Network and information security

```
This content is an analyzed version of Sakib Mahmud's personal notes, generated
using GPT-5 and OpenAI's agentic search tools. To support NU CSE students, please
consider contributing to the repository: https://github.com/Sigmakib2/Nu-CSE-Notes
```

1. Introduction

- Define cryptography
- Computer network security, network security, internet security
- Wireless network components and their purpose
- Cryptography as a security tool
- Network security definition
- Network security model (draw/explain)
- Types of security services
- Types of attacks (on plaintext/message, security attack definitions, examples)
- Plaintext → ciphertext conversion
- Active vs. passive attacks
- Traditional methods for network security
- OSI security model
- Data security & OSI security services
- Threat vs. attack
- Three dimensions of cryptography
- Information security & CIA triad
- Remote access technologies and vulnerabilities
- Accessibility in information security
- E-commerce security

---

2. Traditional Symmetric-Key Ciphers

- Encryption definition
- Symmetric cipher model
- Cryptanalysis and brute force attacks
- Types of cryptanalysis attacks (with diagrams)
- Attacks on conventional encryption scheme
- Principles of public key cryptography
- Public vs. private key cryptography comparison
- Monoalphabetic vs. polyalphabetic cipher
- Unconditionally secure vs. computationally secure cipher
- Plaintext → ciphertext conversion
- Transposition cipher
- Playfair cipher rules/examples

- Applications of public key cryptosystem
- AAA (Authentication, Authorization, Accounting)
- Playfair cipher example with keyword
- Middle-man attack

---

3. Modern Symmetric-Key Ciphers

- Block cipher
- Block cipher modes & convenience
- Block vs. stream cipher
- Modes using encryption only vs. both encryption & decryption
- One-time pad
- Confusion and diffusion
- Feistel cipher (operations, structure, use in DES)
- Steganography
- Cipher feedback model

---

4. Data Encryption Standard (DES)

- Definition & features of DES
- General DES encryption process with diagram
- Merits & demerits of DES
- Strength of DES
- Single round DES architecture
- Triple encryption (3DES)
- Why middle portion of triple DES is decryption

---

5. Advanced Encryption Standard (AES)

- AES evaluation criteria (NIST)
- Features of AES
- Rijndael vs. AES
- AES decryption vs. equivalent inverse cipher
- AES vs. DES
- AES encryption & decryption structure

---

6. Asymmetric-Key Cryptography

- Meaning of asymmetric key
- Symmetric vs. asymmetric cipher model
- Public key vs. conventional encryption
- Symmetric vs. asymmetric techniques (differences)
- Link vs. end-to-end encryption
- Advantages & disadvantages of symmetric/asymmetric encryption
- RSA algorithm (encryption/decryption, with examples)

- Elliptic Curve Cryptography (ECC)
- Attacks on RSA & responses
- Public key cryptography for encryption & authentication
- Requirements for a secure public key cryptosystem
- Public key algorithms & diagrams

---

7. Message Integrity & Authentication

- Message Authentication (MAC)
- Classes of authentication functions
- Secure use of conventional encryption
- Requirements for message authentication
- MD5 vs. SHA comparison
- MD5 algorithm in detail vs. SHA-1
- Approaches to message authentication
- Hash function & SHA-512 logic
- Secure algorithms
- Digital signature for authentication

---

8. Cryptographic Hash Functions

- Hash function & requirements
- Weak vs. strong collision resistance
- MAC vs. hash function
- Role of compression function
- Structure of secure hash function
- Block cipher modes of operation

---

9. Key Management

- Public key distribution schemes
- Kerberos & Kerberos V4
- Keyed cryptography & key types
- Key Distribution Center (KDC)
- Public Key Infrastructure (PKI) & necessity
- Session key vs. master key
- Man-in-the-middle attack
- Certificate revocation
- X.509 certificate contents & revocation lists
- Purpose of X.509 standard
- Applications of IP security
- Kerberos requirements

---

10. Digital Signature

- Definition of digital signature
- Requirements for digital signature
- Properties of digital signatures
- DSA algorithm
- Digital Signature Standard (DSS)
- Direct vs. arbitrated signatures
- RSA for digital signatures
- Digital signature procedure & diagrams
- RSA vs. DSA
- RSA digital signature scheme

---

11. Entity Authentication

- Password concerns (3 main issues)
- Social engineering attack on password
- Classification of password attacks

---

12. Application Layer Security (PGP & S/MIME)

- PGP definition
- MIME attack on Diffie-Hellman
- MIME definition & SMTP limitations
- E-mail compatibility in PGP

---

13. Transport Layer Security (SSL & TLS)

- Diffie-Hellman key exchange algorithm
- SSL algorithm & sockets
- Benefits of SSL
- Diffie-Hellman key exchange examples (numerical)
- SSL handshake protocol
- SSL connection state parameters
- SSH protocol

---

14. Network Layer Security (IPSec)

- Security Association (SA)
- IPSec ESP format
- IPSec applications & benefits
- Tunnel mode vs. transport mode
- IPSec protocol for authentication & integrity

---

15. System Security

- Intrusion & intrusion detection methods

- Firewall (definition, merits, demerits, limitations)
- Worms & digital immune system
- Multiple firewalls in corporations
- Router security

---

16. Random Number Generator

- Pseudorandom generators (definition, example, working)

---

17. Secured Electronic Transaction (SET)

- Definition & features of SET
- Steps of SET transaction

---

18. Encipherment Using Modern Symmetric Ciphers

- Weakness of Electronic Code Book (ECB) mode

---

19. Web Security

- Sandbox & sandbox environments
- Benefits of sandboxing
- Intruder in network security
- DoS & DDoS attacks
- VPN & network security policy/management

---

20. Short Notes

- S/MIME
- Email Security
- ESP
- Steganography
- ECC
- Digital Immune System
- Diffie-Hellman key exchange
- Stream vs. Block cipher
- DSS
- Security attacks
- UNIX password scheme
- SSL
- Hash function
- IPSec ESP format
- Cryptanalysis
- Differential cryptanalysis
- PGP

- SET
- Feistel cipher
- RC4 algorithm
- PRNG
- Wire pool
- X.509 architecture
- PKI
- Generic encryption
- S-Box
- 9-Box

Abbreviations Found in the Syllabus

- **AAA** – Authentication, Authorization, Accounting
- **AES** – Advanced Encryption Standard
- **CIA** – Confidentiality, Integrity, Availability (Triad)
- **CRL** – Certificate Revocation List
- **DES** – Data Encryption Standard
- **DDoS** – Distributed Denial of Service
- **DSS** – Digital Signature Standard
- **DSA** – Digital Signature Algorithm
- **ECC** – Elliptic Curve Cryptography
- **ECB** – Electronic Code Book (mode of operation)
- **ESP** – Encapsulating Security Payload (in IPSec)
- **IPSec** – Internet Protocol Security
- **KDC** – Key Distribution Center
- **MAC** – Message Authentication Code
- **MD5** – Message Digest 5
- **MIME** – Multipurpose Internet Mail Extensions
- **OSI** – Open Systems Interconnection
- **PGP** – Pretty Good Privacy
- **PKI** – Public Key Infrastructure
- **PRNG** – Pseudo Random Number Generator
- **RSA** – Rivest, Shamir, Adleman (asymmetric algorithm)
- **SET** – Secure Electronic Transaction
- **SHA** – Secure Hash Algorithm
- **S/MIME** – Secure/Multipurpose Internet Mail Extensions
- **SMTP** – Simple Mail Transfer Protocol
- **SSH** – Secure Shell
- **SSL** – Secure Socket Layer
- **TLS** – Transport Layer Security
- **VPN** – Virtual Private Network
- **X.509** – ITU-T standard for PKI certificates

📖 Near-Relevant Abbreviations (not in Syllabus but important for prep)

- **2FA** – Two Factor Authentication
- **AES-CTR / CBC / CFB / OFB** – AES modes (Counter, Cipher Block Chaining, Cipher Feedback, Output Feedback)
- **CA** – Certificate Authority
- **CSR** – Certificate Signing Request
- **DH** – Diffie-Hellman
- **DoS** – Denial of Service
- **HMAC** – Hash-based Message Authentication Code
- **IDS** – Intrusion Detection System
- **IPS** – Intrusion Prevention System
- **MITM** – Man in the Middle (attack)
- **OTP** – One-Time Pad / One-Time Password (context dependent)
- **PBKDF2** – Password-Based Key Derivation Function 2
- **RFC** – Request For Comments (IETF standard documents)
- **SSL/TLS Handshake** – Protocol exchange steps for secure connection
- **WEP** – Wired Equivalent Privacy (old Wi-Fi security)
- **WPA / WPA2 / WPA3** – Wi-Fi Protected Access versions
- **HTTPS** – Hyper Text Transfer Protocol Secure
- **DNSSEC** – Domain Name System Security Extensions
- **IP** – Internet Protocol
- **URL** – Uniform Resource Locator

📔 Topic 1: Introduction (Detailed Notes + Google Search as Backup)

---

1. Cryptography

Cryptography is the practice and study of securing communication so that only intended parties can understand the message. It transforms readable data (**plaintext**) into unreadable form (**ciphertext**) using encryption algorithms and secret keys.

- **Purpose:** Protect confidentiality, integrity, authenticity, and non-repudiation.
- **Encryption:** Process of converting plaintext → ciphertext.
- **Decryption:** Reverse process using a secret key.
- **Uses:** Online banking, e-commerce, secure emails, VPNs.
- **Example:** Caesar Cipher shifts letters to hide meaning.

🔗 Google Search

---

2. Computer Network Security / Network Security / Internet Security

- **Computer Network Security:** Ensures protection of transmitted data from hackers, viruses, unauthorized access. Example: firewall protection on office LAN.
- **Network Security:** A broader discipline that involves securing the entire network infrastructure (routers, switches, servers). Includes policies, tools, and monitoring systems.
- **Internet Security:** Specifically addresses threats originating from the Internet (phishing, DoS/DDoS, malware). Tools: antivirus, SSL/TLS for websites, intrusion detection.

🔗 Google Search

---

3. Wireless Network Components and Purpose

Wireless networks remove physical cables and provide mobility, but this increases vulnerability.

- **Access Point (AP):** Connects wireless clients to the wired network.
- **Wireless Clients:** Laptops, smartphones connecting via Wi-Fi.
- **Authentication Server:** Manages who can connect.
- **Router/Firewall:** Provides routing and protects traffic.
- **Purpose:** Provide flexible, mobile connectivity. **Vulnerability:** Prone to eavesdropping, unauthorized access if weak encryption is used (e.g., WEP).

🔗 Google Search

---

4. Cryptography as a Security Tool

Cryptography ensures:

- **Confidentiality:** Only authorized parties read the message.
- **Integrity:** Prevent unauthorized modification.
- **Authentication:** Verifies identity of sender/receiver.
- **Non-repudiation:** Prevents denial of sending a message.
- **Application:** E-commerce (credit card safety), email security, digital signatures, VPN tunnels. Without cryptography, secure communication over open networks (like the Internet) would be impossible.

🔗 Google Search

---

5. Network Security Definition

Network security is the practice of protecting a network and its resources from threats such as unauthorized access, modification, or denial of service. It includes hardware (firewalls, IDS/IPS), software (antivirus, authentication tools), and administrative policies.

🔗 Google Search

---

6. Network Security Model

A conceptual framework showing how security services, mechanisms, and attacks fit together:

- **Sender → Encryption → Transmission Channel → Decryption → Receiver**

- **Security Services:** Confidentiality, authentication, integrity.
- **Security Mechanisms:** Cryptography, firewalls, MACs, hash functions.
- **Attacks:** Active (altering) and Passive (eavesdropping). This model helps visualize how data is protected during communication.

📎 Google Search

---

7. Types of Security Services

- **Confidentiality:** Prevent unauthorized access.
- **Integrity:** Prevent unauthorized modification.
- **Authentication:** Verify source/destination identity.
- **Non-repudiation:** Prevent sender from denying transmission.
- **Access Control:** Limit access to resources.
- **Availability:** Ensure system/data is available when needed. These services form the **backbone of secure systems** and are supported by cryptography and protocols.

📎 Google Search

---

8. Types of Security Attacks

- **Passive:** Only monitoring/reading data without altering (e.g., traffic analysis, packet sniffing). Threatens confidentiality.
- **Active:** Modification, fabrication, or disruption of messages (e.g., DoS, man-in-the-middle, replay attacks). Threatens integrity and availability. **Example:**
- Passive → Hacker intercepts your Wi-Fi traffic.
- Active → Hacker modifies the packets and injects false data.

📎 Google Search

---

9. Plaintext → Ciphertext Conversion

The process of encryption converts **plaintext (readable)** into **ciphertext (unreadable)** using an algorithm + key.

- **Example:** Caesar Cipher shifting letters: "HELLO" → "KHOOR".
- **Modern Example:** AES encryption with a 128-bit key. Conversion ensures even if intercepted, data cannot be understood without the decryption key.

📎 Google Search

---

10. Active vs. Passive Attacks

- **Active Attacks:** Modify data streams, create false data. Examples: DoS, session hijacking, message modification.
- **Passive Attacks:** Only observe or monitor. Examples: traffic analysis, eavesdropping.
- **Key Difference:** Passive affects confidentiality only, Active affects integrity & availability too.

📎 Google Search

---

11. Traditional Methods for Network Security

Before modern cryptography:

- **Passwords & User IDs** – Basic access control.
- **Firewalls** – First line of defense, blocking unwanted traffic.
- **Antivirus Software** – Protect against malware.
- **Physical Security** – Locking access to devices. These were limited, as they didn't protect against sophisticated Internet-based attacks.

📎 Google Search

---

12. OSI Security Model

- Security services (authentication, integrity, confidentiality) are mapped to OSI layers.
- **Goal:** Provide a standard framework.
- Example: Layer 3 (Network) may use IPSec for confidentiality, Layer 4 (Transport) may use SSL/TLS for authentication and encryption.

📎 Google Search

---

13. Data Security & OSI Security Services

- **Data Security:** Protects data from unauthorized access, modification, loss.

- **OSI Security Services:**

  - Authentication
  - Access Control
  - Data Confidentiality
  - Data Integrity
  - Non-repudiation

📎 Google Search

---

14. Threat vs. Attack

- **Threat:** Potential event that can exploit vulnerability (possibility).
- **Attack:** Actual attempt to exploit a vulnerability (execution). **Example:** A weak password is a **threat**. A hacker exploiting it is an **attack**.

📎 Google Search

---

15. Three Dimensions of Cryptography

- **Type of Operation:** Substitution, Transposition, Product cipher.
- **Number of Keys:** Symmetric (1 key), Asymmetric (2 keys).
- **Processing of Plaintext:** Block ciphers (chunks of data), Stream ciphers (bit by bit). This helps classify cryptographic algorithms.

📎 Google Search

---

16. Information Security

- Encompasses all security measures for protecting **data in all forms** (digital, physical, paper-based).
- **Goal:** Protect Confidentiality, Integrity, Availability (CIA).
- Covers people, processes, and technology.

📎 Google Search

---

17. CIA Triad

- **Confidentiality:** Data secrecy.
- **Integrity:** Data accuracy and trustworthiness.
- **Availability:** Ensuring access when needed. Forms the **core principle of security policies** worldwide.

📎 Google Search

---

18. Remote Access Technologies

- Allow employees to connect remotely to secure internal systems.
- **Examples:** VPN, SSH, RDP.
- Essential for business continuity, but needs encryption and authentication.

📎 Google Search

---

19. Vulnerabilities in Remote Access

- Weak authentication (password reuse, weak keys).
- Poorly configured VPN.
- Lack of encryption leading to MITM attacks.
- Malware on remote device can compromise entire network.

📎 Google Search

---

20. Accessibility in Information Security

- Balancing security controls with ease of access.
- Example: Two-factor authentication improves security but may reduce usability.
- Related to **availability** in the CIA triad.

📎 Google Search

---

21. Security for E-commerce

- Uses multiple technologies to protect online transactions.
- **Encryption (SSL/TLS):** Secures communication.
- **Digital Signatures:** Verify authenticity.
- **SET Protocol:** Provides secure card transactions.
- **Firewalls & IDS:** Block external attacks.

📎 Google Search

📓 Topic 2: Traditional Symmetric-Key Ciphers

---

1. Define Encryption

Encryption is the process of converting **plaintext** (readable message) into **ciphertext** (unreadable form) using an algorithm and a secret key. It prevents unauthorized access during data transmission or storage.

- **Types of encryption:**
  - **Symmetric encryption:** Same key for encryption and decryption. Example: DES, AES.
  - **Asymmetric encryption:** Public and private keys. Example: RSA.
- **Goal:** Ensure confidentiality and integrity.

📎 Google Search

---

2. Symmetric Cipher Model

A symmetric cipher model uses a **single secret key** for both encryption and decryption.

- **Ingredients:**
  - **Plaintext:** Original data.
  - **Encryption Algorithm:** Performs substitutions and transformations.
  - **Secret Key:** Shared by sender and receiver.
  - **Ciphertext:** Output after encryption.
  - **Decryption Algorithm:** Uses the same secret key to convert ciphertext back to plaintext.
- **Weakness:** Key distribution is difficult.

📎 Google Search

---

3. Cryptanalysis and Brute Force Attacks

- **Cryptanalysis:** Techniques for breaking ciphers by exploiting weaknesses in algorithms (e.g., frequency analysis).

- **Brute Force Attack:** Trying **all possible keys** until the correct one is found.
  - For a key of length *n bits*, brute force requires (2^n) attempts in the worst case.
  - Example: A 56-bit DES key → (2^{56}) ≈ 72 quadrillion possibilities.

📎 Google Search

---

4. Brute Force Attack & Types of Cryptanalysis Attacks

- **Brute Force:** Exhaustively trying all keys until the message is decrypted.
- **Types of Cryptanalysis Attacks:**
  - **Ciphertext-only attack:** Only ciphertext is available.
  - **Known-plaintext attack:** Attacker has both plaintext and its ciphertext.
  - **Chosen-plaintext attack:** Attacker can encrypt chosen plaintexts.
  - **Chosen-ciphertext attack:** Attacker can decrypt chosen ciphertexts.
- Example: DES can be brute-forced in hours using modern computing power.

📎 Google Search

---

5. Approaches to Attack a Conventional Encryption Scheme

- **Cryptanalysis:** Using mathematical weaknesses.
- **Brute force:** Trying every possible key.
- **Hybrid attacks:** Combination of cryptanalysis and brute force.
- **Social engineering:** Attacking human factors (e.g., tricking someone to reveal the key).

📎 Google Search

---

6. Principles of Public Key Cryptography (Note: appears here though related to Topic 6)

- Relies on **two keys:** Public (encryption) and Private (decryption).
- Based on **hard mathematical problems** (e.g., factoring large numbers, elliptic curves).
- Ensures:
  - Confidentiality
  - Authentication
  - Non-repudiation
  - Digital signatures
- Example: RSA, ECC.

📎 Google Search

---

7. Public vs. Private Key Cryptography Comparison

- **Private (Symmetric):** Same key, fast, efficient for bulk data. Weakness: key distribution.
- **Public (Asymmetric):** Different keys, slower, secure key exchange, supports digital signatures.
- **Real-world usage:** Hybrid systems (e.g., SSL/TLS uses both).

🔗 Google Search

---

8. Monoalphabetic vs. Polyalphabetic Cipher

- **Monoalphabetic Cipher:** Each letter of plaintext is always mapped to the same ciphertext letter. Example: Caesar cipher. Weak to frequency analysis.
- **Polyalphabetic Cipher:** Uses multiple substitution alphabets. Example: Vigenère cipher. More secure because it masks frequency.

🔗 Google Search

---

9. Unconditionally Secure Cipher vs. Computationally Secure Cipher

- **Unconditionally Secure Cipher:** Cannot be broken even with infinite computing power. Example: One-Time Pad (OTP).
- **Computationally Secure Cipher:** Cannot be broken with current technology within a reasonable time (e.g., AES).

🔗 Google Search

---

10. Plaintext → Ciphertext Conversion (Symmetric Example)

- Use of substitution/transposition methods.
- Example: Caesar cipher shifts plaintext letters.
- More advanced: AES converts blocks of data (128 bits) into ciphertext.

🔗 Google Search

---

11. Transposition Cipher

Rearranges the order of characters without changing them.

- Example: **HELLO** → using a columnar transposition → “HLOEL”.
- Different from substitution ciphers, which replace letters.

🔗 Google Search

---

12. Playfair Cipher (Rules)

- Uses a 5x5 matrix of alphabets (I & J treated as one).

- Encryption rules:

    1. Same row → replace with letter to the right.

2. Same column → replace with letter below.
3. Different row/column → form rectangle, replace with opposite corners.

- Example: Plaintext "HELLO" → Ciphertext depends on chosen key.

📎 Google Search

---

13. Application of Public Key Cryptosystem

- Secure email (PGP)
- Digital signatures
- Key exchange in SSL/TLS
- Secure payment (SET protocol)
- Authentication in e-commerce

📎 Google Search

---

14. AAA (Authentication, Authorization, Accounting)

- **Authentication:** Verifying user identity (password, biometrics).
- **Authorization:** Granting access rights to resources.
- **Accounting:** Tracking user activities for audit.
- Widely used in network access control (e.g., RADIUS).

📎 Google Search

---

15. Playfair Cipher Example

Encrypt "The key is hidden under the door" using keyword "guidance".

- Key builds 5x5 matrix.
- Apply Playfair rules.
- Each digraph (pair of letters) is encrypted separately.
- **Exam Note:** Always prepare at least one solved example.

📎 Google Search

---

16. Middle-Man Attack (Man-in-the-Middle, MITM)

An attacker secretly intercepts and possibly alters communication between two parties.

- **Example:** Attacker sits between Alice and Bob during key exchange and relays messages.
- **Result:** Both believe they are communicating securely, but attacker controls the data.
- **Prevention:** Use of strong authentication (digital certificates, PKI).

📎 Google Search

📓 Topic 3: Modern Symmetric-Key Ciphers

---

1. What is Block Cipher?

A **block cipher** is an encryption method that processes data in **fixed-size blocks** (e.g., 64-bit or 128-bit). Each block of plaintext is transformed into a block of ciphertext using the same secret key.

- **Examples:** DES (64-bit blocks), AES (128-bit blocks).
- **Strengths:** Strong security, diffusion of plaintext over ciphertext.
- **Weakness:** If the same block repeats, ciphertext may also repeat (unless a mode of operation is used).
- **Usage:** File encryption, SSL/TLS, VPNs.

📎 Google Search

---

2. Why are Block Cipher Modes Convenient?

Block ciphers must be used in **modes of operation** to handle large messages securely.

- **Modes:**
  - **ECB (Electronic Code Book):** Each block encrypted independently (weak due to repetition patterns).
  - **CBC (Cipher Block Chaining):** Each plaintext block XORed with previous ciphertext → more secure.
  - **CFB (Cipher Feedback):** Converts block cipher into stream cipher.
  - **OFB (Output Feedback):** Similar to CFB but independent of plaintext.
  - **CTR (Counter Mode):** Uses counters for encryption, allows parallel processing.
- **Convenience:** Handle arbitrary message sizes, increase security, prevent pattern leakage.

📎 Google Search

---

3. Block Cipher vs. Stream Cipher

- **Block Cipher:** Encrypts data in fixed blocks. Slower but strong. Example: AES, DES.
- **Stream Cipher:** Encrypts one bit/byte at a time using a keystream. Faster, useful for real-time data. Example: RC4.
- **Difference:**
  - Block: Deterministic on same block input (unless mode used).
  - Stream: Produces variable keystream, good for continuous data streams (video/audio).
- **Security:** Block ciphers generally stronger if used with proper modes.

📎 Google Search

---

4. Why Some Block Cipher Modes Use Only Encryption While Others Use Both Encryption & Decryption?

- **Encryption-Only Modes:** CTR, OFB generate keystream independent of plaintext. Decryption is just reapplying encryption keystream.
- **Encryption & Decryption Modes:** CBC, CFB require different handling for decryption since ciphertext chaining must be reversed.
- **Reason:** Modes that generate independent keystream (CTR, OFB) can reuse encryption for both processes.

🔗 Google Search

---

5. Explain the One-Time Pad with Example

The **One-Time Pad (OTP)** is the only known **unbreakable encryption method**.

- **Process:**
    - Generate a random key as long as the plaintext.
    - XOR plaintext with key → ciphertext.
    - Decryption is reversing with same key.
- **Properties:**
    - **Unconditional Security:** Cannot be broken even with infinite computing power.
    - **Requirements:** Key must be truly random, as long as the message, never reused.
- **Example:** Plaintext: `HELLO` → Binary → XOR with random key → Ciphertext.

🔗 Google Search

---

6. Confusion and Diffusion

- **Confusion:** Hides the relationship between plaintext, ciphertext, and key. Achieved through complex substitutions.
- **Diffusion:** Spreads the influence of one plaintext bit over many ciphertext bits. Achieved through permutations and mixing.
- **Purpose:** Together, they strengthen ciphers by making cryptanalysis difficult.
- **Example:** DES uses substitution boxes (confusion) and permutations (diffusion).

🔗 Google Search

---

7. Feistel Cipher

A **Feistel cipher** is a structure used in many block ciphers (e.g., DES).

- **Operation:**
    - Divide plaintext block into two halves (L, R).

- Process rounds: (L_{i+1} = R_i), (R_{i+1} = L_i \oplus F(R_i, K_i)).
- Swap halves and repeat for multiple rounds.

- **Advantages:**
  - Same structure used for both encryption & decryption.
  - Strong if round function F is secure.

- **Examples:** DES, Blowfish.

🔗 Google Search

---

8. Steganography & Feistel Cipher Parameters

- **Steganography:** Hiding messages inside other digital media (images, audio, video) so presence of the message is hidden.
  - Example: Embedding a secret text into an image's least significant bits.

- **Feistel Parameters:**
  - Block size
  - Key size
  - Number of rounds
  - Round function F design choices These choices determine the **strength and efficiency** of the cipher.

🔗 Google Search

---

9. Cipher Feedback (CFB) Model of Operation

CFB turns a block cipher into a **self-synchronizing stream cipher**.

- **Process:**
  - Encrypt IV (Initialization Vector) → take part of output → XOR with plaintext → ciphertext.
  - Ciphertext also feeds back into encryption for next block.

- **Advantage:** Can process smaller units (bits/bytes) instead of large blocks.

- **Use Case:** Real-time data encryption (secure communication channels).

🔗 Google Search

📘 Topic 4: Data Encryption Standard (DES)

---

1. What is DES? What are its Features?

DES (Data Encryption Standard) is a **symmetric-key block cipher** developed by IBM in the 1970s and adopted by NIST in 1977 as a federal standard.

- **Block Size:** 64 bits.
- **Key Size:** 56-bit effective key (though originally 64-bit with 8 parity bits).
- **Rounds:** 16 Feistel rounds.
- **Structure:** Based on the Feistel network (substitution + permutation).
- **Features:**
  - Symmetric (same key for encryption/decryption).
  - Uses S-boxes (confusion) and permutations (diffusion).
  - Widely used in early banking and secure communications.
- **Limitation:** 56-bit key is too short today → vulnerable to brute-force attacks.

🔗 Google Search

---

2. General DES Encryption Process with Diagram (Merits & Demerits)

Process:

1. **Initial Permutation (IP):** Rearranges bits of plaintext.
2. **Divide into two halves (L, R).**
3. **16 Rounds of Feistel operations:**
   - Expansion (32 → 48 bits),
   - XOR with subkey,
   - Substitution using S-boxes,
   - Permutation.
4. **Swap halves each round.**
5. **Final Permutation (FP):** Produces ciphertext.

Merits:

- Strong diffusion and confusion.
- Resistant to many cryptanalysis methods when designed.

Demerits:

- Small key size (56 bits) → brute-forceable.
- Slower compared to modern algorithms (AES).
- Obsolete for modern security needs.

🔗 Google Search

---

3. Strength of DES Algorithm

- **Strengths:**
  - Resistant to differential & linear cryptanalysis (with full 16 rounds).
  - Well-tested for decades.
- **Weaknesses:**
  - **Key size (56-bit):** Too short, brute force possible in hours with modern hardware.
  - Susceptible to side-channel attacks (timing, power analysis).
- **Historical Note:** In 1998, the EFF machine broke DES in less than 3 days.

🔗 Google Search

---

4. Single Round DES Architecture

A single DES round consists of:

1. **Input:** Left half (L), Right half (R).
2. **Expansion (E):** Expands 32-bit R into 48 bits.
3. **Key Mixing:** XOR with 48-bit round subkey.
4. **Substitution (S-boxes):** 48 bits → 32 bits.
5. **Permutation (P-box):** Rearranges bits.
6. **Output:** ($L_{i+1} = R_i$), ($R_{i+1} = L_i \oplus f(R_i, K_i)$).

Note: 16 such rounds strengthen the cipher.

🔗 Google Search

---

5. What is Triple DES (3DES)? How Many Keys Are Used?

Triple DES (3DES): An enhancement to DES to improve security by applying DES **three times**.

- **Operation:** Encrypt → Decrypt → Encrypt (EDE).
- **Keys Used:**
  - **Two-key 3DES:** Uses 2 keys (K1, K2).
  - **Three-key 3DES:** Uses 3 independent keys (K1, K2, K3).
- **Effective Key Size:**
  - Two-key: 112 bits.
  - Three-key: 168 bits.
- **Use Cases:** Banking industry (ATMs, SWIFT).

🔗 Google Search

---

6. Why is the Middle Portion of 3DES Decryption Instead of Encryption?

- In **3DES EDE mode**, the middle step is **decryption** for compatibility with single DES systems.
- **Reason:**
  - If all 3 keys are the same (K1 = K2 = K3), 3DES reduces to single DES.
  - Ensures backward compatibility with legacy DES systems.
- **Operation:**
  - Ciphertext = (E_{K1}(D_{K2}(E_{K3}(Plaintext)))).
- **Benefit:** Stronger security while maintaining DES compatibility.

🔗 Google Search

📘 Topic 5: Advanced Encryption Standard (AES)

---

1. Final Set of Criteria Used by NIST to Evaluate Candidate AES Cipher

In 1997, NIST initiated the AES competition to replace DES. The evaluation process considered multiple criteria:

- **Security:** Resistance against known attacks (brute force, differential, linear cryptanalysis).
- **Cost:** Efficient implementation in hardware and software.
- **Algorithm & Key Size Flexibility:** Support for 128, 192, and 256-bit keys.
- **Performance:** High speed in encryption/decryption across different platforms (smart cards to servers).
- **Simplicity:** Easy to understand and implement without hidden weaknesses.
- **International Acceptance:** Not controlled by patents. **Winner:** Rijndael algorithm → became AES in 2001.

🔗 Google Search

---

2. Salient Features of AES

- **Block Size:** Fixed 128 bits.
- **Key Sizes:** 128, 192, or 256 bits.
- **Structure:** Substitution–Permutation network, not Feistel.
- **Rounds:**
  - 10 rounds (128-bit key).
  - 12 rounds (192-bit key).
  - 14 rounds (256-bit key).
- **Operations per Round:**

  - SubBytes (byte substitution using S-box).
  - ShiftRows (row-wise permutation).
  - MixColumns (column mixing for diffusion).
  - AddRoundKey (XOR with round key).

- **Security:** Resistant against all practical attacks.

- **Speed:** Very fast in software and hardware, widely adopted globally.

🔗 Google Search

---

3. Differences Between Rijndael and AES

- **Rijndael:** Original algorithm submitted by Belgian cryptographers Joan Daemen and Vincent Rijmen.

  - Variable block sizes: 128, 192, 256 bits.
  - Variable key sizes: 128, 192, 256 bits.

- **AES (Final Standard):**

  - Restricted to **128-bit block size** only.
  - Key sizes: 128, 192, 256 bits.

- **Summary:** AES is a restricted version of Rijndael chosen by NIST.

🔗 Google Search

---

4. Difference Between AES Decryption Algorithm and Equivalent Inverse Cipher

- **AES Decryption:** Applies reverse order of operations (InvShiftRows → InvSubBytes → AddRoundKey → InvMixColumns).
- **Equivalent Inverse Cipher:** A modified form where decryption is performed using the same structure as encryption but with pre-computed round keys.
- **Purpose:** Simplifies hardware implementation by allowing the same design for both encryption and decryption.

🔗 Google Search

---

5. Differentiate Between AES and DES Algorithm

- **Algorithm Type:**

  - DES: Feistel network (16 rounds).
  - AES: Substitution–Permutation network.

- **Block Size:**

  - DES: 64-bit.
  - AES: 128-bit.

- **Key Size:**
  - DES: 56-bit effective.
  - AES: 128, 192, or 256-bit.
- **Security:**
  - DES: Weak due to small key size.
  - AES: Strong, no practical attacks.
- **Speed:**
  - AES is faster and more efficient in both hardware and software.
- **Adoption:**
  - DES is obsolete.
  - AES is the current global standard (used in SSL/TLS, VPNs, Wi-Fi, etc.).

🔗 Google Search

---

6. Detailed Structure of AES Encryption and Decryption

AES Encryption Steps:

1. **Initial Round:** AddRoundKey (XOR plaintext with first round key).
2. **Main Rounds (9, 11, or 13 depending on key size):**
   - SubBytes (non-linear substitution using S-box).
   - ShiftRows (shifts rows left for diffusion).
   - MixColumns (mixes data across columns).
   - AddRoundKey (XOR with subkey).
3. **Final Round:** SubBytes → ShiftRows → AddRoundKey (no MixColumns).

AES Decryption Steps: Reverse process: InvShiftRows, InvSubBytes, InvMixColumns, AddRoundKey.

Strength: Strong resistance against all known attacks, fast and scalable.

🔗 Google Search

📘 Topic 6: Asymmetric-Key Cryptography

---

1. What do you mean by Asymmetric Key?

- **Definition:** Asymmetric key cryptography (public-key cryptography) uses a pair of keys:
  - **Public Key:** Shared openly, used for encryption.
  - **Private Key:** Kept secret, used for decryption.

- **Idea:** What one key encrypts, only the other key can decrypt.
- **Advantages:** Solves the **key distribution problem** in symmetric cryptography.
- **Examples:** RSA, Elliptic Curve Cryptography (ECC), Diffie–Hellman.

🔗 Google Search

---

2. Differences Between Symmetric and Asymmetric Cipher Model

- **Symmetric Cipher Model:**
  - One shared secret key.
  - Faster, efficient for bulk data.
  - Problem: Key distribution is hard.
- **Asymmetric Cipher Model:**
  - Two keys (public + private).
  - Slower but solves key distribution issue.
  - Enables digital signatures and authentication.

🔗 Google Search

---

3. Public Key vs. Conventional Encryption

- **Public Key (Asymmetric):**
  - Different keys for encryption/decryption.
  - Enables digital signatures.
  - Example: RSA.
- **Conventional Encryption (Symmetric):**
  - Same key used for both.
  - Faster but insecure key sharing.
- **Usage:** Often combined (Hybrid Cryptosystem):
  - Public key → secure key exchange.
  - Symmetric key → actual data encryption.

🔗 Google Search

---

4. Symmetric vs. Asymmetric Encryption Techniques

- **Symmetric:**
  - One secret key.
  - Very fast, less computation.

    - Examples: DES, AES.

- **Asymmetric:**
    - Two keys (public + private).
    - Slower due to heavy math (large prime factorization, elliptic curves).
    - Examples: RSA, ECC.

- **Practical Use:** Symmetric used for bulk, Asymmetric used for secure key exchange.

🔗 Google Search

---

5. Link Encryption vs. End-to-End Encryption

- **Link Encryption:**
    - Data encrypted at each hop (router, switch).
    - Each device decrypts & re-encrypts before forwarding.
    - Protects data *while in transit* but intermediate devices see plaintext.

- **End-to-End Encryption:**
    - Data encrypted at sender and decrypted only at receiver.
    - Protects against eavesdropping even at intermediate devices.

- **Example:**
    - Link encryption: WAN between routers.
    - End-to-end encryption: WhatsApp, TLS.

🔗 Google Search

---

6. Advantages and Disadvantages of Symmetric & Asymmetric Encryption

- **Symmetric:**
    - ✔ Faster, requires less computational power.
    - ✖ Key distribution problem, not scalable.

- **Asymmetric:**
    - ✔ Solves key distribution, enables digital signatures.
    - ✖ Slower, resource-intensive.

- **Combination:** Real-world systems use both for balance (e.g., HTTPS).

🔗 Google Search

---

7. RSA Algorithm (Encryption & Decryption)

- **Steps:**

1. Choose two large primes p, q.
2. Compute $n = p \times q$.
3. Compute Euler's totient: $\phi(n) = (p-1)(q-1)$.
4. Choose public exponent e (coprime with $\phi(n)$).
5. Compute private key d such that $e \times d \equiv 1 \ (\text{mod } \phi(n))$.
6. Public Key = (e, n), Private Key = (d, n).

- **Encryption:** $C = M^e \mod n$.
- **Decryption:** $M = C^d \mod n$.
- **Security Basis:** Factoring large n into primes is computationally hard.

📎 Google Search

---

8. Elliptic Curve Cryptography (ECC)

- **Definition:** Public key cryptography based on algebraic structures of elliptic curves over finite fields.
- **Advantage:** Provides same security as RSA but with much smaller key sizes.
  - Example: 256-bit ECC ≈ 3072-bit RSA in strength.
- **Used in:** Mobile devices, IoT, SSL/TLS, cryptocurrencies.
- **Example Curve:** $y^2 = x^3 + ax + b$.

📎 Google Search

---

9. Security of RSA & Possible Attacks

- **Attacks:**
  - **Mathematical Attack:** Factoring modulus n.
  - **Brute Force:** Trying keys (impractical with large n).
  - **Timing Attack:** Observing computation time to reveal private key.
  - **Chosen Ciphertext Attack:** Manipulating ciphertexts.
- **RSA Defense:**
  - Use large key sizes (≥2048 bits).
  - Implement secure padding (OAEP).
  - Use constant-time algorithms to prevent side-channel leaks.

📎 Google Search

---

10. Perform RSA with Given Values (p=7, q=11, e=13, M=8)

- **Step 1:** $n = p \times q = 7 \times 11 = 77$.
- **Step 2:** $\varphi(n) = (p-1)(q-1) = 6 \times 10 = 60$.

- **Step 3:** e = 13 (given). Check gcd(13, 60) = 1 → valid.
- **Step 4:** Find d such that (e × d) mod 60 = 1 → d = 37.
- **Step 5:** Public Key = (13, 77), Private Key = (37, 77).
- **Encryption:** C = M^e mod n = 8^13 mod 77 = 57.
- **Decryption:** M = C^d mod n = 57^37 mod 77 = 8. ✔ Correctly returns original message.

🔗 Google Search

---

11. Public Key Cryptography for Encryption & Authentication

- **Encryption:** Sender encrypts message with recipient's **public key** → only recipient's **private key** can decrypt.
- **Authentication/Digital Signature:** Sender signs message by encrypting hash with their **private key** → anyone can verify with sender's **public key**.
- **Benefit:** Provides both confidentiality and authentication simultaneously.

🔗 Google Search

---

12. Requirements for a Secure Public Key Cryptosystem

- **Correctness:** Encryption + decryption must recover plaintext.
- **Security:** Based on hard mathematical problems (factoring, discrete logs).
- **One-way Trapdoor Function:** Easy to compute one way, hard to reverse without key.
- **Practical Efficiency:** Not too slow for real-world use.
- **Scalability:** Should support multiple users securely.

🔗 Google Search

---

13. Public Key Algorithms (General Overview)

- **RSA:** Based on integer factorization.
- **Diffie–Hellman (DH):** Secure key exchange.
- **ECC:** Uses elliptic curves (more efficient).
- **ElGamal:** Based on discrete logarithm problem.
- **Applications:** Secure email (PGP), digital signatures, SSL/TLS, cryptocurrency wallets.

🔗 Google Search

📘 Topic 7: Message Integrity and Message Authentication

---

1. What is Message Authentication (MAC)?

- **Message Authentication:** The process of verifying that a message has **not been altered** and comes from an **authentic source**.
- **MAC (Message Authentication Code):**
  - A cryptographic checksum generated using a **secret key + message**.
  - Sent along with the message.
  - Receiver recomputes MAC with the same key → if equal, message is authentic.
- **Goal:** Protect **integrity** and **authenticity**, not confidentiality.
- **Example:** HMAC-SHA256 used in APIs and SSL/TLS.

🔗 Google Search

---

2. How is Message Authentication Performed?

- **Methods:**
  1. **Using MACs:** Secret key + message → cryptographic function → MAC.
  2. **Using Cryptographic Hash Functions:** Message → Hash → Sent with message.
  3. **Using Digital Signatures:** Sender signs hash of message using private key → Receiver verifies with public key.
- **Example:** In SSL/TLS, HMAC is used to authenticate packets.

🔗 Google Search

---

3. Message Authentication & Classes of Authentication Functions

- **Message Authentication:** Confirms that data is from an authentic sender and unaltered.
- **Classes of Functions:**
  - **Hash Functions (without key):** Provide integrity, but no authentication.
  - **Message Authentication Codes (MACs):** Hash + secret key → ensures authenticity + integrity.
  - **Digital Signatures:** Use public/private keys → authenticity + non-repudiation.

🔗 Google Search

---

4. Requirements for Secure Use of Conventional Encryption

- Secret key must remain private between sender and receiver.
- Strong encryption algorithm must be used.
- Random IVs (Initialization Vectors) for security.
- Keys must be changed frequently to prevent replay or brute force.
- Both parties must authenticate each other to avoid MITM attacks.

🔗 Google Search

---

5. Requirements for Message Authentication

- **Integrity:** Message must not be altered.
- **Authentication:** Must prove message comes from correct source.
- **Freshness:** Prevent replay attacks (timestamps, sequence numbers).
- **Non-repudiation (optional):** Sender cannot deny sending.
- **Efficiency:** Should be fast and lightweight.

📎 Google Search

---

6. MD5 vs SHA Algorithm (Comparison)

- **MD5 (Message Digest 5):**
  - Output: 128-bit hash.
  - Fast but insecure (collisions found).
  - Used historically in checksums.
- **SHA (Secure Hash Algorithm):**
  - Variants: SHA-1 (160-bit), SHA-2 (256, 512 bits), SHA-3.
  - Stronger security than MD5.
- **Comparison:**
  - MD5 is faster but weak.
  - SHA-256/SHA-512 is secure and widely used in TLS, Bitcoin, PKI.

📎 Google Search

---

7. MD5 Algorithm in Detail & Comparison with SHA-1

- **MD5 Process:**
  - Input broken into 512-bit blocks.
  - Each block processed through 4 rounds of non-linear functions.
  - Produces a **128-bit digest**.
- **SHA-1 Process:**
  - Similar block processing, but stronger design.
  - Produces **160-bit digest**.
- **Performance:** MD5 is faster, but SHA-1 is more secure.
- **Current Status:** Both are deprecated for modern cryptographic use due to collision vulnerabilities.

📎 Google Search

---

8. Approaches to Producing Message Authentication

- **Message Authentication Code (MAC):** Key + message → secure tag.
- **Hash-based Authentication (HMAC):** Hash function + key → strong MAC.
- **Digital Signatures:** Public/private key pair ensures authenticity + non-repudiation.
- **Encrypt + MAC:** Encrypt message + append authentication tag.

📎 Google Search

---

9. Hash Function & SHA-512 Logic Algorithm

- **Hash Function:** A one-way function mapping variable input → fixed-size output (digest).
  - Requirements: Pre-image resistance, collision resistance, avalanche effect.
- **SHA-512:**
  - Input processed in 1024-bit blocks.
  - Produces 512-bit hash.
  - Uses 80 rounds of bitwise operations, modular additions.
  - Used in SSL/TLS, digital signatures, blockchain.

📎 Google Search

---

10. How Does a Secure Algorithm Work?

- A secure algorithm ensures:
  - **Confusion:** Complex substitution hides plaintext-key relation.
  - **Diffusion:** One bit change in plaintext affects many bits in ciphertext.
  - **Large Key Space:** Resistant to brute-force.
  - **Mathematical Hardness:** Based on difficult problems (factoring, discrete logs).
- **Example:** AES is secure due to strong substitution-permutation network and large key size.

📎 Google Search

---

11. Digital Signature Process for Message Authentication

- **Process:**
  1. Sender computes hash of message.
  2. Sender encrypts hash with their private key → digital signature.
  3. Receiver decrypts signature using sender's public key.
  4. Receiver compares computed hash vs. received hash.
- **Purpose:** Provides **authentication, integrity, non-repudiation**.
- **Use Case:** Email signing, software distribution, e-commerce.

📎 Google Search

📓 Topic 8: Cryptographic Hash Functions

---

1. What is Hash Function? Mention the Requirements for Hash Function

- **Definition:** A hash function is a one-way mathematical function that takes an input of arbitrary length and produces a fixed-size output (hash/digest).
- **Purpose:** Used for integrity verification, digital signatures, password storage, and authentication.
- **Requirements:**
  - **Pre-image Resistance:** Hard to find input from its hash.
  - **Second Pre-image Resistance:** Hard to find a different input with the same hash.
  - **Collision Resistance:** Hard to find two different inputs with the same hash.
  - **Efficiency:** Should be fast and easy to compute.
  - **Avalanche Effect:** A small change in input → large unpredictable change in output.

🔗 Google Search

---

2. Weak Collision Resistance vs. Strong Collision Resistance

- **Weak Collision Resistance (Second Pre-image Resistance):** Given a message M1, it should be computationally infeasible to find a different message M2 such that Hash(M1) = Hash(M2).
- **Strong Collision Resistance:** It should be infeasible to find **any two different messages** M1 and M2 that produce the same hash.
- **Example:**
  - Weak: Hard to find another password with same hash as yours.
  - Strong: Hard to find any two files with same hash.
- **Note:** Strong collision resistance is stricter and harder to achieve.

🔗 Google Search

---

3. Differentiate Between Weak and Strong Collision Resistance

- **Weak Collision Resistance:** Focuses on a specific input → find another input with same hash.
- **Strong Collision Resistance:** Focuses on any two arbitrary inputs producing same hash.
- **Comparison:**
  - Weak is easier to break.
  - Strong is more secure and required for digital signatures, certificates.

🔗 Google Search

---

4. Block Cipher Modes of Operation (Brief Overview)

Although mainly studied under symmetric encryption, hash functions can also be built using block cipher modes.

- **ECB (Electronic Code Book):** Each block encrypted independently. Weak due to repetition.
- **CBC (Cipher Block Chaining):** Each block XORed with previous ciphertext block. Stronger diffusion.
- **CFB (Cipher Feedback):** Converts block cipher into stream cipher.
- **OFB (Output Feedback):** Produces keystream independent of plaintext.
- **CTR (Counter Mode):** Uses counters for parallel encryption.
- **Relevance to Hash:** Some hash constructions are based on block ciphers (e.g., Davies-Meyer).

🔗 Google Search

---

5. Differentiate MAC and Hash Function. Role of Compression Function in Hash Function

- **MAC (Message Authentication Code):**
  - Hash + Secret Key.
  - Provides both integrity & authentication.
  - Example: HMAC (Hash-based MAC).
- **Hash Function:**
  - No key required.
  - Provides only integrity.
- **Role of Compression Function in Hash Function:**
  - Hash functions like MD5, SHA use compression functions to process fixed-size blocks (e.g., 512 bits) and compress them into smaller digests.
  - Ensures avalanche effect and collision resistance.

🔗 Google Search

---

6. General Structure of Secure Hash Function

- **Input:** Message broken into fixed-size blocks (e.g., 512 bits).
- **Processing:** Each block processed through a compression function with chaining values.
- **Output:** Final fixed-size hash (e.g., SHA-256 = 256 bits).
- **Structure Example:**
  - **MD5:** 128-bit digest, 64 rounds.
  - **SHA-256:** 256-bit digest, 64 rounds.
  - **SHA-512:** 512-bit digest, 80 rounds.
- **Security:** Depends on avalanche effect, collision resistance, and round functions.

🔗 Google Search

📓 Topic 9: Key Management

---

1. General Characteristics / Categories of Public Key Distribution Schemes

Key Distribution = process of securely delivering cryptographic keys between parties.

- **General Characteristics:**
  - Security: Should resist interception, modification.
  - Efficiency: Must support large networks.
  - Scalability: Work for millions of users.
  - Authentication: Ensure keys come from trusted sources.
- **Four Categories:**
  1. **Public Announcement:** User publicly shares key (insecure, prone to spoofing).
  2. **Publicly Available Directory:** Keys stored in trusted directories (vulnerable if directory is compromised).
  3. **Public Key Authority:** Central authority validates keys.
  4. **Public Key Certificates (PKI):** Keys bound to digital certificates signed by trusted Certificate Authorities (CA).

🔗 Google Search

---

2. Kerberos (V4) and Authentication Service

- **Kerberos:** A network authentication protocol based on **symmetric key cryptography**.
- **Developed:** At MIT for secure login/authentication.
- **Kerberos V4:**
  - Uses a **trusted Key Distribution Center (KDC)**.
  - Issues "tickets" for authentication.
  - Users authenticate once and then use tickets for further access (SSO – Single Sign-On).
- **Authentication Service (AS):** Validates user identity → issues Ticket Granting Ticket (TGT).
- **Benefits:** Prevents password transmission over the network, protects against replay attacks.

🔗 Google Search

---

3. Concept of Keyed Cryptography & Types of Keys

- **Keyed Cryptography:** Cryptography that requires the use of a secret key.

- **Types of Keys:**
  - **Symmetric Keys:** Same key for encryption & decryption (DES, AES).
  - **Asymmetric Keys:** Public key for encryption, private key for decryption (RSA, ECC).
  - **Session Keys:** Temporary symmetric keys for one communication session.
  - **Master Keys:** Long-term keys used to derive session keys.

📎 Google Search

---

4. Key Distribution Center (KDC)

- **Definition:** A trusted server that distributes secret keys to users.
- **Functions:**
  - Provides **session keys** securely.
  - Reduces the number of keys each user must store.
- **Process:**
  - User → KDC request.
  - KDC issues session key encrypted with user's long-term key.
- **Used in:** Kerberos, enterprise networks.

📎 Google Search

---

5. Public Key Infrastructure (PKI) & Its Necessity

- **PKI:** A framework that manages digital certificates and public keys.
- **Functions:**
  - Certificate Authority (CA) issues digital certificates.
  - Provides authentication, integrity, confidentiality, and non-repudiation.
- **Without PKI:** It is very difficult to trust public keys, leading to impersonation attacks.
- **Applications:** SSL/TLS, VPNs, digital signatures, secure email.

📎 Google Search

---

6. Session Key vs. Master Key

- **Session Key:**
  - Temporary key used for one communication session.
  - Provides confidentiality for that session only.
  - Example: In HTTPS, session keys are negotiated via TLS handshake.
- **Master Key:**

  - Long-term key used to generate session keys.
  - Must be kept highly secure.

- **Comparison:** Session key is short-lived (less risk if compromised), master key is long-lived.

📎 Google Search

---

7. Man-in-the-Middle (MITM) Attack

- **Definition:** An attacker secretly intercepts and possibly alters communication between two parties.
- **Process:** Attacker sits between sender & receiver → relays messages → both think they're communicating directly.
- **Example:** During Diffie–Hellman key exchange, attacker establishes separate keys with each party.
- **Defense:** Use authentication (digital certificates, signed keys).

📎 Google Search

---

8. Certificate Revocation

- Certificates may need to be **revoked** before expiry due to:

  - Compromised private key.
  - User left organization.
  - CA misissued certificate.

- **Handled by:**

  - **CRL (Certificate Revocation List):** List of revoked certificates.
  - **OCSP (Online Certificate Status Protocol):** Real-time certificate validation.

📎 Google Search

---

9. X.509 Certificate Format & Revocation List (CRL)

- **X.509:** Standard for digital certificates used in PKI.

- **Typical Contents:**

  - Version
  - Serial Number
  - Signature Algorithm
  - Issuer (CA)
  - Validity Period
  - Subject (user, server info)
  - Public Key

- **Certificate Revocation List (CRL):** Contains serial numbers of revoked certificates.

- **Delta CRL:** Only lists certificates revoked since the last full CRL update.

🔗 Google Search

---

10. Purpose of X.509 Standard

- Provides a **standardized format** for public key certificates.
- Used in SSL/TLS, secure email, VPNs.
- Ensures interoperability between different systems.
- Defines certificate chains and trust hierarchy (Root CA → Intermediate CA → End-user cert).

🔗 Google Search

---

11. Applications of IP Security (IPSec)

- **IPSec:** A protocol suite for securing IP communication by authenticating and encrypting each packet.
- **Applications:**
    - VPNs (Virtual Private Networks).
    - Secure branch-office communication.
    - Secure email.
    - Protecting routing protocols.
- **Benefits:** Provides confidentiality, integrity, authentication at the **network layer**.

🔗 Google Search

---

12. Requirements of Kerberos

Four main requirements defined:

1. **Secure Authentication:** No plaintext passwords transmitted.
2. **Reliability:** Must be available continuously.
3. **Transparency:** Users authenticate once (SSO).
4. **Scalability:** Must work across large distributed systems.

🔗 Google Search

📘 Topic 10: Digital Signature

---

1. What is Digital Signature?

- **Definition:** A digital signature is the electronic equivalent of a handwritten signature or a stamped seal.
- **How it Works:**
    - Sender generates a **hash** of the message.
    - Hash is **encrypted with sender's private key** → forms the digital signature.

- Receiver decrypts signature using sender's **public key** → verifies hash.

- **Purpose:** Provides **integrity, authenticity, and non-repudiation**.

- **Example:** Used in software updates, e-commerce, legal documents.

📎 Google Search

---

2. Requirements for a Digital Signature

A valid digital signature scheme must provide:

1. **Authenticity:** Only the signer could have generated it.
2. **Integrity:** Message must not be altered after signing.
3. **Non-repudiation:** Signer cannot deny signing.
4. **Efficiency:** Must be easy to generate and verify.
5. **Security:** Must be computationally infeasible to forge.

📎 Google Search

---

3. Properties a Digital Signature Should Have

- **Unforgeability:** Cannot be forged without the private key.
- **Authenticity:** Confirms the identity of sender.
- **Integrity:** Protects against tampering.
- **Non-repudiation:** Prevents denial of authorship.
- **Verifiability:** Anyone with public key can verify.
- **Portability:** Must be transferable and verifiable by third parties.

📎 Google Search

---

4. DSA Algorithm (Digital Signature Algorithm)

- **DSA (1991, NIST Standard):**

  - Based on **modular exponentiation & discrete logarithms**.

- **Steps:**

  1. Generate parameters (p, q, g).
  2. Generate private key (x).
  3. Compute public key (y = g^x mod p).
  4. For signing: Pick random k, compute signature pair (r, s).
  5. For verification: Receiver checks validity using signer's public key.

- **Output:** Digital signature = (r, s).

- **Use Case:** U.S. federal standards for secure digital signatures.

📎 Google Search

---

5. Digital Signature Standard (DSS)

- **DSS = NIST Standard (1994):** Specifies algorithms for digital signatures.
- **Includes:**
  - DSA (Digital Signature Algorithm).
  - RSA-based signatures.
  - ECDSA (Elliptic Curve DSA).
- **Purpose:** Define secure and standardized digital signature methods for federal systems.

🔗 Google Search

---

6. Direct vs. Arbitrated Digital Signature

- **Direct Digital Signature:**
  - Sender signs message directly with private key.
  - Receiver verifies with sender's public key.
  - Simple but risks disputes if sender denies later.
- **Arbitrated Digital Signature:**
  - A trusted third party (arbiter) validates and stores signatures.
  - Provides stronger non-repudiation.
- **Example:** Digital notary services use arbitrated signatures.

🔗 Google Search

---

7. Requirements a Digital Signature Scheme Should Satisfy

- **Correctness:** Signature must verify correctly with public key.
- **Unforgeability:** Cannot forge without private key.
- **Non-repudiation:** Signer cannot deny.
- **Efficiency:** Must be practical in speed and memory.
- **Security against Attacks:** Must resist chosen-plaintext and chosen-message attacks.

🔗 Google Search

---

8. RSA Digital Signature

- **How it Works:**
  - Sender computes hash of message.
  - Sender encrypts hash with **private key** → digital signature.
  - Receiver decrypts signature with **public key**, compares with hash.

- **Security:** Based on difficulty of factoring large integers.
- **Applications:** Used in SSL/TLS, email signing, blockchain.

🔗 Google Search

---

9. Digital Signature Procedure (with Diagram)

Steps:

1. **Sender:** Hash(M) → Encrypt hash with private key = Signature.
2. **Send:** Message + Signature → Receiver.
3. **Receiver:** Hash(M) again → Decrypt signature with sender's public key.
4. **Compare:** If both match → message is authentic & unmodified. **Diagram (conceptual):**

```
Message → Hash → Encrypt with Private Key → Signature
Receiver: Message → Hash → Compare with Signature Decrypted by Public Key
```

🔗 Google Search

---

10. RSA vs. DSA

- **RSA:**
  - Based on integer factorization.
  - Can be used for both encryption & signature.
  - Slower for signature generation, faster for verification.
- **DSA:**
  - Based on discrete logarithm problem.
  - Only for signatures (not encryption).
  - Faster signing, slower verification.
- **Summary:** RSA = versatile, DSA = specialized for signatures.

🔗 Google Search

---

11. RSA Digital Signature Scheme (with Diagram)

- **Steps:**
  - Generate RSA keys (public & private).
  - Hash message → encrypt with private key = signature.
  - Verify using public key.
- **Diagram:**

   - **Sender:** M → H(M) → Encrypt with d (private key).
   - **Receiver:** M → H(M) → Compare with Decrypt(Signature, e).

- **Applications:** Email security, legal documents, financial transactions.

📎 Google Search

📓 Topic 11: Entity Authentication

---

1. Three Main Concerns with Passwords for Authentication & Social Engineering Attack

Entity Authentication: Proving the identity of a user or system. The simplest method is using passwords, but they come with major concerns:

- **Concerns with Passwords:**
   1. **Weak Passwords:** Users often choose simple, guessable passwords (e.g., "123456", "password").
   2. **Password Reuse:** Same password used across multiple platforms increases risk.
   3. **Password Storage/Transmission:** If stored in plaintext or transmitted without encryption, attackers can steal them.

- **Social Engineering Attack on Passwords:**
   - Involves tricking people into revealing their passwords instead of directly attacking the system.
   - Examples: Phishing emails, fake login pages, phone calls pretending to be IT staff.
   - Defense: User awareness, 2FA, avoiding suspicious requests.

📎 Google Search

---

2. Classification of Password Attacks

Password attacks are generally divided into two broad categories:

- **1. Active Attacks (Online Attacks):**
   - Direct interaction with the authentication system.
   - **Types:**
      - **Brute Force Attack:** Trying all possible passwords.
      - **Dictionary Attack:** Using lists of common passwords.
      - **Credential Stuffing:** Using leaked passwords from other sites.
      - **Password Guessing:** Based on user info (birthday, pet’s name).

- **2. Passive Attacks (Offline Attacks):**
   - Attacker gains hashed/encrypted password file, then attempts cracking offline.
   - **Types:**

- **Rainbow Table Attack:** Pre-computed hash dictionary.
- **Hash Cracking:** Using GPU power to compute hashes.
- **Phishing/Keylogging:** Stealing passwords silently.

- **Mitigation:**
    - Strong password policies, password hashing (bcrypt, Argon2), salting, 2FA/MFA, monitoring login attempts.

🔗 Google Search

📓 Topic 12: Security at the Application Layer – PGP & S/MIME

---

1. Define PGP (Pretty Good Privacy)

- **Definition:** PGP is an encryption program developed by Phil Zimmermann in 1991 for secure email communication.

- **How It Works:**
    - Combines **symmetric encryption** (for fast message encryption) with **asymmetric encryption** (for secure key exchange).
    - Uses **digital signatures** for authentication and non-repudiation.

- **Features:**
    - Provides **confidentiality, integrity, and authentication**.
    - Uses hybrid cryptography:
        - Symmetric (AES, IDEA, 3DES) → encrypts actual message.
        - Asymmetric (RSA, ECC) → encrypts the session key.
    - Includes **compression** to reduce data size and improve performance.

- **Applications:** Email security, file encryption, data integrity.

🔗 Google Search

---

2. MIME Attack on Diffie–Hellman Key Exchange Process

- **MIME (Multipurpose Internet Mail Extensions):** Extends email format to support text, images, attachments.

- **Attack Scenario:**

  - In a Diffie–Hellman key exchange, an attacker can intercept the key negotiation between sender and receiver.
  - MIME-based attacks may involve inserting malicious attachments or altering content during exchange.
  - If email applications don't verify digital signatures, attackers can replace legitimate MIME parts with harmful ones.

- **Defense:**

  - Use signed MIME (S/MIME) with digital signatures.
  - Implement authenticity checks.

🔗 Google Search

---

3. What is MIME? List the Limitations of SMTP/RFC 822

- **MIME (Multipurpose Internet Mail Extensions):** Standard for formatting non-text data in emails (audio, video, images, applications).

- **Why Needed:** SMTP and RFC 822 were text-only protocols.

- **Limitations of SMTP/RFC 822:**

  1. Supports only 7-bit ASCII text (no binary data like images or files).
  2. No encryption or authentication support.
  3. Limited attachment capability.
  4. Not designed for modern security requirements.

- **MIME Solves These By:**

  - Encoding binary files (Base64).
  - Adding headers for content type, transfer encoding.
  - Supporting multipart messages.

🔗 Google Search

---

4. Why is E-mail Compatibility Function in PGP Needed?

- **Problem:** Traditional email systems often alter characters (extra spaces, line breaks, encoding).

- **Solution in PGP:**

  - Uses **radix-64 encoding (Base64)** to convert binary output of encryption into text-only format that is safe for email transport.
  - Ensures encrypted messages survive email transmission without corruption.

- **Benefit:** Makes PGP-encrypted emails universally compatible across email systems.

🔗 Google Search

📔 Topic 13: Security at the Transport Layer – SSL & TLS

---

1. Explain Diffie–Hellman Key Exchange Algorithm

- **Definition:** A method for two parties to securely generate a shared secret key over an insecure channel without sending the key directly.

- **Steps:**

  1. Choose a large prime number q and a primitive root a.
  2. Alice picks private key X_A, computes public key $Y_A = a^{X_A} \mod q$.
  3. Bob picks private key X_B, computes public key $Y_B = a^{X_B} \mod q$.
  4. Exchange public keys.
  5. Alice computes shared key: $K = Y_B^{X_A} \mod q$.
  6. Bob computes shared key: $K = Y_A^{X_B} \mod q$.
  - Both result in the same key due to modular arithmetic.

- **Security Basis:** Discrete Logarithm Problem (hard to compute X from $a^X \mod q$).

🔗 Google Search

---

2. What is a Socket? Describe SSL (Secure Socket Layer) Algorithm

- **Socket:** Endpoint for sending/receiving data between two processes across a network (identified by IP + port).

- **SSL Algorithm:**

  - **Record Layer:** Provides fragmentation, compression, encryption.

  - **Handshake Layer:** Establishes secure session by authenticating server, optionally client.

  - **Handshake Steps:**

    1. ClientHello → supported ciphers, random nonce.
    2. ServerHello → chosen cipher, server certificate.
    3. Key Exchange → Diffie–Hellman / RSA used.
    4. Session Keys derived → secure communication begins.

  - **Security:** Provides confidentiality, authentication, and integrity.

🔗 Google Search

---

3. Benefits of Using SSL (and TLS)

- **Confidentiality:** Encryption prevents eavesdropping.
- **Integrity:** Message authentication codes ensure no tampering.

- **Authentication:** Certificates verify server (and client if needed).
- **Trust:** Users trust websites with HTTPS (SSL/TLS enabled).
- **Protection Against Attacks:** Helps prevent MITM, replay attacks, and session hijacking.
- **Use Cases:** HTTPS, online banking, VPNs, secure email.

📎 Google Search

---

4. Diffie–Hellman Key Exchange Example (a=5, q=11, X_A=2, X_B=3)

- Given: ($a=5, q=11, X_A=2, X_B=3$).

1. Compute Alice's public key: ($Y_A = a^{X_A} \mod q = 5^2 \mod 11 = 25 \mod 11 = 3$).

2. Compute Bob's public key: ($Y_B = a^{X_B} \mod q = 5^3 \mod 11 = 125 \mod 11 = 4$).

3. Shared secret key:

    - Alice: ($K = Y_B^{X_A} \mod q = 4^2 \mod 11 = 16 \mod 11 = 5$).
    - Bob: ($K = Y_A^{X_B} \mod q = 3^3 \mod 11 = 27 \mod 11 = 5$). ✔ Both get the same key ($K = 5$).

📎 Google Search

---

5. Diffie–Hellman Example (q=71, a=7, X_A=5, X_B=12)

- Given: ($q=71, a=7, X_A=5, X_B=12$).

1. Alice's public key: ($Y_A = 7^5 \mod 71 = 16807 \mod 71 = 61$).

2. Bob's public key: ($Y_B = 7^{12} \mod 71$).

    - Compute: ($7^{12} \mod 71 = 19$).

3. Shared secret key:

    - Alice: ($K = Y_B^{X_A} \mod 71 = 19^5 \mod 71 = 6$).
    - Bob: ($K = Y_A^{X_B} \mod 71 = 61^{12} \mod 71 = 6$). ✔ Shared secret key = **6**.

📎 Google Search

---

6. SSL Handshake Protocol

- **Goal:** Establish secure session keys before transmitting data.

- **Steps:**

    1. **ClientHello:** Lists supported ciphers, random value.
    2. **ServerHello:** Selects cipher suite, sends certificate.
    3. **Key Exchange:** Server proves identity, exchange of session key (RSA or DH).
    4. **Finished Messages:** Both sides verify handshake success.

- **Result:** Session established with symmetric keys for fast encryption.

🔗 Google Search

---

7. SSL Connection State Parameters

SSL connection maintains several security parameters:

- **Client Random & Server Random:** Random numbers exchanged for session key generation.
- **Session ID:** Identifies session uniquely.
- **Cipher Suite:** Defines algorithms used (AES, 3DES, SHA).
- **Master Secret:** Generated during handshake.
- **MAC Secrets:** Used for message integrity.
- **Keys:** Separate keys for encryption/decryption in both directions.

🔗 Google Search

---

8. Secure Shell (SSH) Protocol

- **Definition:** A cryptographic network protocol for secure remote login and file transfer.
- **Functions:**
  - Encrypts traffic (confidentiality).
  - Authenticates server & user (public key authentication).
  - Protects against MITM & replay attacks.
- **Use Cases:**
  - Remote login to servers (Linux).
  - Secure file transfer (SCP, SFTP).
  - Port forwarding, tunneling.

🔗 Google Search

📘 Topic 14: Security at the Network Layer – IPSec

---

1. What do you mean by Security Association (SA)? Specify its Parameters

- **Security Association (SA):**
  - A set of security parameters agreed upon between two communicating parties to establish a secure channel in IPSec.
  - Each SA is **unidirectional** (one for inbound, one for outbound).
- **Parameters that Identify an SA:**
  1. **SPI (Security Parameter Index):** Unique identifier in packet headers.

2. **Destination IP Address:** Identifies peer system.
3. **Security Protocol Identifier:** Specifies if AH (Authentication Header) or ESP (Encapsulating Security Payload) is used.

- **Other Parameters (inside SA):**
  - Cryptographic algorithms (AES, 3DES).
  - Authentication algorithms (HMAC-SHA).
  - Lifetime of SA.

📎 Google Search

---

2. Explain IPSec ESP (Encapsulating Security Payload) Format

- **ESP (Encapsulating Security Payload):** Provides **confidentiality, integrity, authentication, and optional replay protection**.

- **ESP Packet Format:**
  1. **Security Parameters Index (SPI):** Identifies SA.
  2. **Sequence Number:** Protects against replay attacks.
  3. **Payload Data:** Encrypted message (AES, 3DES).
  4. **Padding:** Align data to block size.
  5. **Next Header:** Identifies type of data in payload (e.g., TCP, UDP).
  6. **Authentication Data (Optional):** Provides data integrity.

- **Advantage:** Unlike AH (Authentication Header), ESP can encrypt the payload.

📎 Google Search

---

3. Applications and Benefits of IPSec

- **Applications:**
  - Virtual Private Networks (VPNs).
  - Secure site-to-site connections.
  - Remote access to corporate networks.
  - Secure email & VoIP.

- **Benefits:**
  - Provides **end-to-end security** at the network layer.
  - Protects against spoofing, replay, eavesdropping.
  - Transparent to applications → no need to modify software.
  - Supports **tunnel mode** (entire packet encrypted) and **transport mode** (payload encrypted).

📎 Google Search

---

4. Tunnel Mode vs. Transport Mode of IPSec

- **Transport Mode:**
  - Only the payload of the IP packet is encrypted/authenticated.
  - Header remains visible.
  - Used for end-to-end communication (host-to-host).
- **Tunnel Mode:**
  - Entire IP packet (header + payload) encrypted.
  - A new IP header is added.
  - Used for VPNs, gateway-to-gateway communication.
- **Summary:**
  - Transport → End-to-end security.
  - Tunnel → Gateway-to-gateway, secure tunneling.

🔗 Google Search

---

5. IPSec Protocol for Authentication and Data Integrity

- IPSec provides **two main protocols**:
  1. **AH (Authentication Header):**
     - Provides authentication, integrity, and replay protection.
     - Does not provide encryption (no confidentiality).
     - Protects entire packet except fields that change in transit.
  2. **ESP (Encapsulating Security Payload):**
     - Provides authentication, integrity, confidentiality.
     - More widely used because it supports encryption.
- **Data Integrity:** Achieved using **HMAC (Hash-based Message Authentication Code)** with SHA-1 or SHA-256.
- **Authentication:** Ensures the packet is from a legitimate source.

🔗 Google Search

📓 Topic 15: System Security

---

1. Define Intrusion and Methods of Intrusion Detection

- **Intrusion:** Any unauthorized attempt to access, manipulate, or disable a computer system, network, or data.
- **Types of Intrusions:**

- Unauthorized login attempts.
- Malware infections.
- Denial of Service (DoS) attacks.
- Exploiting software vulnerabilities.

- **Intrusion Detection Methods (IDS – Intrusion Detection Systems):**

  1. **Signature-Based IDS:** Detects known attack patterns (like antivirus).
  2. **Anomaly-Based IDS:** Detects deviations from normal behavior (useful for zero-day attacks).
  3. **Host-Based IDS (HIDS):** Monitors activities on a single host (e.g., log files, file integrity).
  4. **Network-Based IDS (NIDS):** Monitors network traffic (packet inspection).

📎 Google Search

---

2. Firewall – Definition, Merits, and Demerits

- **Definition:** A firewall is a security system (hardware, software, or both) that monitors and controls incoming/outgoing network traffic based on security rules.

- **Merits (Advantages):**

  - Protects internal network from external threats.
  - Blocks unauthorized access.
  - Can log and monitor traffic.
  - Implements access control policies.

- **Demerits (Disadvantages):**

  - Cannot protect against internal threats (if attacker is inside).
  - Cannot prevent attacks via allowed applications (e.g., malware in email).
  - Cannot detect encrypted malicious traffic.

📎 Google Search

---

3. Worms and Digital Immune System

- **Worm:** A self-replicating malicious program that spreads across networks without human action.

  - Example: Code Red, SQL Slammer.
  - Can cause bandwidth exhaustion and system crashes.

- **Digital Immune System (DIS):**

  - A defense model inspired by the biological immune system.
  - Detects new, unknown viruses → creates a defense → distributes updates across systems.
  - Example: Symantec used this concept to automatically detect and counter new malware.

📎 Google Search

---

4. Firewall & Its Limitations – Why Corporate Houses Use Multiple Firewalls

- **Firewall Limitations:**
  - Cannot protect against social engineering attacks.
  - Does not stop insider attacks.
  - Limited visibility into encrypted traffic.
  - May not detect zero-day exploits.
- **Why Multiple Firewalls?**
  - **Defense-in-Depth:** Multiple firewalls at different layers (perimeter, internal).
  - **Segmentation:** Separate sensitive servers (finance, HR) from rest of the network.
  - **High Availability:** Redundancy for failover.
  - **Layered Security:** Different vendors for different strengths.

🔗 Google Search

---

5. How to Ensure Router Security

Routers are critical as entry/exit points for networks. Weak router security = vulnerable network.

- **Methods to Secure Routers:**
  - Change default admin usernames/passwords.
  - Disable unused ports/services.
  - Use strong encryption for management (SSH, not Telnet).
  - Apply firmware updates regularly.
  - Implement access control lists (ACLs).
  - Enable firewall features in routers.
  - Monitor router logs and traffic.
- **Enterprise Security:** Use IDS/IPS integrated with routers for enhanced protection.

🔗 Google Search

📓 Topic 16: Random Number Generator (RNG)

---

1. What is a Pseudorandom Generator (PRNG)? Give an Example Describing How it Works

- **Random Numbers in Cryptography:**
  - Essential for key generation, initialization vectors (IVs), session keys, nonces.
  - If randomness is predictable → cryptosystem becomes weak.
- **True Random Number Generator (TRNG):**

- Uses physical sources (radioactive decay, thermal noise, mouse movements).
- Unpredictable but harder to implement.

- **Pseudorandom Number Generator (PRNG):**
  - Uses mathematical algorithms to generate sequences of numbers that “appear random.”
  - Deterministic: If seed is known, the sequence is reproducible.
  - Used when high speed and efficiency are required.

- **Properties of PRNG in Cryptography:**
  - Must pass statistical randomness tests.
  - Must be unpredictable (next number cannot be guessed).
  - Should have a long period before repeating.

- **Example – Linear Congruential Generator (LCG):**
  - Formula: $X_{n+1} = (aX_n + c) \mod m$.
  - Parameters (a, c, m) define generator behavior.
  - If chosen well, produces long sequence before repeating.
  - **Weakness:** Predictable once some outputs are known → not secure for cryptography.

- **Cryptographically Secure PRNG (CSPRNG):**
  - Example: Fortuna, Yarrow, ANSI X9.17.
  - Based on secure hash functions or block ciphers.
  - Strong enough for key generation in AES, RSA, TLS.

🔗 Google Search

📓 Topic 17: Secured Electronic Transaction (SET)

---

1. Define SET. Write Down the Features of SET

- **Definition:**
  - **SET (Secure Electronic Transaction)** is a security protocol developed by VISA and MasterCard (1996) for securing online credit card payments.
  - It ensures that credit card details are transmitted safely over insecure networks like the Internet.
  - Uses **digital signatures, digital certificates, and encryption**.

- **Features of SET:**
  1. **Confidentiality:** Card details are encrypted (RSA + DES/AES).
  2. **Integrity:** Digital signatures protect message from tampering.
  3. **Authentication:**
     - Cardholder and merchant identities verified using **digital certificates** (X.509).

4. **Dual Signature:**
    - Splits order info and payment info → merchant cannot see card details, bank cannot see order details.
5. **Interoperability:** Based on open standards (X.509, RSA).
6. **Non-repudiation:** Both merchant and customer cannot deny their actions.
7. **Scalability:** Works globally for e-commerce.

🔗 Google Search

---

2. Steps Involved in a SET Transaction

A SET transaction has multiple parties: **Customer, Merchant, Payment Gateway, Bank, and Certificate Authority (CA).**

- **Step 1: Initialization**
    - Customer and merchant obtain digital certificates from CA.
- **Step 2: Purchase Request**
    - Customer selects goods/services online.
    - Customer creates two messages:
        1. Order Information (OI) for merchant.
        2. Payment Information (PI) for bank.
    - A **Dual Signature** is created to bind OI and PI together.
- **Step 3: Merchant Processing**
    - Merchant verifies customer's certificate and dual signature.
    - Merchant forwards PI securely to payment gateway/bank.
- **Step 4: Authorization**
    - Payment gateway requests bank to authorize payment.
    - Bank checks card validity, balance, etc.
    - Authorization response sent back.
- **Step 5: Capture**
    - Merchant requests payment gateway to capture funds.
    - Payment is transferred from customer's bank to merchant's account.
- **Step 6: Completion**
    - Merchant ships goods/services.
    - Both merchant and customer get receipts.

- **Security Highlight:**
  - Merchant never sees card details.
  - Bank never sees order details.

🔗 Google Search

📓 Topic 18: Encipherment Using Modern Symmetric-Key Ciphers

---

1. Mention the Weakness of Electronic Code Book (ECB) Mode

- **Electronic Code Book (ECB) Mode:**
  - Simplest block cipher mode of operation.
  - Message divided into fixed-size blocks (e.g., 128 bits for AES).
  - Each block encrypted **independently** using the same key: $C_i = E_K(P_i)$
  - Where P_i = plaintext block, C_i = ciphertext block.
- **Weaknesses:**
  1. **Pattern Leakage:** Identical plaintext blocks → identical ciphertext blocks.
     - Example: Encrypting an image with ECB preserves visible patterns.
  2. **No Diffusion:** Changing one block affects only that block (no chaining).
  3. **Replay Vulnerability:** Same plaintext always produces same ciphertext → attacker can replay captured ciphertexts.
  4. **Not Suitable for Long Data:** Repeated messages expose structure.
- **Visual Example:**
  - Encrypting a bitmap image (e.g., Linux Tux Penguin) with ECB → encrypted image still shows penguin outline because repeated pixel patterns remain visible.
- **Why Not Used in Practice:**
  - Insecure for most applications.
  - Modern systems prefer **CBC, CFB, OFB, or CTR** modes.

🔗 Google Search

📓 Topic 19: Web Security

---

1. What is Sandbox?

- **Definition:** A **sandbox** is a controlled environment where software can run in isolation from the main system.
- **Purpose:** Prevents malicious or untrusted code from harming the system.
- **Example:**
  - Browser sandboxes (Chrome/Edge run each tab in isolated processes).
  - Mobile app sandboxes (Android/iOS apps run with restricted permissions).
- **Benefit:** Protects against malware, zero-day exploits, and unsafe scripts.

🔗 Google Search

---

2. What are Sandbox Environments?

- A **sandbox environment** is a virtual test setup where new programs, code, or files are executed safely.
- **Types:**
  - **Software Sandbox:** Applications run with limited system access.
  - **Hardware Sandbox:** Isolated hardware environment.
  - **Cloud Sandbox:** Online sandbox for testing files/emails for malware.
- **Use Case:** Security researchers analyze suspicious files in sandbox before allowing them on a live network.

🔗 Google Search

---

3. Benefits of Sandboxing

- Protects production systems from malicious or buggy software.
- Prevents malware from spreading across networks.
- Provides safe environment for testing software updates.
- Helps detect zero-day exploits.
- Allows IT admins to analyze unknown files without risk.

🔗 Google Search

---

4. Describe Intruder in Network Security

- **Intruder:** An entity (human or malicious software) that attempts to gain unauthorized access to computer systems.
- **Types:**
  - **Masquerader:** Outsider who pretends to be a legitimate user.
  - **Misfeasor:** Legitimate user misusing privileges.

  - **Clandestine User:** Attacker who hides their identity/activities.
- **Intruder Goals:** Data theft, denial of service, spreading malware.
- **Defense:** Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), firewalls, logging, and monitoring.

🔗 Google Search

---

5. Denial-of-Service (DoS) and Distributed DoS (DDoS) Attacks

- **DoS Attack:** Overwhelms a server or network with traffic, making it unavailable.
  - Example: Flooding a web server with fake requests.
- **DDoS Attack:** Multiple compromised computers (botnets) launch a coordinated attack.
  - Example: Mirai Botnet DDoS (2016) took down Twitter, Netflix, GitHub.
- **Types of DDoS Attacks:**
  - **Volume-Based:** Flood with high traffic (UDP flood, ICMP flood).
  - **Protocol Attacks:** Exploit weaknesses in protocols (SYN flood, Ping of Death).
  - **Application Layer Attacks:** Target web applications (HTTP flood).
- **Defenses:**
  - Firewalls & rate limiting.
  - Load balancing & CDNs.
  - Specialized DDoS protection services (Cloudflare, Akamai).

🔗 Google Search

---

6. VPN (Virtual Private Network) & Network Security Policy/Management

- **VPN (Virtual Private Network):**
  - Creates an **encrypted tunnel** over a public network (like the Internet).
  - Ensures **confidentiality, integrity, authentication** of data.
  - Used by remote workers, businesses, and for secure browsing.
- **Network Security Policy:**
  - A set of rules defining how data, resources, and services are protected.
  - Covers access control, password rules, acceptable use, incident response.
- **Network Security Management:**
  - Involves monitoring, configuring firewalls, IDS/IPS, patch management, and compliance.
  - Ensures security policy is enforced consistently.

🔗 Google Search

📓 Topic 20: Short Notes

---

1. S/MIME (Secure/Multipurpose Internet Mail Extensions)

- **Definition:** Standard for secure email communication, built on top of MIME.
- **Features:**
  - Provides **confidentiality** (encryption with RSA/AES).
  - Provides **authentication & integrity** (digital signatures).
  - Uses **X.509 certificates** for trust.
- **Applications:** Secure email in Outlook, Thunderbird, Gmail (corporate).
- **Benefit:** End-to-end encryption of email messages.

🔗 Google Search

---

2. Email Security

- **Threats:** Phishing, spam, malware attachments, spoofing.
- **Techniques for Security:**
  - **Encryption (PGP, S/MIME).**
  - **Digital signatures** for authenticity.
  - **Spam filters, antivirus.**
  - **Authentication protocols:** SPF, DKIM, DMARC.
- **Goal:** Protect confidentiality, integrity, and authenticity of emails.

🔗 Google Search

---

3. ESP (Encapsulating Security Payload – IPSec)

- **Function:** Part of IPSec protocol suite.
- **Provides:** Confidentiality (encryption), integrity, authentication, anti-replay protection.
- **Packet Fields:** SPI, Sequence Number, Payload Data, Padding, Next Header, Authentication Data.
- **Advantage:** Protects message content from eavesdropping.

🔗 Google Search

---

4. Steganography

- **Definition:** Hiding secret information inside non-secret data (images, audio, video).
- **Example:** Text hidden in the least significant bits (LSB) of an image.

- **Difference from Cryptography:** Cryptography scrambles content, steganography hides content's existence.
- **Use Cases:** Covert communication, digital watermarking.

🔗 Google Search

---

5. ECC (Elliptic Curve Cryptography)

- **Definition:** Asymmetric cryptography using elliptic curves over finite fields.
- **Advantages:**
  - Strong security with small key sizes (256-bit ECC ≈ 3072-bit RSA).
  - Efficient for mobile, IoT.
- **Applications:** SSL/TLS, cryptocurrencies (Bitcoin, Ethereum), secure messaging.

🔗 Google Search

---

6. Digital Immune System (DIS)

- **Concept:** Inspired by biological immune systems.
- **Working:**
  - Detects unknown malware → creates signature → distributes defense across systems.
- **Example:** Antivirus companies (Symantec) use this model.
- **Benefit:** Fast, automated malware response.

🔗 Google Search

---

7. Diffie–Hellman Key Exchange

- Secure key exchange protocol over insecure channels.
- Based on **discrete logarithm problem**.
- Used in SSL/TLS, VPNs.
- Vulnerable to **MITM attack** if no authentication.

🔗 Google Search

---

8. Stream Cipher vs. Block Cipher

- **Stream Cipher:** Encrypts bit/byte at a time (RC4). Fast, lightweight.
- **Block Cipher:** Encrypts fixed-size blocks (AES, DES). Stronger but slower.
- **Comparison:** Stream better for real-time, Block better for file encryption.

🔗 Google Search

---

9. Digital Signature Standard (DSS)

- **NIST Standard (1994).**
- Specifies algorithms for digital signatures: DSA, RSA, ECDSA.
- Provides **integrity, authenticity, non-repudiation**.
- Used in US government systems.

📎 Google Search

---

10. Security Attacks

- **Active Attacks:** Modification, fabrication, DoS.
- **Passive Attacks:** Eavesdropping, traffic analysis.
- **Goal of Attacker:** Break confidentiality, integrity, availability.

📎 Google Search

---

11. UNIX Password Scheme

- **UNIX systems:** Store encrypted passwords in `/etc/shadow`.
- Uses **salted hashing** to prevent dictionary attacks.
- Modern systems use **bcrypt, SHA-512** for stronger protection.
- Weak if short/guessable passwords used.

📎 Google Search

---

12. SSL (Secure Socket Layer)

- Transport layer security protocol.
- Provides encryption, authentication, and integrity.
- Replaced by TLS but still commonly called "SSL."
- Used in HTTPS (websites with padlock 🔒).

📎 Google Search

---

13. Hash Function

- Converts arbitrary input → fixed-size digest.
- Properties: Pre-image resistance, collision resistance, avalanche effect.
- Examples: MD5, SHA-256, SHA-512.
- Used in password storage, digital signatures, blockchain.

📎 Google Search

---

14. IPSec ESP Format

- Already covered in **Topic 14** (Encapsulating Security Payload).

- Key fields: SPI, Sequence Number, Payload Data, Authentication Data.

📎 Google Search

---

15. Cryptanalysis

- The science of breaking ciphers and recovering plaintext/keys.
- Types:
  - Ciphertext-only attack.
  - Known-plaintext attack.
  - Chosen-plaintext attack.
  - Differential & Linear Cryptanalysis.

📎 Google Search

---

16. Differential Cryptanalysis

- A type of cryptanalysis technique analyzing differences in input vs. output.
- Applied to block ciphers (e.g., DES).
- Requires large numbers of chosen plaintexts.
- Goal: Recover secret key.

📎 Google Search

---

17. PGP (Pretty Good Privacy)

- Already covered in **Topic 12**.
- Hybrid encryption for secure email.
- Uses symmetric + asymmetric cryptography + compression.

📎 Google Search

---

18. SET (Secure Electronic Transaction)

- Already covered in **Topic 17**.
- Protocol for secure online payments.

📎 Google Search

---

19. Feistel Cipher

- Symmetric cipher structure (used in DES).
- Splits block into two halves → applies substitution & permutation in rounds.
- Same design works for both encryption & decryption.

📎 Google Search

---

20. RC4 Algorithm

- A stream cipher designed by Ron Rivest (RSA Security).
- Generates pseudorandom keystream XORed with plaintext.
- Used in early SSL/TLS and WEP (now insecure).
- Weakness: Key scheduling vulnerabilities.

🔗 Google Search

---

21. PRNG (Pseudorandom Number Generator)

- Already covered in **Topic 16**.
- Deterministic algorithm generating random-like sequences.
- Example: Linear Congruential Generator (LCG).
- Cryptographically secure PRNGs needed for keys.

🔗 Google Search

---

22. Wire Pool

- A mechanism used in **cryptographically secure PRNGs** to collect entropy (randomness) from multiple sources.
- Helps improve unpredictability of random numbers.
- Example: Used in PGP's random number generation.

🔗 Google Search

---

23. X.509 Architecture Format

- Standard for **digital certificates** used in PKI.
- Includes: Version, Serial Number, Issuer, Validity, Subject, Public Key, Signature.
- Used in SSL/TLS, email security, VPNs.

🔗 Google Search

---

24. PKI (Public Key Infrastructure)

- A framework for managing digital certificates and public/private keys.
- Components: CA, RA (Registration Authority), CRL, OCSP.
- Provides authentication, confidentiality, integrity, non-repudiation.
- Basis of HTTPS, VPNs, digital signatures.

🔗 Google Search

---

25. Generic Encryption

- Refers to a **general model of encryption** → plaintext + key → encryption algorithm → ciphertext.
- Covers both symmetric and asymmetric methods.
- Used as a conceptual framework in security studies.

🔗 Google Search

---

26. S-Box (Substitution Box)

- A component of block ciphers like DES and AES.
- Provides **non-linearity (confusion)** by mapping input bits to output bits.
- Example: DES uses 8 S-boxes in each round.
- Essential for resistance against linear/differential cryptanalysis.

🔗 Google Search

---

27. 9-Box

- A variation of substitution/permutation structure in cryptography.
- Sometimes used as an educational model for demonstrating simple ciphers.
- Provides basic confusion and diffusion properties.
- Less common in practical systems compared to S-boxes.

🔗 Google Search
