Cryptography & Computer Security – Detailed Notes

1. Attacks on Computers and Computer Security: Introduction

Computer security refers to protecting computer systems, networks, and data from unauthorized access, misuse, damage, or disruption.
It ensures **confidentiality, integrity, and availability** of information.

---

2. The Need for Security

Security is needed to:

- Protect sensitive data from unauthorized access.
- Prevent loss of data due to attacks.
- Ensure business continuity.
- Maintain user trust and system reliability.
- Prevent financial loss, identity theft, and cybercrimes.

---

3. Security Approaches

There are three major approaches:

a) Layered Security Approach

Security is implemented in multiple layers so that if one layer fails, another protects.

b) Security through Obscurity

Security depends on hiding internal system details.
(Not recommended alone.)

c) Principle of Least Privilege

Provide only the minimum required access to users or processes.

---

4. Principles of Security

The core principles are known as **CIA Triad**:

a) Confidentiality

Ensuring that information is accessible only to authorized users.

b) Integrity

Ensuring data is accurate and unaltered.

c) Availability

Ensuring systems and data are available when required.

Additional principles:

- **Authentication**
- **Authorization**
- **Non-repudiation**
- **Accountability**

---

5. Types of Security Attacks

a) Passive Attacks

- **Eavesdropping**
- **Traffic analysis** No modification is done; attacker only observes.

b) Active Attacks

- **Masquerade**
- **Replay**
- **Modification of data**
- **Denial of Service (DoS)**
  Attacker modifies data or disrupts operations.

---

6. Security Services

Security services are mechanisms that provide protection:

- **Confidentiality** – protects information from unauthorized disclosure.
- **Integrity** – ensures message remains unchanged.
- **Authentication** – verifies identity of sender/receiver.
- **Non-repudiation** – prevents denial of actions.
- **Access Control** – regulates who can access what.
- **Availability** – keeps systems operational.

---

7. Security Mechanisms

These are tools/techniques that implement security services:

- **Encryption**
- **Digital signatures**
- **Access control mechanisms**
- **Firewalls**
- **Intrusion detection systems**
- **Security audit trails**
- **Hash functions**

---

8. A Model for Network Security

A general security model includes:

1. **Sender** – creates the message.
2. **Encryption Algorithm** – converts plaintext into ciphertext using key.
3. **Receiver** – decrypts the ciphertext.
4. **Key Management** – secure exchange of keys.
5. **Security services** – ensure confidentiality, integrity, authentication.

Cryptography: Concepts and Techniques

9. Introduction

Cryptography is the science of securing information by transforming it into unreadable format for unauthorized users.
It ensures secure communication over insecure channels.

---

10. Plain Text and Cipher Text

- **Plaintext:** Original readable message.
- **Ciphertext:** Encrypted unreadable message.

Encryption Formula (General):

```
Ciphertext = E(Key, Plaintext)
```

Decryption Formula:

```
Plaintext = D(Key, Ciphertext)
```

---

11. Substitution Techniques

Encryption method where characters are replaced with other characters.

Examples:

- **Caesar Cipher**
  Shift each letter by fixed number.
  Formula:
- `C = (P + K) mod 26`
- `P = (C - K) mod 26`
- **Monoalphabetic Cipher** One-to-one substitution using a key table.
- **Playfair Cipher** Uses 5×5 matrix and digraph substitution.

---

12. Transposition Techniques

Rearrange the characters of plaintext without changing them.

Example: Rail Fence Cipher

Plaintext is written in zigzag pattern and then read row-wise.

Columnar Transposition

Plaintext written in rows; columns are re-arranged based on key.

---

13. Encryption and Decryption

Encryption

Process of converting plaintext into ciphertext using an algorithm and a key.

Decryption

Reverse process of converting ciphertext back to plaintext.

---

14. Symmetric Key Cryptography

- Same key is used for encryption and decryption.
- Fast and efficient.

Examples:

- DES
- AES
- RC4
- Blowfish

Formula:

```
C = E(K, P)
P = D(K, C)
```


---

15. Asymmetric Key Cryptography

- Uses **two different keys**: Public key and Private key.

Properties:

- Public key is for encryption.
- Private key is for decryption.
- More secure but slower.

Examples:

- RSA
- Diffie-Hellman
- ECC

RSA Formula:

Public Key = (e, n)
Private Key = (d, n)

```
Encryption:   C = P^e mod n
Decryption:   P = C^d mod n
```

---

16. Steganography

Technique of hiding data inside other data like images, audio, video.

Examples:

- Hiding text in image pixels.
- Hiding messages in audio signals.

---

17. Key Range and Key Size

- **Key size determines strength of encryption.**
- Larger key → more secure → harder to brute force.

Common key sizes:

- DES – 56 bits
- AES – 128/192/256 bits
- RSA – 1024/2048/4096 bits

---

18. Possible Types of Attacks in Cryptography

a) Brute Force Attack

Attacker tries all possible keys.

b) Cryptanalysis

Mathematical attacks to break cipher (e.g., linear, differential).

c) Man-in-the-Middle Attack

Attacker intercepts communication between two parties.

d) Replay Attack

Intercepted data is resent by attacker.

e) Side-Channel Attacks

Based on timing, power consumption etc.

Symmetric Key Ciphers – Detailed

---

1. Symmetric Key Ciphers

Symmetric key cryptography uses the **same secret key** for both encryption and decryption.
It is fast, efficient, and widely used for large data encryption.

General Formula:

```
Encryption:   C = E(K, P)
Decryption:   P = D(K, C)
```

Where:

- **P** = Plaintext
- **C** = Ciphertext
- **K** = Shared secret key

---

2. Block Cipher Principles

Block ciphers encrypt data in **fixed-size blocks** (64-bit, 128-bit, etc.).
The plaintext is divided into blocks and each block is encrypted separately using the same key.

Design Principles

- **Confusion:** Relationship between key and ciphertext should be complex.
- **Diffusion:** Changing one bit of plaintext should change many bits of ciphertext.
- **Product Ciphers:** Combination of substitution + permutation rounds.
- **Feistel Structure:** Common design where each round splits data into two halves (e.g., DES).

Block Size Examples

- DES: 64-bit block
- AES: 128-bit block

---

3. Block Cipher Algorithms

A. DES (Data Encryption Standard)

DES is a **64-bit block cipher** with a **56-bit key**.
Uses **Feistel structure** with **16 rounds**.

DES Process

1. Initial Permutation (IP)
2. 16 Feistel rounds:
    - Expansion (32 → 48 bits)
    - XOR with round key
    - S-Box substitution
    - Permutation
3. Swap left & right
4. Final Permutation (FP)

DES Key Size

```
Original key = 64 bits
Effective key = 56 bits (8 bits used for parity)
```

DES Weakness

- Small key size → vulnerable to brute-force attack.

B. AES (Advanced Encryption Standard)

AES is a **128-bit block cipher** with key sizes **128/192/256 bits**.
Based on **Substitution–Permutation Network (SPN)** structure (not Feistel).

AES Rounds

- AES-128 → 10 rounds
- AES-192 → 12 rounds
- AES-256 → 14 rounds

AES Steps

1. **SubBytes** – S-box substitution
2. **ShiftRows** – cyclic shift
3. **MixColumns** – mixing columns
4. **AddRoundKey** – XOR with key

AES is strong, fast, and widely used in modern security systems.

---

4. Block Cipher Modes of Operation

Modes define **how block ciphers encrypt large messages**.

---

1) ECB (Electronic Codebook Mode)

```
C_i = E(K, P_i)
```

- Each block encrypted independently.
- Simple but insecure: identical plaintext blocks → identical ciphertext blocks.
- Not recommended.

---

2) CBC (Cipher Block Chaining Mode)

```
C_i = E(K, P_i XOR C_{i-1})
```

(Uses Initialization Vector IV for first block.)

- Each block depends on previous block.
- Provides diffusion.
- Widely used.

---

3) CFB (Cipher Feedback Mode)

```
C_i = P_i XOR E(K, C_{i-1})
```

- Converts block cipher into stream cipher.
- No need to pad plaintext.

---

4) OFB (Output Feedback Mode)

```
Output_i = E(K, Output_{i-1})
C_i = P_i XOR Output_i
```

- Error does not propagate.
- Pre-computable keystream.

---

5) CTR (Counter Mode)

```
C_i = P_i XOR E(K, Counter_i)
```

- Very fast, parallel encryption.
- Most used in modern systems (IPSec, TLS).

---

5. Stream Ciphers

A stream cipher encrypts data **bit-by-bit** or **byte-by-byte** using a keystream.

Characteristics

- Fast for real-time communication.
- No padding required.
- Key stream must be unpredictable.

Examples

- RC4
- Salsa20
- ChaCha20

---

6. RC4 (Rivest Cipher 4)

RC4 is a widely used **stream cipher** (historically used in SSL, WEP, WPA).

Key Features

- Variable key length: 40 to 2048 bits
- Generates pseudorandom byte stream
- Encryption is XOR-based

RC4 Algorithm Steps

1. **KSA (Key Scheduling Algorithm)**
   Initializes 256-byte array S using the key.
2. **PRGA (Pseudo-Random Generation Algorithm)**
   Generates keystream bytes.

Encryption/Decryption:

```
Ciphertext = Plaintext XOR Keystream
```

```
Plaintext  = Ciphertext XOR Keystream
```

Weakness

- Vulnerable in WEP/WPA because of poor IV usage.
- Not recommended today.

---

7. Location and Placement of Encryption Function

Encryption can be placed at different layers of network architecture:

---

1. Application Layer Encryption

Examples:

- HTTPS
- Email encryption (PGP, S/MIME)

✔ End-to-end security
✔ User controlled
✗ Higher overhead

---

2. Transport Layer Encryption

Examples:

- SSL/TLS

✔ Protects data between client and server
✔ Widely used for secure websites

---

3. Network Layer Encryption

Examples:

- IPSec
  ✔ Secures all IP packets
  ✔ Transparent to applications

---

4. Link Layer Encryption

Examples:

- WPA/WPA2 for Wi-Fi
  ✔ Fast
  ✗ Only protects local links, not end-to-end

---

8. Key Distribution

Key distribution is one of the biggest challenges in symmetric cryptography.

Methods for Key Distribution

1. **Physical delivery**
   - Key delivered manually (secure but slow).
2. **Using a trusted third party (KDC – Key Distribution Center)**
   - Common in Kerberos.
3. **Using public key cryptography**
   - Symmetric key exchanged using RSA/Diffie-Hellman.
4. **Session keys**
   - Short-term keys for one communication session.

Basic Key Distribution Formula (Using KDC):

```
K_AB = E(K_A, Session_Key) || E(K_B, Session_Key)
```

Asymmetric Key Ciphers – Detailed

---

1. Introduction to Asymmetric Key Cryptography

Asymmetric key cryptography (also called **Public Key Cryptography**) uses **two keys**:

- **Public Key** → used for encryption
- **Private Key** → used for decryption

The two keys are mathematically related, but the **private key cannot be derived from the public key**.

---

2. Principles of Public Key Cryptosystems

A public-key cryptosystem must satisfy the following principles:

---

1. Two-Key Concept

Each user generates:

- **Public Key (shared)**
- **Private Key (kept secret)**

The public key encrypts the message, and only the private key can decrypt it.

---

2. One-Way Function

Public key cryptography is based on mathematical functions that are:

- Easy to compute in one direction
- Hard to invert without special information

Examples:

- Integer factorization (RSA)
- Discrete logarithm (Diffie–Hellman, ECC)

3. Security Depends on Key Secrecy, Not Algorithm

The algorithms are public; secrecy comes from:

- Large key sizes
- Mathematical hardness

---

4. Digital Signatures

Public key systems provide:

- Authentication
- Integrity
- Non-repudiation

Signature process:

```
Sender signs using private key.
Receiver verifies using public key.
```

---

5. Key Distribution

Public keys can be shared openly (certificates), unlike symmetric keys.

---

3. RSA Algorithm

RSA is the most widely used public key algorithm.

3.1 Key Generation

Steps to generate RSA keys:

1. Choose two large prime numbers:

```
p, q
```

2. Compute modulus:

```
n = p * q
```

3. Compute Euler's totient:

```
φ(n) = (p - 1)(q - 1)
```

4. Choose public exponent **e** such that:

```
1 < e < φ(n)  AND  gcd(e, φ(n)) = 1
```

5. Compute private key exponent:

```
d = e^(-1) mod φ(n)
```

Public Key = (e, n)

Private Key = (d, n)

3.2 Encryption

```
C = P^e mod n
```

3.3 Decryption

```
P = C^d mod n
```

Where:

- **P** = Plaintext (integer)
- **C** = Ciphertext

3.4 Security of RSA

RSA security is based on difficulty of:

- **Factoring large integers** (n = p × q)

Large key sizes (2048/4096 bits) provide strong security.

---

4. Diffie–Hellman Key Exchange

Diffie–Hellman is a method to securely exchange symmetric keys over an insecure channel.

4.1 Steps of Diffie–Hellman Algorithm

Public Values

Both users agree on:

```
Public base (g)
Large prime number (p)
```

Private Keys

Each user chooses a secret number:

```
Alice: a
Bob: b
```

Compute Public Values

```
Alice sends: A = g^a mod p
Bob sends: B = g^b mod p
```

Compute Shared Secret Key

```
Alice computes: K = B^a mod p
Bob computes:   K = A^b mod p
```

Both get the **same shared secret key**:

```
K = g^(ab) mod p
```

4.2 Security of Diffie–Hellman

Security depends on difficulty of:

- **Discrete Logarithm Problem (DLP)**

No one can compute `a` or `b` from `A` or `B`.

4.3 Weakness

Man-in-the-middle attack is possible without authentication.

---

5. Key Distribution in Asymmetric Systems

Asymmetric cryptography simplifies key distribution.

5.1 Public Key Directory

A trusted database stores user public keys.

5.2 Public Key Certificates

Issued by **Certificate Authority (CA)**.
Each certificate contains:

- User identity
- Public key
- Digital signature of CA

Used in:

- HTTPS
- Email security
- VPN

5.3 Key Exchange Using Asymmetric + Symmetric

A common modern method:

1. Public key cryptography → exchange symmetric keys
2. Symmetric key → encrypts bulk data (faster)

Used in:

- TLS/SSL
- IPsec
- SSH

5.4 Diffie–Hellman for Key Distribution

Used to securely share symmetric session keys between parties.

5.5 Hybrid Key Distribution Model

```
Symmetric key (session key) = encrypted using public key
Session key used for fast encryption/decryption
```

---

6. Summary Table ()

Topic	Key Points
Public Key Cryptosystem	Uses two keys (public/private), one-way functions
RSA	Based on integer factorization; encrypt: C = P^e mod n
Diffie–Hellman	Secure key exchange using g^ab mod p

Message Authentication Algorithms and Hash Functions – Detailed Notes

---

1. Authentication Requirements

Message authentication ensures that:

1. **Message is from a legitimate sender.**
2. **Message is not altered during transmission.**

3. **Sender cannot deny sending the message (non-repudiation).**
4. **Receiver and sender identities are verified.**

---

2. Authentication Functions

Authentication functions are mechanisms that provide:

- **Data Integrity**
- **Data Origin Authentication**
- **Message Authentication**

Common authentication functions:

- **Message Authentication Code (MAC)**
- **Hash Functions**
- **Digital Signatures**

---

3. Message Authentication Codes (MAC)

A **MAC** is an authentication tag generated using a **secret key** and a **message**.

MAC Formula

```
MAC = C(K, M)
```

Where:

- **K** = Secret key
- **M** = Message

Properties

- Ensures **integrity** and **authenticity**.
- Used in banking, secure communication channels, IPsec, TLS.

Examples

- CBC-MAC
- HMAC

- CMAC

---

4. Hash Functions

A hash function takes input of any size and produces a fixed-size output.

Hash Function Formula

```
H = h(M)
```

Properties of Cryptographic Hash Functions

1. **Pre-image resistance:** Hard to find message M from hash H.
2. **Second pre-image resistance:** Hard to find another message M2 such that:
3. `h(M1) = h(M2)`
4. **Collision resistance:** Hard to find two different messages that produce the same hash.

Hash Uses

- Password storage
- Digital signatures
- Integrity checking (MD5, SHA)
- Data indexing

---

5. Secure Hash Algorithms (SHA)

SHA is a family of hash functions developed by NIST.

Common SHA Versions

- **SHA-1** → 160-bit output (obsolete)
- **SHA-2** → 256/384/512 bit (secure)
- **SHA-256** is most widely used.

SHA Process (General)

1. Preprocessing (padding, parsing).
2. Generate message blocks.

3. Apply compression function.
4. Output hash of fixed size.

---

6. Whirlpool Hash Function

Whirlpool is a **512-bit cryptographic hash function**.

Features

- Based on AES-like structure
- Strong collision resistance
- Used for file integrity and password hashing

Output Size

```
Hash = 512 bits
```

---

7. HMAC (Hash-based Message Authentication Code)

HMAC uses a **hash function + secret key** to generate a MAC.

HMAC Formula

```
HMAC(K, M) = H( (K ⊕ opad) || H( (K ⊕ ipad) || M ) )
```

Properties

- Very secure
- Works with SHA-1, SHA-256, SHA-512
- Used in TLS, IPSec, API authentication

---

8. Digital Signatures

Digital signature is electronic equivalent of a handwritten signature.

Provides

- Authentication

- Integrity
- Non-repudiation

Process

Signing:

```
Signature = D_private( Hash(Message) )
```

Verification:

```
Hash2 = E_public(Signature)
Check: Hash(Message) == Hash2
```

Examples

- RSA Digital Signature
- DSA
- ECDSA

---

9. Authentication Applications

A. Kerberos

Kerberos is a network authentication protocol using **secret key cryptography** and **tickets**.

Main Components

- **KDC** – Key Distribution Center
- **AS** – Authentication Server
- **TGS** – Ticket Granting Server
- **Client & Server**

Kerberos Steps (Simplified)

1. Client requests Ticket-Granting Ticket (TGT).
2. KDC issues TGT encrypted with client key.
3. Client requests service ticket from TGS.
4. TGS sends service ticket.
5. Client uses ticket to authenticate with server.

Benefits

- Prevents replay attacks
- Mutual authentication
- No passwords sent over network

B. X.509

X.509 is a standard for **Public Key Certificates**.

Certificate Contains

- Subject name
- Public key
- Issuer (CA)
- Serial number
- Validity period
- Digital signature of CA

Uses

- HTTPS
- TLS/SSL
- Email security (S/MIME)

C. Authentication Service

Provides:

- Identity verification
- Credential validation
- Token/ticket generation

Examples

- Kerberos
- OAuth
- OpenID Connect

D. Public Key Infrastructure (PKI)

PKI is a framework for managing public keys and certificates.

Components of PKI

1. **Certificate Authority (CA)**
2. **Registration Authority (RA)**
3. **Certificate Repository**
4. **Certificate Revocation List (CRL)**

PKI Functions

- Create, store, distribute, revoke public key certificates.
- Ensures trust between users and systems.

E. Biometric Authentication

Uses biological characteristics to authenticate users.

Types

- Fingerprint
- Iris scan
- Face recognition
- Voice recognition
- Retina scan
- Hand geometry

Advantages

- Unique to individuals
- Cannot be forgotten or shared
- Difficult to forge

Disadvantages

- Expensive

- Privacy issues
- May be affected by injuries or physical changes

E-Mail Security and IP Security – Detailed

---

1. E-Mail Security

Email is one of the most widely used communication tools, but it is vulnerable to:

- Eavesdropping
- Message modification
- Spoofing
- Replay attacks

Two major email security standards are: **Pretty Good Privacy (PGP)** and **S/MIME**.

1.1 Pretty Good Privacy (PGP)

PGP is a widely used email encryption tool developed by Phil Zimmermann.

Main Features

1. **Confidentiality** – Uses symmetric encryption (IDEA, CAST, AES).
2. **Authentication** – Digital signatures using RSA or DSA.
3. **Compression** – Message is compressed before encryption.

4. **Radix-64 Encoding** – Converts binary to ASCII for email compatibility.
5. **Key Management** – Uses a **Web of Trust** model.

PGP Working Process

1. **Message created**
2. **Hash the message** using SHA-1/SHA-256
3. **Digital signature** generated using sender's private key
4. Message compressed
5. Generate random session key
6. Encrypt message with **symmetric key**
7. Encrypt session key with receiver's **public key**
8. Use Radix-64 encoding
9. Transmit

Advantages of PGP

- Strong security
- Free and widely available
- Uses hybrid encryption

Disadvantages

- Complex key management (Web of Trust)
- No centralized certificate authority

1.2 S/MIME (Secure / Multipurpose Internet Mail Extensions)

S/MIME is an industry standard for secure email supported by major email clients.

Main Features

1. **Authentication** using digital certificates (X.509)
2. **Message integrity**

3. **Non-repudiation**
4. **Confidentiality** (AES, 3DES)
5. **PKI-based key management**

S/MIME Architecture

- Uses **CMS (Cryptographic Message Syntax)**
- Uses **PKI** for certificate management
- Supports both **digital signing** and **encryption**

Differences: PGP vs S/MIME

Feature	PGP	S/MIME
Key Management	Web of Trust	PKI (Certificates)
Encryption	Hybrid	Hybrid
Certificates	Optional	Mandatory
Standardization	No	Yes

---

2. IP Security (IPsec)

IPsec is a framework for securing IP communication at the **network layer**.

2.1 IP Security Overview

IPsec provides:

1. **Confidentiality** – Encrypts IP packets
2. **Authentication** – Verifies sender
3. **Integrity** – Ensures packets are unchanged
4. **Access Control**
5. **Replay Protection**

IPsec works for:

- VPN
- Secure site-to-site tunnels
- Remote access security

2.2 IP Security Architecture

IPsec consists of:

1. **AH (Authentication Header)**
2. **ESP (Encapsulating Security Payload)**
3. **Security Associations (SA)**
4. **Key Management (IKE)**

IPsec modes:

- **Transport Mode** (host-to-host)
- **Tunnel Mode** (network-to-network)

---

3. Authentication Header (AH)

AH provides:

- Authentication
- Integrity
- Anti-replay protection

AH **does not provide confidentiality** (no encryption).

AH Packet Fields

- Next Header
- Payload Length
- SPI (Security Parameter Index)
- Sequence Number
- Authentication Data (Integrity Check Value)

AH Protection

AH protects:

- Entire packet (except mutable fields like TTL)

---

4. Encapsulating Security Payload (ESP)

ESP provides:

1. **Confidentiality** (encryption)
2. **Integrity**
3. **Authentication**
4. **Anti-replay**

ESP is more widely used than AH because it **encrypts data**.

ESP Packet Fields

- SPI
- Sequence Number
- Payload Data (Encrypted)
- Padding
- Trailer
- Authentication Data (optional)

Difference: AH vs ESP

Feature	AH	ESP
Encryption	✗ No	✓ Yes
Integrity	✓ Yes	✓ Yes
Authentication	✓ Yes	✓ Yes
Anti-replay	✓ Yes	✓ Yes

Feature	AH	ESP
Usage	Less common	Most common

---

5. Combining Security Associations

Security Association (SA) = a set of security parameters shared between two parties.

5.1 What is SA?

SA defines:

- Encryption algorithm
- Authentication algorithm
- Keys
- Mode (Tunnel/Transport)
- SPI

For each communication direction, a separate SA is needed.

5.2 Types of SA Combinations

1. **Transport AH + Transport ESP**
    - Authentication + Encryption
2. **Transport ESP + Tunnel AH**
3. **Tunnel ESP** (most common for VPN)

---

6. Key Management in IPsec

Key management is performed using **IKE (Internet Key Exchange)**.

IKE Phases

Phase 1

- Establishes secure, authenticated channel
- Uses Diffie–Hellman
- Creates **ISAKMP SA**

Phase 2

- Negotiates IPsec SAs
- Establishes keys for AH/ESP

IKE Modes

- **Main Mode** (more secure)
- **Aggressive Mode** (faster)
- **Quick Mode** (for SA updates)

Key Management Functions

1. Exchange of public keys
2. Negotiation of algorithms
3. Generation of session keys
4. Re-keying

Summary ()

Topic	Key Points
PGP	Hybrid encryption, Web of Trust, digital signatures
S/MIME	PKI-based, certificate-based email security
IPsec	Network layer security framework

Topic	Key Points
AH	Integrity + authentication (no encryption)
ESP	Encryption + integrity + authentication
SA	Defines security parameters for communication
IKE	Key exchange and SA negotiation

WEB SECURITY – FULL DETAILED NOTES

---

1. Web Security: Introduction

Web security focuses on protecting websites, web servers, and web applications from attacks, unauthorized access, data breaches, and misuse.
It ensures confidentiality, integrity, and availability of online services.

Key Web Security Considerations

1. **Confidentiality** – Ensuring sensitive data such as passwords and financial information remains private.
2. **Integrity** – Data should not be modified by unauthorized users.
3. **Availability** – Websites must remain accessible and resilient against DoS/DDoS attacks.
4. **Authentication** – Verifying the identity of users (passwords, biometrics, tokens).
5. **Authorization** – Providing appropriate access based on privileges.
6. **Input Validation** – Preventing injection attacks by validating user-provided data.
7. **Session Management** – Securing session IDs, cookies, and preventing session hijacking.
8. **Secure Communication** – Using encryption protocols like SSL/TLS for secure data transfer.

---

2. Secure Socket Layer (SSL) and Transport Layer Security (TLS)

What is SSL/TLS?

SSL (now replaced by TLS) is a cryptographic protocol used to provide secure communication over the internet.

Goals of SSL/TLS

- **Authentication** of server and optionally the client
- **Confidentiality** through encryption
- **Integrity** through MAC/HMAC
- **Secure key exchange**

SSL/TLS Handshake Steps

1. **ClientHello:**
   Client sends supported cipher suites and random value.
2. **ServerHello:**
   Server responds with selected cipher suite and its certificate.
3. **Server Certificate Verification:**
   Client validates server's digital certificate using CA public key.
4. **Key Exchange:**
   Client and server exchange keys (RSA or Diffie-Hellman).
5. **Session Key Generation:**
   A symmetric session key is generated for encryption.
6. **Secure Communication:**
   All data between client and server is encrypted using AES/3DES.

---

3. Secure Electronic Transaction (SET)

What is SET?

SET is a secure protocol for online credit card payments, developed by Visa and MasterCard.

Objectives

- Secure credit card transactions
- Prevent fraud
- Ensure privacy

- Authenticate cardholder and merchant

Key Features

- Dual signature for separating payment and order information
- Digital certificates for cardholder and merchant
- Confidential payment information

---

4. Intruders

Intruders are individuals or software attempting unauthorized access to a computer system.

Types of Intruders

1. **Masquerader:**
   Gains access by pretending to be an authorized user.
2. **Misfeasor:**
   Legitimate user who abuses their privileges.
3. **Clandestine User:**
   Takes control of the system and hides their activities.

Intruder Techniques

- Password cracking
- Exploiting vulnerabilities
- Spyware and keyloggers
- Rootkits

---

5. Intrusion Detection Systems (IDS)

What is IDS?

IDS monitors network/system activities to detect malicious actions.

Types of IDS

1. **Host-Based IDS (HIDS):**
   Monitors system logs and files.

2. **Network-Based IDS (NIDS):**
   Analyzes network traffic.

IDS Detection Techniques

1. **Signature-based detection:**
   Matches known attack patterns.
2. **Anomaly-based detection:**
   Detects deviations from normal behavior.

Intrusion Prevention System (IPS)

IPS not only detects but also **blocks** malicious activities.

---

6. Password Management

Types of Password Attacks

- Brute force
- Dictionary attack
- Rainbow table attack
- Keylogging
- Shoulder surfing

Password Protection Techniques

- Enforcing complexity
- Using password hashing (SHA-256, bcrypt)
- Multi-factor authentication
- Regular password changes

---

7. Viruses and Related Threats

Virus

A virus attaches itself to programs and replicates by infecting other files.

Other Malware

1. **Worm:**
   Self-replicating and spreads through networks.
2. **Trojan Horse:**
   Appears legitimate but executes malicious activity.
3. **Spyware:**
   Collects user information secretly.
4. **Ransomware:**
   Encrypts data and demands payment.
5. **Logic Bomb:**
   Executes when a specific condition is met.

Virus Phases

1. Dormant
2. Propagation
3. Trigger
4. Execution

---

8. Countermeasures to Malware

- Regular antivirus scanning
- Keeping systems updated
- Email filtering
- Network segmentation
- User training
- Firewalls and IDS

---

9. Firewalls

What is a Firewall?

A firewall is a security system that filters incoming and outgoing traffic based on rules.

Firewall Design Principles

- All traffic must pass through the firewall
- Only authorized traffic is allowed

- Must be resistant to attacks

Types of Firewalls

1. **Packet-Filtering Firewall:**
   Filters based on source/destination IP, port.
2. **Stateful Inspection Firewall:**
   Tracks the state of active connections.
3. **Application-Level Firewall (Proxy Firewall):**
   Filters traffic at application layer (HTTP, FTP).
4. **Next-Generation Firewall (NGFW):**
   Deep packet inspection, IPS, content filtering.

---

10. Case Studies on Cryptography and Security

Case Study 1: Secure Inter-Branch Payment Transactions

Banks use secure channels (e.g., TLS, VPN) to transfer financial data between branches.

Security Measures

- End-to-end encryption
- Digital signatures for transaction validation
- Blockchain for integrity
- OTP-based transaction confirmation
- HSM (Hardware Security Module)

Case Study 2: Cross-Site Scripting (XSS) Vulnerability

XSS allows attackers to inject malicious JavaScript into websites.

Types of XSS

1. **Stored XSS** – Script permanently stored on the server.
2. **Reflected XSS** – Script embedded in URL.
3. **DOM-based XSS** – Manipulated via browser's DOM.

Prevention

- Input validation
- Output encoding
- Disabling inline JavaScript
- Using Content Security Policy (CSP)

Case Study 3: Virtual Elections

Electronic voting systems require high security.

Threats

- Tampering
- Malware in voting machines
- Network attacks
- Data manipulation

Security Solutions

- Strong authentication
- End-to-end encryption
- Blockchain for vote verification
- Physical security of machines
- Independent auditing