**CB3491-Cryptography and Cyber Security**

**PART-B (13 Marks)**

**UNIT-1 Introduction to Security**

**1.Explain the followings: (a) Playfair cipher. (8) (b) Vernam cipher in detail.**

**(a) Playfair Cipher**

The **Playfair Cipher** is a classical encryption technique invented by Charles Wheatstone in 1854 but popularized by Lord Playfair. It is a **digraph substitution cipher**, meaning it encrypts two letters at a time.

**Steps for Encryption:**

1. **Key Square Formation:** A **5×5 matrix** is created using a keyword, removing duplicate letters and combining 'I' and 'J' into one cell.
2. **Message Preparation:** The plaintext is split into digraphs (pairs of two letters). If a pair has the same letter, insert an 'X' between them.
3. **Encryption Rules:**
   * If both letters are in the **same row**, replace each with the letter to its **right** (circular shift).
   * If both are in the **same column**, replace each with the letter **below** (circular shift).
   * If they form a **rectangle**, swap letters diagonally.

**Example:**

**Key:** MONARCHY  
Matrix:

mathematica

CopyEdit

M O N A R

C H Y B D

E F G I/J K

L P Q S T

U V W X Z

Plaintext: **HELLO** → **HE LX LO**  
Ciphertext: **GC BM NA**

**(b) Vernam Cipher**

The **Vernam Cipher**, also known as the **One-Time Pad (OTP)**, is a **perfectly secure** cipher.

**Process:**

1. **Generate a Key:** A **random** key (same length as the plaintext) is used.
2. **Encryption:** Uses **modulo-2 addition (XOR)**.
   * Formula: **C = P ⊕ K**
   * Where:
     + **P** = Plaintext (binary)
     + **K** = Key (binary)
     + **C** = Ciphertext (binary)
3. **Decryption:** Apply XOR again.
   * Formula: **P = C ⊕ K**

**Example:**

Plaintext: **HELLO**  
ASCII (Binary): 01001000 01000101 01001100 01001100 01001111  
Random Key: 11010111 10101010 11011010 11101110 10011011  
Ciphertext: **XOR the two binary values** → Encrypted data.

**Advantages:**

* **Unbreakable** if key is truly random and never reused.

**2. Explain simplified DES with example.**

**Simplified DES (S-DES)** is a lightweight version of DES with **10-bit keys** and **8-bit blocks**.

**Process:**

1. **Key Generation:**
   * Start with a **10-bit key**.
   * Apply **P10 permutation**.
   * Split into two 5-bit halves.
   * Apply **circular left shifts (LS-1, LS-2)**.
   * Apply **P8 permutation** to get two subkeys **K1** and **K2**.
2. **Encryption:**
   * Apply **Initial Permutation (IP)**.
   * Split into two 4-bit halves (Left & Right).
   * Apply **Feistel Function** with **K1**:
     + Expansion/Permutation (E/P)
     + XOR with K1
     + S-Box substitution (S0, S1)
     + P4 permutation
   * Swap halves.
   * Apply **Feistel Function** again with **K2**.
   * Apply **Inverse Permutation (IP⁻¹)** to get ciphertext.
3. **Decryption:** Reverse the process using the same subkeys.

**Example:**

Plaintext: **10101010**, Key: **1010000010**  
Ciphertext after encryption.

**3. Write short notes on i) Steganography**

Steganography is the practice of **hiding information** within another medium, such as images, audio, or text.

**Types of Steganography:**

1. **Image Steganography:** Hide text within image pixels using **LSB (Least Significant Bit)** modification.
2. **Audio Steganography:** Embed data within audio signals using **phase coding** or **echo hiding**.
3. **Video Steganography:** Hide messages in video frames.
4. **Text Steganography:** Modify text formatting or use **invisible characters**.
5. **Network Steganography:** Hide data within network packets.

**Example (LSB in Image):**

* Original Pixel (Binary): 10101100 11001001 10110101
* Hidden Message (010): 10101100 11001000 10110100
* The difference is **imperceptible** to the human eye.

**Advantages:**

* Harder to detect compared to cryptography.
* Can be used for **covert communication**.

**4.** . **Explain classical Encryption techniques in detail.**

Classical encryption techniques include **Substitution Ciphers** and **Transposition Ciphers**.

**(A) Substitution Ciphers:**

1. **Caesar Cipher:** Shift letters by a fixed number (e.g., Shift=3 → A→D, B→E).
2. **Monoalphabetic Cipher:** Uses a random mapping of alphabets.
3. **Playfair Cipher:** Uses **digraph substitution**.
4. **Vigenère Cipher:** Uses a keyword and shifts each letter accordingly.

**(B) Transposition Ciphers:**

1. **Rail Fence Cipher:** Rearranges characters in a zigzag pattern.
   * Example: "HELLO WORLD" → HLOOL ELWRD
2. **Columnar Transposition Cipher:** Rearranges text in columns.
   * Example:

css

CopyEdit

P L A Y

F A I R

C I P H

E R K E

Y Z X Q

Read column-wise to get ciphertext.

**5.Write short notes on (a) Security services(8) (b) Feistel cipher structure(5)**

Security services ensure data protection, confidentiality, and integrity.

**Types of Security Services:**

1. **Confidentiality:** Prevents unauthorized access (e.g., Encryption).
2. **Authentication:** Verifies user identity (e.g., Digital Signatures).
3. **Integrity:** Ensures data is not altered (e.g., Hash Functions).
4. **Non-repudiation:** Prevents denial of transactions (e.g., Digital Signatures).
5. **Access Control:** Restricts access to sensitive information.
6. **Availability:** Ensures system resources are available.

**Example:** **TLS/SSL encryption** in secure web communication.

**5. (b) Feistel Cipher Structure (13 Marks)**

Feistel Cipher is the foundation of many modern encryption algorithms (e.g., DES, Blowfish).

**Structure:**

1. **Splitting:** Plaintext is divided into two halves (L₀, R₀).
2. **Rounds:** Several rounds of encryption using:
   * **Round Function (F)**: Uses **subkeys (K)** and operations like XOR and substitution.
   * **Swapping:** Left and right halves are swapped.
3. **Final Combination:** After multiple rounds, recombine L and R.

**Mathematical Representation:**

Each round:

* Li+1=RiL\_{i+1} = R\_iLi+1​=Ri​
* Ri+1=Li⊕F(Ri,Ki)R\_{i+1} = L\_i \oplus F(R\_i, K\_i)Ri+1​=Li​⊕F(Ri​,Ki​)

**Example (DES):**

* 16 rounds of Feistel operations.
* Uses **S-boxes** and **XOR operations**.

**Advantages:**

* Allows encryption and decryption to use the same structure (only subkeys differ).

**UNIT-II SYMMETRIC CIPHERS**

**1.State and explain the principles of public key cryptography?**

**Principles of Public Key Cryptography (13 Marks)**

Public Key Cryptography (PKC) uses **two mathematically linked keys**:

* **Public Key (Shared)**
* **Private Key (Secret)**

**Principles:**

1. **Key Pair Generation:** Two keys are created (e.g., RSA keys).
2. **Encryption & Decryption:**
   * Sender encrypts with the **recipient's public key**.
   * Recipient decrypts using their **private key**.
3. **Authentication (Digital Signatures):**
   * Sender **signs** a message with their **private key**.
   * Receiver **verifies** using the **public key**.
4. **Mathematical Complexity:** Security relies on **factorization (RSA) or discrete logarithms (Diffie-Hellman, ECC)**.

**Example:** Used in **SSL/TLS, PGP encryption, and Bitcoin transactions**.

**2.Explain Diffie Hellman key Exchange in detail with an example?**

Diffie-Hellman (DH) is a method for **securely exchanging a shared key** over an insecure channel.

**Process:**

1. **Select a prime number (p) and a base (g).**
2. **Key Generation:**
   * Alice picks **private key (a)** and computes A=gamod  pA = g^a \mod pA=gamodp.
   * Bob picks **private key (b)** and computes B=gbmod  pB = g^b \mod pB=gbmodp.
3. **Key Exchange:** Alice sends **A**, Bob sends **B**.
4. **Key Computation:**
   * Alice computes K=Bamod  pK = B^a \mod pK=Bamodp.
   * Bob computes K=Abmod  pK = A^b \mod pK=Abmodp.
   * Since K=gabmod  pK = g^{ab} \mod pK=gabmodp, they now share the same key.

**Example:**

* **p = 23, g = 5**
* Alice chooses **a = 6** → A=56mod  23=8A = 5^6 \mod 23 = 8A=56mod23=8
* Bob chooses **b = 15** → B=515mod  23=19B = 5^{15} \mod 23 = 19B=515mod23=19
* Shared Key: K=196mod  23=2K = 19^6 \mod 23 = 2K=196mod23=2

**Use Case:** Used in **TLS, VPNs, and secure messaging**.

**3.Explain the key management of public key encryption in detail?**

Managing public-key cryptography involves handling keys securely.

**Key Management Functions:**

1. **Key Generation:** Creating strong key pairs (RSA, ECC).
2. **Key Distribution:**
   * **Public keys** can be shared openly.
   * **Private keys** must be stored securely.
3. **Key Storage:**
   * Use **Hardware Security Modules (HSMs)** or **smart cards**.
4. **Key Revocation:** If a key is compromised, it must be revoked (Certificate Revocation Lists).
5. **Key Renewal:** Periodic key updates enhance security.

**Example:** **PKI (Public Key Infrastructure)** manages keys using **Certificate Authorities (CAs)**.

**4.Explain RSA algorithm in detail with an example?**

RSA is a widely used **asymmetric encryption algorithm** based on **prime factorization**.

**Steps:**

1. **Key Generation:**
   * Select two large primes **p, q**.
   * Compute **n = p × q** and **φ(n) = (p-1)(q-1)**.
   * Choose **e (public key)** such that **1 < e < φ(n) and gcd(e, φ(n)) = 1**.
   * Compute **d (private key)** as **d ≡ e⁻¹ mod φ(n)**.
2. **Encryption:**
   * Ciphertext **C = M^e mod n**.
3. **Decryption:**
   * Plaintext **M = C^d mod n**.

**Example:**

* **p = 3, q = 11** → **n = 33, φ(n) = 20**
* Choose **e = 7** → Compute **d = 3** (since **7 × 3 ≡ 1 mod 20**)
* Encrypt **M = 4** → C=47mod  33=16C = 4^7 \mod 33 = 16C=47mod33=16
* Decrypt **C = 16** → M=163mod  33=4M = 16^3 \mod 33 = 4M=163mod33=4

**Use Case:** Used in **secure emails, digital signatures, and SSL/TLS**.

**5.Briefly explain the idea behind Elliptic Curve Cryptosystem?**

ECC provides **strong security** with **smaller keys** compared to RSA.

**Idea Behind ECC:**

* Uses **elliptic curves** over a finite field:
  + **Equation:** y2=x3+ax+bmod  py^2 = x^3 + ax + b \mod py2=x3+ax+bmodp
* Instead of multiplication (RSA), ECC relies on **point addition & scalar multiplication**.
* The **Elliptic Curve Discrete Logarithm Problem (ECDLP)** makes ECC highly secure.

**Advantages of ECC:**

1. **Smaller Key Size:** 256-bit ECC is as secure as 3072-bit RSA.
2. **Faster Computation:** Uses **fewer resources**, ideal for mobile devices.
3. **Stronger Security:** Harder to break due to **ECDLP**.

**Use Case:** Used in **Bitcoin, Signal App, and TLS encryption**.

**UNIT-III ASYMMETRIC CIPHERS**

**1.Explain Data Encryption Standard (DES) in detail.**

DES is a **block cipher** that encrypts **64-bit plaintext blocks** using a **56-bit key** in **16 rounds**.

**Steps in DES:**

1. **Initial Permutation (IP):** Reorders the 64-bit plaintext.
2. **Key Generation:**
   * 56-bit key → **Divided into two 28-bit halves**
   * Left shift + **PC-2 permutation** → Generates 16 subkeys.
3. **16 Feistel Rounds:**
   * Each round **splits data** into Left (L) and Right (R).
   * Right half goes through **Feistel function (F):**
     + **Expansion (E):** 32 → 48 bits
     + **XOR with subkey**
     + **S-Box Substitution:** 48 → 32 bits
     + **Permutation (P)**
   * Left part is XORed with F(R) and swapped.
4. **Final Permutation (IP⁻¹):** Produces the ciphertext.

**Decryption:**

* **Same process** but subkeys are used in **reverse order**.

**Example:**

* **Plaintext:** "HELLO" (binary form)
* **Key:** 56-bit
* **Ciphertext:** Encrypted binary output

**Use Case:** Used in banking but replaced by AES due to security concerns.

2.**Explain the key management of public key encryption in detail.**

Public Key Encryption requires **secure key management** for maintaining confidentiality.

**Key Management Components:**

1. **Key Generation:**
   * RSA, ECC, and DH generate **asymmetric key pairs**.
2. **Key Distribution:**
   * Public keys are shared openly, while private keys remain secret.
   * **Public Key Infrastructure (PKI)** issues and verifies keys.
3. **Key Storage:**
   * **Hardware Security Modules (HSMs)**, smart cards, or software vaults store keys securely.
4. **Key Revocation & Expiry:**
   * If compromised, keys are **revoked** using **Certificate Revocation Lists (CRLs)**.
5. **Key Renewal:**
   * Keys are periodically **regenerated** to enhance security.
6. **Key Backup & Recovery:**
   * Prevents **data loss** due to accidental key deletion.

**Example:** **SSL/TLS** encryption in websites uses **PKI for key management**.

**4.Explain ECC - Diffie Hellman key Exchange with both keys in detail with an example**

ECC-DH is a **secure key exchange protocol** based on **Elliptic Curve Cryptography (ECC)**.

**Key Exchange Process:**

1. **Select Curve Parameters:**
   * Choose **elliptic curve equation**: y2=x3+ax+bmod  py^2 = x^3 + ax + b \mod py2=x3+ax+bmodp.
   * Select **base point (G)**.
2. **Key Generation:**
   * Alice picks **private key (a)** → Computes **public key A = aG**.
   * Bob picks **private key (b)** → Computes **public key B = bG**.
3. **Key Exchange:**
   * Alice sends **A** to Bob, Bob sends **B** to Alice.
4. **Shared Key Computation:**
   * Alice computes K=aB=abGK = aB = abGK=aB=abG.
   * Bob computes K=bA=abGK = bA = abGK=bA=abG.
   * Since both computed the same value, KKK becomes the shared secret.

**Example:**

* **Curve:** y2=x3+2x+3mod  97y^2 = x^3 + 2x + 3 \mod 97y2=x3+2x+3mod97
* **Base point:** G=(3,6)G = (3, 6)G=(3,6)
* **Private keys:** a=5,b=7a = 5, b = 7a=5,b=7
* **Public keys:**
  + Alice → A=5GA = 5GA=5G
  + Bob → B=7GB = 7GB=7G
* **Shared Secret:** K=abG=35GK = abG = 35GK=abG=35G

**Use Case:** **Secure messaging (Signal, WhatsApp), TLS, VPNs**.

**4.Write about elliptic curve architecture in detail and how they are useful for cryptography.**

ECC is based on **elliptic curves** and provides **high security with smaller keys**.

**Elliptic Curve Structure:**

* Defined by **equation**: y2=x3+ax+bmod  py^2 = x^3 + ax + b \mod py2=x3+ax+bmodp.
* **Operations on the curve:**
  1. **Point Addition:** Adding two points on the curve.
  2. **Point Doubling:** Doubling a point.
  3. **Scalar Multiplication:** Repeated addition of a point.

**Why ECC is Useful for Cryptography?**

1. **Smaller Key Sizes:**
   * 256-bit ECC = **3072-bit RSA security**.
2. **Faster Computation:**
   * Less resource-intensive than RSA.
3. **Stronger Security:**
   * Relies on **Elliptic Curve Discrete Logarithm Problem (ECDLP)**, harder to break.
4. **Efficiency in Key Exchange & Digital Signatures:**
   * Used in **ECDSA (Elliptic Curve Digital Signature Algorithm)**.

**Applications of ECC:**

* **Blockchain (Bitcoin, Ethereum)**
* **TLS/SSL certificates**
* **Secure IoT communications**

**5.Write about key distribution in detail**

Key distribution ensures **secure sharing of encryption keys** between parties.

**Methods of Key Distribution:**

1. **Manual Distribution:**
   * Physically delivering keys (e.g., USB drive).
2. **Public Key Infrastructure (PKI):**
   * Uses **Certificate Authorities (CAs)** to manage public keys.
3. **Diffie-Hellman Key Exchange:**
   * Securely establishes a **shared secret** over an insecure network.
4. **Kerberos Authentication:**
   * Uses a **trusted third party (KDC)** to distribute symmetric keys.
5. **Quantum Key Distribution (QKD):**
   * Uses **quantum mechanics** for unbreakable security.

**Challenges in Key Distribution:**

* **Man-in-the-middle attacks:** Prevented by **digital certificates**.
* **Key Compromise:** Requires **key rotation and revocation**.
* **Scalability Issues:** Solved using **hierarchical PKI**.

**Example:** TLS uses **Diffie-Hellman + PKI** for key distribution.

**UNIT-IV INTEGRITY AND AUTHENTICATION ALGORITHMS**

**1.Write and explain the digital signature algorithm.**

**Digital Signature Algorithm (DSA) (13 Marks)**

DSA is an asymmetric cryptographic algorithm used for **authentication and integrity**.

**Steps of DSA:**

1. **Key Generation:**
   * Choose a **prime number (p)** and a **generator (g)**.
   * Select a **private key (x)**.
   * Compute **public key (y) = g^x mod p**.
2. **Signing Process:**
   * Compute **hash (H)** of the message.
   * Generate a random number **k** and compute **r = (g^k mod p) mod q**.
   * Compute **s = (H + xr) k⁻¹ mod q**.
3. **Verification:**
   * Compute **w = s⁻¹ mod q**.
   * Compute **u1 = H × w mod q** and **u2 = r × w mod q**.
   * Compute **v = (g^u1 \* y^u2 mod p) mod q**.
   * If **v = r**, the signature is valid.

**Use Case:** Used in **SSL, Digital Certificates, E-Governance**.

**2. Explain in detail Hash Functions.**

A hash function converts input data into a **fixed-length output (digest)**.

**Properties of Hash Functions:**

1. **Deterministic:** Same input → Same output.
2. **Fast Computation:** Generates hash quickly.
3. **Preimage Resistance:** Hard to find input from hash.
4. **Small Change Resistance:** Minor changes in input → Completely different hash.
5. **Collision Resistance:** No two different inputs should have the same hash.

**Examples:**

* **MD5 (128-bit)**
* **SHA-256 (256-bit)**

**Use Case:** Digital signatures, password storage, blockchain.

**3.(i) Compare the Features of SHA-1 and MD5 algorithm. (8)**

**(ii) Discuss about the objectives of HMAC and it security features. (5)**

| **Feature** | **SHA-1** | **MD5** |
| --- | --- | --- |
| **Digest Size** | 160-bit | 128-bit |
| **Speed** | Slower | Faster |
| **Security** | Weak (collision attacks) | Weaker (broken) |
| **Usage** | Digital signatures | File integrity check |

**(ii) HMAC Objectives and Security Features (8 Marks)**

**HMAC (Hashed Message Authentication Code)** is used for **message integrity & authentication**.

**Objectives:**

1. Prevents **message tampering**.
2. Uses a **shared secret key** for verification.
3. **Faster than asymmetric cryptography**.

**Security Features:**

1. **Resistant to length extension attacks.**
2. **Uses a cryptographic hash function (e.g., SHA-256).**
3. **Only parties with the secret key can verify authenticity.**

**Use Case:** **TLS, API authentication, blockchain.**

**4. Describe the MD5 message digest algorithm with necessary block diagrams.**

MD5 generates a **128-bit hash** from an input message.

**Steps of MD5:**

1. **Padding:** Message is padded to **512-bit blocks**.
2. **Initialize Buffers:** Uses **A, B, C, D (32-bit registers)**.
3. **Processing Blocks:**
   * Each block goes through **64 rounds** using bitwise operations.
4. **Final Output:** Concatenates A, B, C, D → **128-bit hash**.

**Block Diagram:**

**[Message] → [Padding] → [64 Rounds] → [Hash Output]**

**Use Case:** Was used in **password hashing & checksums** but is now insecure.

**5.(i) Draw the general structure of DES and explain the encryption decryption process. (8)**

**(ii) Mention the strengths and weakness of DES algorithm.**

DES is a **Feistel cipher** with **16 rounds** of encryption.

**Encryption Process:**

1. **Initial Permutation (IP).**
2. **Divide plaintext into L and R.**
3. **16 Feistel Rounds:**
   * Expand R → XOR with key → Apply S-box → Permutation.
4. **Final Permutation (IP⁻¹).**

**Decryption:**

* **Same process, but subkeys are used in reverse order.**

**(ii) Strengths & Weaknesses of DES (6 Marks)**

**Strengths:**

✔ Simple & fast implementation.  
✔ Uses **Feistel structure**, allowing easy decryption.  
✔ **Better than substitution ciphers**.

**Weaknesses:**

❌ **56-bit key is too small** (vulnerable to brute-force).  
❌ **Not resistant to modern cryptanalysis attacks**.  
❌ **Replaced by AES.**

**Use Case:** Banking transactions (now replaced by AES).

**UNIT-V CYBER CRIMES AND CYBER SECURITY**

**1.Assume when an attacker tries to modify the database content by inserting an UPDATE statement. Identify this SQL injection attack method and modify and justify. Detail the methods used to prevent SQL injection attack.**

SQL injection occurs when an attacker **injects malicious SQL queries** to manipulate a database.

**Example Attack:**

sql

CopyEdit

Input: ' OR '1'='1'; --

Query: SELECT \* FROM users WHERE username = '' OR '1'='1';

**Impact:** Returns all records, bypassing authentication.

**Types of SQL Injection:**

1. **Error-Based Injection:** Uses error messages to extract data.
2. **Union-Based Injection:** Merges malicious results using UNION.
3. **Blind SQL Injection:** No visible output, uses true/false conditions.

**Prevention Methods:**

✔ **Use Prepared Statements (Parameterized Queries):**

python

CopyEdit

cursor.execute("SELECT \* FROM users WHERE username = ? AND password = ?", (user, pwd))

✔ **Use ORM (e.g., SQLAlchemy, Hibernate).**  
✔ **Validate User Input & Escape Special Characters.**  
✔ **Use Web Application Firewalls (WAFs).**  
✔ **Limit Database Privileges & Disable Error Messages.**

**2. What is keylogger? Explain types of keyloggers with its advantages and disadvantages.**

A **keylogger** is malware that records keystrokes to steal sensitive data.

**Types of Keyloggers:**

1. **Hardware Keylogger:**
   * **USB or keyboard circuit-based.**
   * **Advantage:** Hard to detect.
   * **Disadvantage:** Physical access required.
2. **Software Keylogger:**
   * Installed on a system to record keystrokes.
   * **Advantage:** Can be remotely installed.
   * **Disadvantage:** Detectable by antivirus.
3. **Rootkit Keylogger:**
   * Embedded in the OS to evade detection.
   * **Advantage:** Stealthy and hard to remove.
   * **Disadvantage:** Requires advanced knowledge to develop.
4. **Clipboard & Screen Recorders:**
   * Captures copied text and screenshots.

**Prevention:**

✔ Use **antivirus & anti-malware tools**.  
✔ Enable **two-factor authentication (2FA)**.  
✔ Use **virtual keyboards** for sensitive input.

**3. Write short notes on network access control.**

NAC is a **security framework** that regulates device access to a network.

**Key Features:**

1. **Authentication & Authorization:**
   * Uses **passwords, biometrics, or certificates**.
2. **Endpoint Security Checks:**
   * Ensures devices comply with security policies.
3. **Role-Based Access Control:**
   * Limits network access based on user roles.
4. **Quarantine Non-Compliant Devices:**
   * Prevents infected devices from joining the network.

**Types of NAC:**

1. **Agent-Based NAC:** Installed software checks security status.
2. **Agentless NAC:** Uses network-based authentication.
3. **Inline vs. Out-of-Band NAC:** Direct control vs. monitoring mode.

✔ **Example:** Used in **corporate networks, banking systems**.

**4. What is cybercrime? Explain the types of cybercrime**

Cybercrime refers to **criminal activities involving computers and networks**.

**Types of Cybercrime:**

1. **Hacking:** Unauthorized access to systems.
2. **Phishing:** Fraudulent emails to steal credentials.
3. **Ransomware:** Encrypts data and demands ransom.
4. **Identity Theft:** Stealing personal data for fraud.
5. **Cyberbullying:** Harassment via digital platforms.
6. **DoS/DDoS Attacks:** Overloading systems to crash them.
7. **Financial Fraud:** Online scams & card cloning.

**Prevention:**

✔ Use **strong passwords** and 2FA.  
✔ **Keep software updated** and use **firewalls**.  
✔ **Avoid suspicious emails & links**.

**5. What is web security? Discuss web security issues. Explain transport layer security.**

Web security protects **websites & web applications** from cyber threats.

**Web Security Issues:**

1. **Cross-Site Scripting (XSS):** Injects malicious scripts into web pages.
2. **Cross-Site Request Forgery (CSRF):** Forces users to perform unintended actions.
3. **SQL Injection:** Modifies database queries.
4. **Broken Authentication:** Weak session management.
5. **Insecure APIs:** Leads to data breaches.

**Transport Layer Security (TLS):**

* Secures **internet communication via encryption**.
* Uses **handshake protocol** for authentication.
* Replaced **SSL** due to security flaws.

✔ **Example:** HTTPS (TLS + HTTP) secures websites.