and easy to implement
Limitations
Easily broken using brute-force attack

2. Monoalphabetic Cipher
In a monoalphabetic cipher, each plaintext letter is mapped to a unique ciphertext letter.
Example Mapping:
Plaintext : A B C D
Ciphertext: Q W E R
Merits
Larger key space than Caesar cipher
Limitations
Vulnerable to frequency analysis

3. Playfair Cipher
The Playfair cipher encrypts pairs of letters (digraphs) instead of individual letters.
Steps:
Construct a 5×5 key matrix using a keyword
Divide plaintext into digraphs
Apply encryption rules based on row, column, or rectangle
Merits
Stronger than monoalphabetic ciphers
Limitations
Still vulnerable to digraph frequency analysis

4. Hill Cipher
Hill cipher is a matrix-based polygraphic substitution cipher.
Encryption Formula:
C=KPmod" " 26

Where:
K= Key matrix
P= Plaintext vector
C= Ciphertext vector
Merits
Resists simple frequency analysis
Limitations
Vulnerable if plaintext–ciphertext pairs are known

Module – 2

Q3. a) Explain the principles of public-key cryptosystem with a neat diagram. (10 Marks)

Public-key (asymmetric) cryptography was introduced by **Diffie and Hellman (1976)** to overcome two major problems of symmetric cryptography:

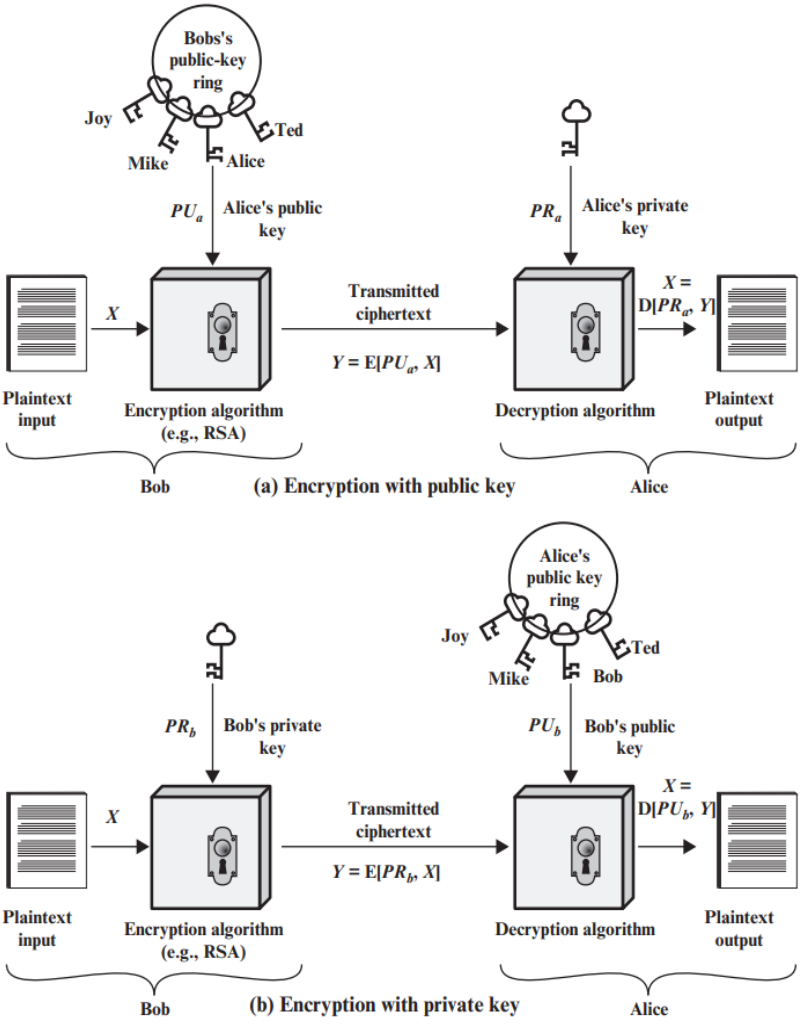
1. **Key distribution** - In symmetric systems, securely exchanging keys is difficult.
2. **Digital signatures** - Symmetric cryptography cannot provide non-repudiation or proof of origin.

Principles of Public-Key Cryptosystem

Public-key (asymmetric) cryptography was introduced by **Diffie and Hellman (1976)** to overcome two major problems of symmetric cryptography:

1. **Key distribution** – In symmetric systems, securely exchanging keys is difficult.
2. **Digital signatures** – Symmetric cryptography cannot provide non-repudiation or proof of origin.

Basic Principles



Applications of Public-Key Cryptosystem

1. **Secure Communication (Encryption/Decryption):**
    - Sender encrypts with **receiver's public key**.
    - Receiver decrypts with their **private key**.
    - Ensures **confidentiality**.
2. **Digital Signatures:**
    - Sender signs message using their **private key**.
    - Receiver verifies using sender's **public key**.
    - Provides **integrity, authenticity, and non-repudiation**.

3. **Key Exchange:**
    - Used to securely exchange session keys for symmetric encryption.
    - Example: Diffie–Hellman, RSA in SSL/TLS.
4. **Authentication:**
    - Users, devices, and servers can prove their identity using public-private keys.
    - Example: SSH login, certificate-based authentication.
5. **Public Key Infrastructure (PKI):**
    - Digital certificates bind public keys to identities.
    - Basis for **SSL/TLS in web security, e-commerce, and online banking**.
6. **Blockchain and Cryptocurrencies:**
    - Wallets use private keys for ownership and transactions.
    - Public keys act as addresses.

b) Write a short note on public key cryptanalysis. (6 Marks)

Public key cryptanalysis involves attempts to:

- Derive the private key
- Recover plaintext without authorization

Types of attacks:

1. Brute-force attack
2. Mathematical attack
3. Chosen-plaintext attack

Security depends on:
Computational infeasibility of solving underlying mathematical problems.

c) Illustrate the security features of ECC with examples. (4 Marks)

Elliptic Curve Cryptography (ECC) provides:

- Smaller key size
- Faster computation
- Lower memory and power usage

Example:

- 256-bit ECC ≈ 3072-bit RSA

Application:
Mobile devices and IoT systems.

OR
Q4. a) Explain the Diffie-Hellman key exchange algorithm with an example and diagram. (10 Marks)

- The Diffie–Hellman algorithm allows **two parties (A and B)** to securely agree on a **shared secret key** over an insecure channel.
- This shared key can then be used with **symmetric encryption** to secure communication

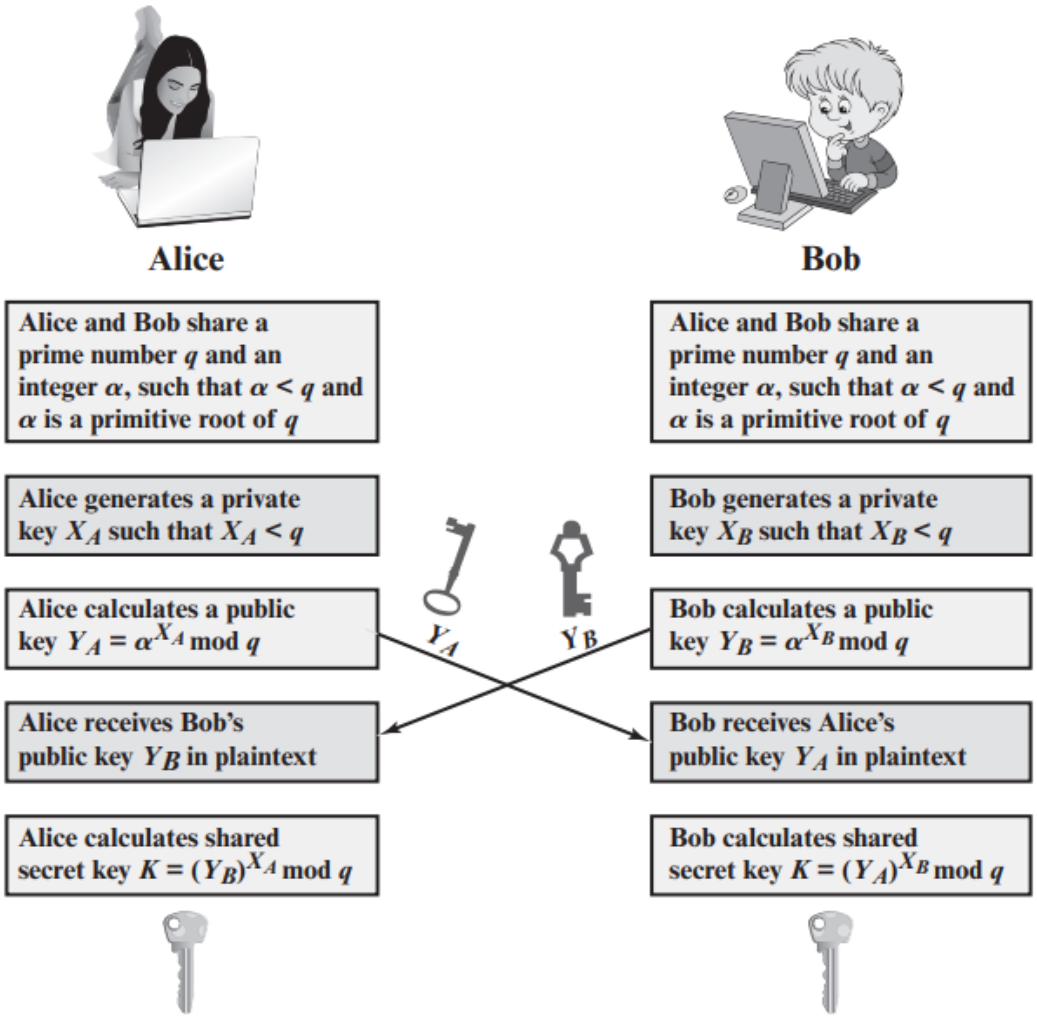


Figure 10.1 The Diffie–Hellman Key Exchange

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q && \text{by the rules of modular arithmetic} \\
&= \alpha^{X_B X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$

$$
\begin{aligned}
K &= (Y_B)^{X_A} \bmod q \\
&= (\alpha^{X_B} \bmod q)^{X_A} \bmod q \\
&= (\alpha^{X_B})^{X_A} \bmod q && \text{by the rules of modular arithmetic} \\
&= \alpha^{X_B X_A} \bmod q \\
&= (\alpha^{X_A})^{X_B} \bmod q \\
&= (\alpha^{X_A} \bmod q)^{X_B} \bmod q \\
&= (Y_A)^{X_B} \bmod q
\end{aligned}
$$

1) Public parameters

- Prime $q = 29$
- Base (primitive root) $\alpha = 2$

  These are public.

2) Each user's public value

Compute $Y_A = \alpha^{X_A} \bmod q$ and $Y_B = \alpha^{X_B} \bmod q$.

- $Y_A = 2^5 \bmod 29 = 32 \bmod 29 = 3$.
- $Y_B = 2^{11} \bmod 29$.

  Compute quickly: $2^5 \equiv 3$ so $2^{10} \equiv 3^2 = 9$. Then $2^{11} \equiv 9 \cdot 2 = 18$.

  So $Y_B = 18$.

Public keys: $Y_A = 3,\ Y_B = 18$.

3) Shared secret (computed by both sides)

- A computes $K = Y_B^{X_A} \bmod 29 = 18^5 \bmod 29$.
  Work: $18^2 = 324 \equiv 5$.
  $18^4 \equiv 5^2 = 25$.
  $18^5 \equiv 18^4 \cdot 18 \equiv 25 \cdot 18 = 450 \equiv 450 - 15 \cdot 29 = 450 - 435 = 15$.
  So A gets $K = 15$.
- B computes $K = Y_A^{X_B} \bmod 29 = 3^{11} \bmod 29$.
  Work: $3^2 = 9,\ 3^4 = 9^2 = 81 \equiv 23,\ 3^8 \equiv 23^2 = 529 \equiv 7$.
  $3^{11} = 3^8 \cdot 3^2 \cdot 3 \equiv 7 \cdot 9 \cdot 3 = 7 \cdot 27 = 189 \equiv 189 - 6 \cdot 29 = 189 - 174 = 15$.
  So B also gets $K = 15$.

Shared secret key = 15.

b) How does the Man-in-the-Middle attack affect Diffie-Hellman? Explain. (6 Marks)

In a MITM attack:

Attacker intercepts exchanged values

Establishes two separate keys with sender and receiver

Result:

Confidentiality is compromised.

Prevention:

Authentication using digital certificates.

c) What is the role of a Pseudo Random Number Generator (PRNG) in cryptography? (4 Marks)

PRNGs generate pseudo-random values used for:

- Key generation
- Initialization vectors
- Nonces

Security depends on unpredictability.

Module – 3

Q5. a) Describe key management fundamentals with neat diagram. (10 Marks)

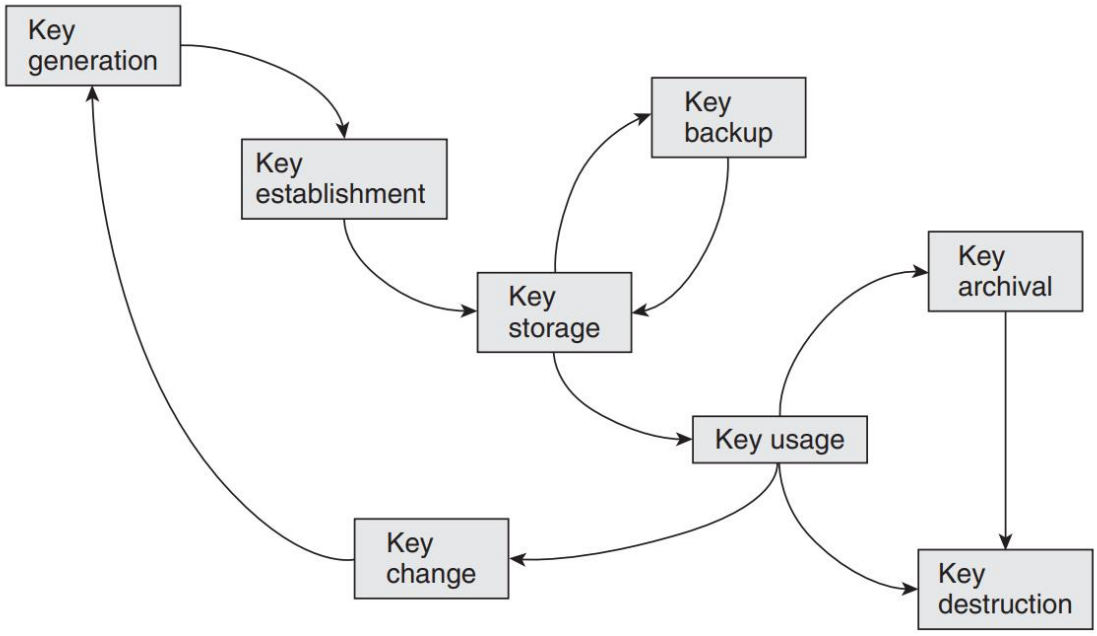


Figure 10.1. The key lifecycle

Key management involves secure handling of cryptographic keys throughout their lifecycle.

1.Key Generation

- This is where a cryptographic key is **created**.
- It must be generated in a **secure, random, and unpredictable way**.
- If key generation is weak, the whole cryptosystem is compromised.

2.Key Establishment

- The process of making sure that the key reaches the **endpoints** (the parties who need to use it).
- This is often the **most difficult phase** in practice, because keys must be transmitted securely.
- Two methods:
    - **Key transport** – one side generates the key and securely delivers it.
    - **Key agreement** – both sides contribute and compute the same key (e.g., Diffie–Hellman).

3. Key Storage

- Keys need to be **safely kept** during their usable lifetime.
- Includes:
    - **Secure storage** – e.g., Hardware Security Modules (HSMs), encrypted databases.
    - **Key backup** – to recover a key if it is lost.
    - **Key archival** – storing old keys for compliance or historical verification.

4. Key Usage

- How the key is actually **applied** in cryptographic operations (encryption, signing, authentication, etc.).
- This phase includes:
    - **Key change/rotation** – replacing keys regularly to reduce risk.
    - **Key destruction** – when a key's lifetime ends, it must be securely erased so it cannot be recovered.

5. Key Destruction

- The final phase of a key's life.

Keys must be **securely deleted/erased** so that attackers cannot recover them (even with forensic tools).

b) Explain the governing aspects of key management. (10 Marks)

Why governance matters?

- Key management is the **bridge** between cryptography and real-world systems/users.
- For individuals: it may just mean choosing techniques for each stage of the key lifecycle.
- For organizations: it's far more complex due to many interrelated processes
- Therefore, organizations need **rules and processes** to govern key management effectively.

Key Management Policies, Practices, and Procedures

1. Policies

- Define **overall requirements & strategy**.
- Example: "All cryptographic keys must be stored only in hardware."

2. Practices

- Define **tactics** to achieve policy goals.
- Example: "All devices using cryptography must include a built-in Hardware Security Module (HSM)."

3. Procedures

- Define **step-by-step tasks** for implementing practices.
- Example: "Specify the key establishment protocol used between two devices.

Principles of Effective Key Management Governance

- **By Design** → lifecycle planned from the start, not improvised.
- **Coherent** → phases of the lifecycle are linked together as part of a bigger unified process.

Integrated → aligned with the wider organizational requirements and priorities.
OR
Q6. a) Discuss key generation, establishment and storage methods. (10 Marks)

Cryptographic keys are fundamental to the security of encryption systems. The overall strength of a cryptosystem depends not only on the algorithm used but also on how securely the keys are **generated, established, and stored**. Improper handling of keys can lead to complete system compromise.

1. Key Generation

Definition

Key generation is the process of creating cryptographic keys with sufficient **randomness, unpredictability, and length** to resist attacks.

Methods of Key Generation

a) Random Number Generators (RNGs)

- RNGs produce sequences of numbers that appear random.
- Used in generating secret keys, nonces, and initialization vectors.

Types:

- **True Random Number Generators (TRNGs):**
  Use physical phenomena such as thermal noise or electronic jitter.
- **Pseudo Random Number Generators (PRNGs):**
  Use mathematical algorithms and an initial seed.

Requirement:
Generated keys must be **unpredictable** and **uniformly distributed**.

---

b) Hardware-Based Key Generators

- Use hardware sources of entropy.
- Provide high-quality randomness.
- Resistant to software-based attacks.

Advantages

- High security
- Difficult to predict or reproduce

---

2. Key Establishment

Definition

Key establishment refers to the process by which cryptographic keys are securely shared between communicating entities.

Methods of Key Establishment

a) Key Distribution Center (KDC)

- A **trusted third party** that generates and distributes secret keys.

- Used in symmetric-key systems.

Process:

1. Users authenticate to KDC
2. KDC generates a session key
3. Securely distributes the key to both parties

Advantages

- Centralized control
- Efficient for closed networks

Limitation

- Single point of failure

---

b) Public-Key Protocols

- Use asymmetric cryptography to establish keys.
- No need for prior shared secret.

Example: Diffie–Hellman Key Exchange

Advantages

- Secure over insecure networks
- Scalable

Limitation

- Computationally expensive

Key Establishment Diagram (Neat – VTU Style)

```
 User A                  User B
   |                       |
   | Public-key exchange   |
   |---------------------->|
```

| |

| Shared secret established |

| |

---

3. Key Storage

Definition

Key storage involves securely storing cryptographic keys to prevent unauthorized access, disclosure, or modification.

Methods of Key Storage

a) Hardware Security Modules (HSMs)

- Dedicated hardware devices for key storage and management.
- Perform cryptographic operations internally.

Features

- Tamper-resistant
- High physical security

Applications

- Banking systems
- Certificate Authorities

b) Encrypted Software Storage

- Keys stored in encrypted form in files or databases.
- Protected using passwords or master keys.

Advantages

- Low cost
- Easy deployment

Limitation

- Vulnerable if system is compromised

b) Explain the importance of key usage and storage in cryptosystems. (10 Marks)

In any cryptosystem, **keys** are the most critical components because they control access to the encrypted data. Proper usage and secure storage of keys ensure the confidentiality, integrity, and authenticity of information. Poor key management can completely compromise even the strongest encryption algorithms.

1. Importance of Key Usage

- **Access Control:** Only authorized parties with the correct key can decrypt the ciphertext and access the original information.
- **Limiting Exposure:** Keys should have limited lifetime and scope of use (e.g., session keys vs. master keys) to reduce the risk of compromise.
- **Separation of Duties:** Different keys can be used for encryption, decryption, signing, and verification to prevent misuse.
- **Algorithm Efficiency:** Correct key usage ensures that cryptographic algorithms function optimally and securely.

Example:
Using the same key for multiple purposes (e.g., encryption and digital signatures) can lead to attacks like key-recovery exploits.

2. Importance of Key Storage

- **Prevention of Unauthorized Access:** Keys must be stored securely to prevent theft, duplication, or tampering.
- **Hardware and Software Storage:**
    - **Hardware Security Modules (HSMs):** Provide tamper-resistant storage for cryptographic keys.
    - **Encrypted Key Storage:** Keys can be encrypted using a master key and stored in databases or files.
- **Backup and Recovery:** Keys must have secure backup mechanisms to avoid permanent data loss in case of hardware or software failure.

- **Lifecycle Management:** Proper storage ensures keys are rotated, expired, or revoked according to security policies.

3. Consequences of Poor Key Management

- Unauthorized data access
- Data corruption or loss
- Impersonation attacks (e.g., forging digital signatures)
- Complete compromise of cryptosystem security

Module – 4

Q7. a) Explain web security considerations. (8 Marks)

Web security ensures the protection of websites, web applications, and web services from threats that can compromise confidentiality, integrity, and availability of data. With the increasing use of web applications for e-commerce, banking, and data sharing, robust web security is essential.

1. Importance of Web Security

- Prevent unauthorized access to sensitive information.
- Ensure integrity of data transmitted over the internet.
- Maintain availability of web services for legitimate users.
- Protect against financial and reputational losses caused by cyber attacks.

2. Key Web Security Considerations

A. Authentication and Authorization

- **Authentication:** Verify the identity of users before granting access.
    - Example: Passwords, OTP, biometric login.
- **Authorization:** Ensure users can access only resources they are permitted to.
    - Example: Role-based access control (RBAC).

B. Data Encryption

- Encrypt data during transmission using **HTTPS/TLS** to prevent eavesdropping.
- Store sensitive data like passwords and credit card numbers in encrypted form.

C. Input Validation

- Prevent malicious inputs that can lead to attacks like **SQL injection** or **cross-site scripting (XSS)**.
- Ensure that all user inputs are sanitized before processing.

D. Session Management

- Securely manage user sessions to prevent **session hijacking** or **session fixation**.
- Techniques: Use secure cookies, timeout inactive sessions, regenerate session IDs after login.

E. Secure Coding Practices

- Avoid common programming mistakes that introduce vulnerabilities.
- Regularly update frameworks, libraries, and dependencies.

F. Regular Security Updates and Patching

- Apply security patches to web servers, applications, and operating systems to fix known vulnerabilities.

G. Monitoring and Logging

- Track user activity and system events to detect suspicious behavior.
- Helps in early detection of attacks and forensic analysis.

H. Backup and Recovery

- Regularly backup web application data to recover from attacks such as **ransomware** or **data corruption**.

3. Common Web Threats

- **Cross-Site Scripting (XSS):** Inject malicious scripts into web pages.
- **SQL Injection:** Manipulate database queries via unvalidated input.
- **Denial of Service (DoS):** Overload server to make it unavailable.
- **Phishing and Malware:** Trick users into revealing sensitive information.

b) Illustrate the working of Transport Layer Security (TLS). (6 Marks)

Working of Transport Layer Security (TLS)

Transport Layer Security (TLS) is a cryptographic protocol designed to provide secure communication over the internet. It ensures **confidentiality, integrity, and authentication** between a client (e.g., web browser) and a server (e.g., web server). TLS is widely used in HTTPS, email, and VPNs.

1. Objectives of TLS

1. **Confidentiality:** Encrypt data to prevent eavesdropping.
2. **Integrity:** Ensure data is not modified during transmission.
3. **Authentication:** Verify the identity of communicating parties.

2. TLS Protocol Structure

TLS works in **two main layers**:

1. **TLS Record Layer:**
    - Handles **fragmentation, compression, encryption, and message authentication**.
    - Provides **confidentiality and integrity** of the data.
2. **TLS Handshake Protocol:**
    - Establishes **session keys** and negotiates cryptographic algorithms.
    - Ensures **mutual authentication** between client and server.

3. TLS Handshake Process

The TLS handshake consists of the following steps:

Step 1: Client Hello

- Client sends:
    - TLS version
    - Supported cipher suites
    - Random number (nonce)

Step 2: Server Hello

- Server responds with:
    - Chosen TLS version and cipher suite
    - Server's digital certificate (for authentication)
    - Random number

Step 3: Certificate Verification

- Client verifies the server certificate using a **trusted Certificate Authority (CA)**.
- Ensures server authenticity.

Step 4: Key Exchange

- Client and server exchange keys to generate a **shared session key**.
- Methods:
    - RSA key exchange
    - Diffie-Hellman / ECDHE for Perfect Forward Secrecy

Step 5: Session Key Generation

- Both parties generate the **symmetric session key** using exchanged information.
- This key is used for **encrypting the application data**.

Step 6: Finished Messages

- Both client and server send **Finished messages** encrypted with the session key.
- Confirms that handshake is successful and secure communication can begin.

4. Data Transmission

- Once the handshake is complete, all application data (e.g., HTTP requests and responses) is encrypted using the session key.
- TLS ensures **confidentiality, integrity, and protection against tampering or replay attacks**.

c) What is a Security Parameters Index (SPI) in IP security? (6 Marks)

Security Parameters Index (SPI) in IP Security (6 Marks)

Definition:
The **Security Parameters Index (SPI)** is a **32-bit value** used in **IP Security (IPsec)** to uniquely identify a particular **security association (SA)** for an IP packet. It tells the receiving system **which security protocols, keys, and algorithms** to use for processing the packet.

1. Purpose of SPI

- SPI is part of the **IPsec header** (AH or ESP) and acts as an **index** to a Security Association (SA) in the Security Association Database (SAD).
- Ensures that the **correct cryptographic parameters** are applied to incoming packets.
- Provides flexibility to handle **multiple SAs** simultaneously between hosts.

2. Components of a Security Association

Each SA associated with an SPI contains:

1. **SPI Value:** Unique identifier for the SA.
2. **IP Destination Address:** Specifies the target host for the SA.
3. **Security Protocol:** AH (Authentication Header) or ESP (Encapsulating Security Payload).
4. **Encryption & Authentication Keys:** Keys to encrypt/decrypt and authenticate the packets.
5. **Other Parameters:** Lifetime, algorithms, sequence numbers, etc.

3. How SPI Works

- When a packet arrives, the **IPsec header** contains the SPI.
- The receiver looks up the SPI in the **Security Association Database (SAD)**.
- The SA specifies the **cryptographic algorithm and key** to process the packet.

- The packet is then decrypted and/or authenticated accordingly.

Example:

- SPI = 0x12345678
- Protocol = ESP
- Destination = 192.168.1.2
- Algorithm = AES-256
- Key = K1A2B3C4...

The SPI ensures the packet is processed using **the correct security context**.

OR

Q8. a) Describe the IP security overview and the need for IP security policies. (8 Marks)

1. Overview of IP Security (IPsec)

IP Security (IPsec) is a suite of protocols designed to provide **secure communication over IP networks**. It protects IP packets at the **network layer**, ensuring **confidentiality, integrity, and authentication** of data between communicating devices.

Key Features of IPsec:

- **Data Confidentiality:** Encrypts IP packet payloads to prevent eavesdropping.
- **Data Integrity:** Ensures packets are not modified in transit using hash-based authentication.
- **Authentication:** Verifies the identity of the sender using digital signatures or shared keys.
- **Anti-Replay Protection:** Prevents attackers from resending captured packets.

Main IPsec Protocols:

1. **Authentication Header (AH):** Provides integrity, authentication, and anti-replay protection; does not encrypt the payload.
2. **Encapsulating Security Payload (ESP):** Provides encryption for confidentiality, plus optional integrity and authentication.

Modes of Operation:

- **Transport Mode:** Protects only the payload of the IP packet.

- **Tunnel Mode:** Encapsulates the entire IP packet for secure communication between gateways.

2. Need for IP Security Policies

IPsec policies define **rules and parameters** that determine how traffic is secured and which security measures to apply. They are critical for **consistent and effective network security**.

Reasons for IP Security Policies:

Need	Explanation
Traffic Classification	Determine which types of traffic require protection (e.g., email, database access, VPN).
Algorithm Selection	Specify which encryption and authentication algorithms to use for different traffic.
Key Management	Define how keys are generated, exchanged, and rotated for secure communication.
Access Control	Ensure only authorized devices or users can access sensitive resources.
Consistency and Compliance	Maintain uniform security across the network and comply with regulations.
Performance Optimization	Apply security only where necessary to avoid unnecessary overhead.

Example Policy:

- All traffic between branch office and HQ must use **ESP with AES-256 encryption** in **tunnel mode**.
- Only authenticated hosts can access sensitive databases.

b) What is Encapsulating Security Payload (ESP)? Explain its purpose. (6 Marks)

1. Definition of ESP

Encapsulating Security Payload (ESP) is a core protocol of **IP Security (IPsec)** that provides **confidentiality, integrity, and authentication** for IP packets. Unlike the

Authentication Header (AH), ESP **encrypts the payload** of the IP packet, ensuring that the data cannot be read by unauthorized parties.

ESP can operate in **transport mode** or **tunnel mode**:

- **Transport Mode:** Encrypts only the payload of the IP packet; header remains intact.
- **Tunnel Mode:** Encapsulates the entire IP packet, creating a new IP header; commonly used in VPNs.

2. Structure of an ESP Packet

An ESP packet generally contains the following fields:

Field	Purpose
SPI (Security Parameters Index)	Identifies the security association (SA) for the packet.
Sequence Number	Protects against replay attacks by numbering packets.
Payload Data	The encrypted original data (e.g., TCP/UDP payload).
Padding	Ensures proper alignment for encryption algorithms.
Authentication Data	Optional integrity check (MAC) to verify authenticity.

3. Purpose of ESP

ESP provides the following security services:

1. **Confidentiality:** Encrypts the payload to protect against eavesdropping.
    - Example: Using AES or 3DES algorithms.
2. **Authentication and Integrity (Optional):** Ensures that data is from a legitimate sender and has not been modified.
    - Example: Using HMAC-SHA256.
3. **Anti-Replay Protection:** Uses sequence numbers to prevent attackers from re-sending captured packets.
4. **Support for Secure Tunneling:** Encapsulates entire IP packets in tunnel mode for secure site-to-site or remote access VPNs.

4. Advantages of ESP

- Provides **strong encryption** for confidentiality.
- Supports both **host-to-host** and **gateway-to-gateway** communication.
- Can provide **authentication and integrity** along with encryption.
- Works in conjunction with **Security Associations (SA)** for flexible security management.

c) Illustrate the role and phases of Internet Key Exchange (IKE) protocol. (6 Marks)

1. Role of IKE

The Internet Key Exchange (IKE) protocol is a key management protocol used in IP Security (IPsec). Its primary role is to establish, negotiate, and manage Security Associations (SAs) and cryptographic keys between two IPsec peers.

Key Roles:

Automatically negotiates security parameters (encryption, authentication algorithms).

Performs mutual authentication between peers.

Generates shared secret keys for IPsec communication.

Refreshes keys periodically to maintain security.

Benefit: IKE eliminates the need for manual key exchange, reducing configuration errors and enhancing security.

2. Phases of IKE

IKE works in two main phases:

Phase Purpose Details

Phase 1 – IKE SA Establishment Establishes a secure, authenticated channel (IKE SA) between peers.- Peers authenticate each other (using pre-shared keys, certificates, or public keys).

- Negotiates encryption and authentication algorithms.

- Creates a secure tunnel for Phase 2 communication.

- Modes: Main Mode (more secure) or Aggressive Mode (faster).

Phase 2 – IPSec SA Negotiation Establishes IPsec SAs for actual data transfer. - Uses the secure channel from Phase 1.

- Negotiates IPsec protocols (ESP or AH), keys, and lifetimes.

- Can support perfect forward secrecy by generating new keys without Phase 1 re-authentication.

3. Flow of IKE Protocol

Phase 1:

Exchange proposals for algorithms and modes.

Authenticate peers and establish IKE SA.

Phase 2:

Negotiate IPsec SAs.

Agree on encryption and integrity keys.

Begin secure data transmission using ESP or AH.

Module – 5

Q9. a) Discuss various data hiding techniques in text. (10 Marks)
Analyze the LSB (Least Significant Bit) encoding technique for image steganography and demonstrate its application with a suitable example.

Definition:

LSB (Least Significant Bit) encoding is a spatial-domain steganography technique in which secret data is embedded into the least significant bits of the pixel values of a digital image. Since the LSB contributes the least to image intensity, modifying it does not produce visible distortion.

Principle of LSB Encoding

- An 8-bit grayscale pixel has bits b7 b6 b5 b4 b3 b2 b1 b0.
- The LSB (b0) has the smallest effect on pixel brightness.
- Replacing the LSB with message bits keeps the image visually unchanged.
- In RGB images, each pixel has 3 bytes (R,G,B), so 3 bits of data can be stored per pixel.

Steps of LSB Encoding

1. Convert the secret message to a binary bitstream.

2. Take each bit of the message and place it in the LSB of each pixel (or color component).

3. Continue embedding until all bits are stored.

4. The resulting image is called the stegoimage.

Example

Cover Image Pixels (grayscale):

(100, 150, 212, 99)

Binary values:

- 100 → 01100100
- 150 → 10010110
- 212 → 11010100
- 99 → 01100011

Secret Message: "A" → ASCII 65 → binary 01000001

Embedding the first 4 bits into LSBs:

Pixel	Original Binary	Replace LSB	Modified Binary	New Pixel
100	01100100	0	01100100	100
150	10010110	1	10010111	151
212	11010100	0	11010100	212
99	01100011	0	01100010	98

This small change is imperceptible, proving LSB's effectiveness. b) Explain the LSB encoding technique with a suitable example. (10 Marks)

OR

Q10. a) Explain watermarking and its intuitive and digital methods. (10 Marks)

Watermarking is a technique used to embed information into digital media (images, audio, video, or documents) to assert ownership, ensure authenticity, or protect against unauthorized copying. Unlike encryption, watermarking does not prevent access to the content but makes tampering or illegal use detectable.

1. Definition and Purpose of Watermarking

- **Definition:** A watermark is a recognizable pattern or signal embedded into the media that can be extracted or detected to prove ownership or integrity.
- **Purpose:**
    - Copyright protection
    - Authentication of content
    - Data integrity verification
    - Tracking unauthorized distribution

2. Types of Watermarking Methods

Watermarking can be broadly categorized into **intuitive (or perceptual) methods** and **digital methods**:

A. Intuitive Methods

Intuitive watermarking refers to **manual or perceptual ways** of embedding marks that are visible or noticeable to humans.

- **Visible Watermarks:**
    - Example: Logo or text overlaid on images or videos.
    - Easy for humans to recognize.
    - Mainly used in media previews or branding.
- **Semi-visible Watermarks:**

    - Slightly transparent marks that do not obstruct viewing but indicate ownership.

- **Pros:** Simple to implement, easy to detect visually.
- **Cons:** Can be removed by cropping, filtering, or editing.

Example: Newspaper logos printed on digital photos to prevent misuse.

B. Digital (Invisible) Watermarking Methods

Digital watermarking embeds information into the **digital representation of media** in a way that is **imperceptible to humans**, but detectable by software.

1. **Spatial Domain Techniques:**
    - Modify pixel values directly.
    - Example: Least Significant Bit (LSB) insertion in images.
    - Pros: Simple and fast.
    - Cons: Vulnerable to compression and noise.
2. **Frequency Domain Techniques:**
    - Embed watermark in transformed coefficients (e.g., DCT, DWT, FFT).
    - Pros: Robust against attacks like compression, scaling, and filtering.
    - Example: Embedding watermark in DCT coefficients of JPEG images.
3. **Spread Spectrum Watermarking:**
    - Watermark signal spread across multiple frequencies.
    - Difficult to detect or remove without the key.
4. **Robust vs. Fragile Watermarks:**
    - **Robust Watermark:** Survives attacks like compression, resizing, or noise. Used for copyright.
    - **Fragile Watermark:** Designed to break if media is altered. Used for tamper detection.

b) Illustrate the process of data hiding in text using innocuous text techniques. (10 Marks)

Data hiding in text is a form of **steganography**, where secret information is embedded within a text document without drawing attention. The main goal is to transmit information covertly while the text appears normal.

1. Innocuous Text Techniques

Innocuous text techniques hide information in a text that appears **ordinary and harmless**. These techniques rely on **subtle manipulations of text** rather than visible encryption.

Common methods include:

A. Synonym Substitution

- Replaces words in the cover text with their synonyms to encode bits of secret data.
- **Example:**
    - Secret bit 0 → word "quick"
    - Secret bit 1 → word "fast"
    - Original sentence: "The quick brown fox jumps over the lazy dog."
    - Encoded sentence (bit 1): "The fast brown fox jumps over the lazy dog."
- **Advantage:** Text looks natural.
- **Limitation:** Contextual errors may arise if synonyms are inappropriate.

B. Abbreviation/Substitution

- Uses abbreviations or alternate forms to encode information.
- **Example:**
    - Secret bit 0 → "Doctor"
    - Secret bit 1 → "Dr."
- Original: "Doctor Smith arrived at noon."

- Encoded: “Dr. Smith arrived at noon.”

C. Punctuation Manipulation

- Secret data is encoded by adding, removing, or modifying punctuation marks.
- **Example:**
    - Secret bit 0 → no extra space after a comma
    - Secret bit 1 → extra space after a comma
- Works well for covertly embedding binary data.

D. Line or Word Spacing

- Encodes data using subtle variations in spacing:
    - Extra space → 1
    - Single space → 0

E. Capitalization Patterns

- Using capitalization of certain letters to encode information.
- Example: capitalizing first letter of every word at positions corresponding to 1s.

2. Process of Data Hiding Using Innocuous Text Techniques

Step 1: Select Cover Text

- Choose a normal-looking text (e.g., email, article, story) that can carry hidden data.

Step 2: Encode Secret Data

- Convert secret information into binary format.

Step 3: Apply Innocuous Text Technique

- Use synonym substitution, abbreviation, punctuation, or spacing to embed binary data into text.

Step 4: Generate Stego-Text

- Resulting text appears normal and readable but contains the hidden information.

Step 5: Extraction

- Receiver uses the agreed-upon technique to decode the secret bits from the stego-text.

3. Example

Secret Message: 101
Original Text: "The quick brown fox jumps."

Bit	Technique	Resulting Word
1	Synonym	"fast"
0	Synonym	"brown"
1	Synonym	"leaps"

Stego-Text: "The fast brown leaps fox jumps."

The text looks natural, but the hidden binary 101 can be extracted using the agreed mapping.

4. Advantages

- Text appears normal → low suspicion
- Simple and low-cost to implement
- Works for email, articles, documents

5. Limitations

- Limited data capacity
- Sensitive to text editing or reformatting