13.LectureNotes

UNIT-I

1. Introduction & The Need for Security

- **Security Attack:** Any action that compromises the security of information owned by an organization. It is an intentional threat agent attempting to evade security services and violate a system's security policy.
- **Security Mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack. Examples include cryptographic algorithms, digital signatures, and access control mechanisms.
- **Security Service:** A processing or communication service provided by a system to give a specific kind of protection to system resources. Security services implement security policies and are realized by one or more security mechanisms.

The Need for Security

The explosive growth of computer networks, e-commerce, and cloud computing has transformed information into a critical strategic asset. The necessity of implementing robust network security stems from the following critical factors:

- **Vulnerability of Data in Transit:** When data travels over public networks (like the Internet), it passes through multiple intermediate routers and switches. Without security, this data is vulnerable to eavesdropping, tampering, and interception.
- **Rise in Cybercrime and Sophisticated Attacks:** The profile of attackers has shifted from amateur hobbyists to organized cybercriminal syndicates, state-sponsored actors, and hacktivists. Attacks have become automated, distributed (e.g., DDoS), and highly sophisticated.
- **Financial and Economic Impact:** A successful security breach can result in catastrophic financial losses, including direct theft of funds, intellectual property (IP) leakage, regulatory fines (GDPR, PCI-DSS compliance failures), and expensive system remediation.
- **Reputation and Trust:** For enterprise businesses, banks, and healthcare providers, trust is paramount. A publicly known data breach destroys brand reputation, leading to customer churn and loss of stakeholder confidence.
- **Protection of Critical Infrastructure:** Modern networks control vital physical systems like electrical grids, water supply chains, transportation systems, and defense communications. Securing these networks is a matter of national security.

2. Security Approaches & Principles of Security

(a) Security Approaches

Feature	Perimeter Security Approach (Traditional)	Layered Security / Defense-in-Depth Approach
Core Concept	Focuses on building a strong boundary wall (Firewall) around the network.	Implements multiple independent layers of defense throughout the system.
Assumed Trust	High trust inside the network; zero trust outside.	Assumes threats can exist both outside and inside the network (Zero Trust model).
Failure Scenario	If the perimeter is breached, the entire internal network is compromised.	If one defense layer fails, subsequent layers (host security, data encryption) contain the threat.
Components	Firewalls, Edge Routers, VPN gateways.	Firewalls, IAM, Endpoint Protection, Encrypted Databases, MFA.

(b) Principles of Security

The Core CIA Triad

1. Confidentiality: Ensures that sensitive assets are accessed only by authorized individuals. Prevention of unauthorized disclosure of information.

- *Mechanism:* Symmetric/Asymmetric encryption (AES, RSA), Access Control Lists (ACLs).

2. Integrity: Assures that information and programs are changed only in a specified and authorized manner. It prevents unauthorized modification or deletion of data.

- *Mechanism:* Cryptographic hash functions (SHA-256), Digital Signatures.

3. Availability: Ensures that systems work promptly and service is not denied to authorized users. It guarantees reliable access to data and resources.

- *Mechanism:* Redundant systems (RAID), Load balancing, DDoS mitigation techniques.

Supplementary Principles

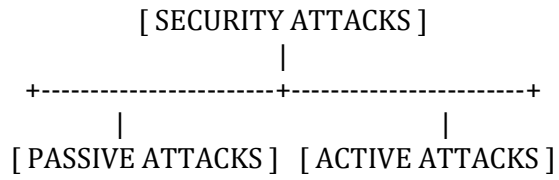
4. Authenticity: The property of being genuine and being able to be verified and trusted. It ensures that users or systems are exactly who they claim to be.

5. Non-Repudiation: Prevents either sender or receiver from denying a transmitted message. When a message is sent, the receiver can prove that the alleged sender did in fact send the message.

6. Accountability: The security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports non-repudiation, fault isolation, and intrusion detection through audit logs.

3. Types of Security Attacks

(a) Passive vs. Active Attacks



- Goal: Eavesdropping / Monitoring
  - Hard to detect (no data altered)
  - Prevented via Encryption

- Goal: Alteration of data / system status
  - Easy to detect via system anomalies
  - Prevented via Firewalls / IDS / IPS

Passive Attacks: These are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the attacker is to obtain information that is being transmitted.

- *Characteristics:* They do not alter data or system resources, making them extremely difficult to detect. Prevention (via encryption) rather than detection is the primary defense.
- *Examples:* Release of message contents (reading an unencrypted email), Traffic Analysis (observing frequency and length of messages to guess communication patterns).

Active Attacks: These attacks involve some modification of the data stream or the creation of a false stream.

- *Characteristics:* They involve physical or logical alteration of system states and data. They are easier to detect through logs and performance drops, but harder to prevent absolutely.
- *Examples:* Modification of messages, Resource exhaustion.

(b) Classification of Specific Attacks

- **Snooping / Interception:** A passive attack where an unauthorized entity gains access to an asset. This is an attack on *Confidentiality* (e.g., packet sniffing using Wireshark).
- **Modification / Alteration:** An active attack where an unauthorized party tampers with data in transit or data at rest. This is an attack on *Integrity* (e.g., changing an electronic fund transfer amount from $100 to $10,000).
- **Masquerading (Spoofing):** Occurs when one entity pretends to be a different entity. An active attack that usually includes other forms of active attacks (e.g., IP spoofing or Phishing sites mimicking bank logins).
- **Replay Attack:** Involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (e.g., intercepting an encrypted login token and resending it later to gain unauthorized access).
- **Denial of Service (DoS):** Prevents or inhibits the normal use or management of communications facilities. This targets *Availability* by flooding a network layer with illegitimate traffic or crashing services.

4. Security Services & Security Mechanisms (X.800)

Five Main Categories of Security Services (X.800)

1. **Authentication:** Provides assurance that the communicating entity is the correct one.
   - *Peer-entity authentication:* Verifies identities during a connection setup.

- *Data-origin authentication:* Verifies the source of a data unit (does not protect against duplication).

2. **Access Control:** The prevention of unauthorized use of a resource (identifies who can access what files/servers).
3. **Data Confidentiality:** Protection of data from unauthorized disclosure (includes connection confidentiality, connectionless confidentiality, and traffic-flow confidentiality).
4. **Data Integrity:** Assures that data received is exactly as sent by an authorized entity (without insertion, deletion, or modification).
5. **Non-Repudiation:** Provides proof of the origin or delivery of data.

Specific Security Mechanisms

To implement these services, X.800 defines specific mechanisms that can be embedded into the appropriate protocol layer:

- **Encipherment:** Using mathematical algorithms to transform data into an unreadable form (Encryption).
- **Digital Signature:** Appending a cryptographic value to a data unit that allows the recipient to prove the source and integrity of the data.
- **Access Control Mechanisms:** Schemes like ACLs, Role-Based Access Control (RBAC), and passwords to enforce rights.
- **Data Integrity Mechanisms:** Appending a checksum or Message Authentication Code (MAC) to data.
- **Authentication Exchange:** Mechanisms intended to ensure the identity of an entity by means of information exchange (e.g., challenge-response, cryptographic handshakes).
- **Traffic Padding:** Inserting extra, random bits into data streams to frustrate traffic analysis attacks.
- **Routing Control:** Selecting specific physically secure routes for certain data.
- **Notarization:** The use of a trusted third party to assure certain properties of a data exchange (e.g., digital notary).

Service-to-Mechanism Mapping Matrix

Security Service	**Primary Supporting Mechanism(s)**
Authentication	Encipherment, Digital Signature, Authentication Exchange
Access Control	Access Control Mechanisms (ACLs, Tokens)
Data Confidentiality	Encipherment, Routing Control

Security Service	Primary Supporting Mechanism(s)
Data Integrity	Digital Signature, Data Integrity Mechanisms (Hashing/MAC)
Non-Repudiation	Digital Signature, Notarization

5. A Model for Network Security

Structural Architecture of the Model

The Network Security Model explains how information is securely transmitted over an unsecure communication channel in the presence of potential adversaries.

The model consists of four primary components:

1. **A Principal (Sender):** The entity wishing to transmit data securely.
2. **A Secret Information Provider / Trusted Third Party (TTP):** An independent authority responsible for distributing secret keys, certificates, or arbitrating disputes (e.g., Certificate Authority or Key Distribution Center).
3. **The Unsecure Communication Channel:** The physical or logical path (e.g., Internet, Wi-Fi) through which data moves, subject to exploitation.
4. **The Adversary:** An opportunistic threat agent capable of intercepting, altering, or destroying data on the channel.

The Four Basic Tasks in Designing a Secure Service

To implement this model effectively, any network security solution must execute four core operational steps:

1. Design a Security Transformation Algorithm

A robust cryptographic algorithm must be designed or chosen to transform the plain text into ciphertext. This transformation must be mathematically complex enough so that an opponent cannot invert the process without the corresponding key, even if they know the algorithm itself (Kerckhoffs's Principle).

2. Generate Secret Information (Keys)

The transformation algorithm requires an input factor independent of the data: a key. Systems must securely generate high-entropy keys (K) that serve as the foundation of security.

- *Symmetric Key Model:* Same key used by sender and receiver:

  Ciphertext = E_K(Plaintext)
- *Asymmetric Key Model:* A mathematically linked public/private key pair is used.

3. Develop Methods for Key Distribution

An infrastructure must be built to securely distribute the secret keys or shared information to the principals without the adversary intercepting them. This often involves using a Trusted Third Party (like Diffie-Hellman Key Exchange over TLS, or Kerberos ticket grants).

4. Specify a Protocol for Operation

A structured communication protocol must be established between the entities. The protocol dictates the sequence of actions, formatting, and structural synchronization required to use the security algorithms and keys over the unsecure channel (e.g., HTTPS, IPsec, SSH).

Part-2

Cryptographic Concepts & Terminology

Cryptographic Core Terms

- **Plaintext:** The original, readable message or data that is fed into an algorithm as input.
- **Ciphertext:** The scrambled, unreadable message produced as the output of an encryption algorithm. It depends entirely on the plaintext and the encryption key.
- **Encryption:** The process of converting plaintext into ciphertext using a cryptographic algorithm and a specific key.
- **Decryption:** The reverse process of encryption; it converts ciphertext back into plaintext using a cryptographic algorithm and a matching key.

Key Size and Key Range

- **Key Size:** This refers to the length of the cryptographic key measured in bits (e.g., 128-bit, 256-bit, 2048-bit).
- **Key Range:** This is the total number of possible key combinations available for a given key size. If a key size is nbits, the total key range (or key space) is 2^n.

Significance in Cryptographic Strength

The security of modern cryptography relies on the assumption that the algorithm is publicly known, but the key is kept secret (**Kerckhoffs's Principle**).

- **Resistance to Brute-Force:** If the key size is too small (e.g., DES with a 56-bit key, yielding 2^56 combinations), modern computational clusters can try every possible key within hours.
- **Exponential Growth:** Increasing the key size by just 1 bit doubles the key range. For instance, moving from AES-128 to AES-256 does not just double the security; it increases the key space by a factor of 2^128, making brute-force attacks mathematically infeasible with current physical laws.

2. Classical Encryption: Substitution vs. Transposition

Substitution Techniques

In a substitution technique, the letters of plaintext are replaced by other letters, numbers, or symbols.

1. Monoalphabetic Cipher (Caesar Cipher)

Each plaintext letter is shifted by a fixed number of positions down the alphabet.

- *Mathematical Formula:*

  C = (P + k) mod 26
  P = (C - k) mod 26
  (Where P is Plaintext, C is Ciphertext, and kis the key).
- *Example:*If k = 3, the plaintext word DATA becomes ciphertext GDWD.

2. Polyalphabetic Cipher (Vigenère Cipher)

Uses a repeating keyword to determine shifting, meaning a single letter in the plaintext can map to multiple different letters in the ciphertext depending on its position.

- *Example:* Plaintext ATTACK, Key KEY (repeated as KEYKEY). The shift changes for every letter based on the numerical value of the key letters, effectively flattening frequency analysis attacks.

(b) Transposition Techniques

Transposition techniques do not replace letters; instead, they achieve encryption by performing a permutation (reordering) on the plaintext letters.

1. Rail Fence Cipher

The plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.

- *Example:* Plaintext SECRET with a rail depth of 2:

Plaintext

S .C .E .

. E .R . T

Ciphertext: SCEERT

2. Row Transposition Cipher

The plaintext is written in a grid of fixed row length, and the columns are read back out in a scrambled order dictated by a key.

- *Example:* Key = 3 1 4 2 (tells you the order to read the columns).

Plaintext

Key: 3 1 4 2

Row1: C O D E

Row2: B R E A

Row3: K D O W

Reading columns in order 1, 2, 3, 4: ORD REA CBK EAW

Fundamental Difference

- **Substitution** hides the *identity* of the characters but preserves their *positions*. It relies on confusion.
- **Transposition** hides the *position* of the characters but preserves their *identity*. It relies on diffusion.

3. Symmetric vs. Asymmetric Key Cryptography

Detailed Comparison

Feature	Symmetric Key Cryptography	Asymmetric Key Cryptography
Number of Keys	A single shared secret key is used for both encryption and decryption.	A mathematically linked key pair: a **Public Key** (shared openly) and a **Private Key** (kept secret).
Key Distribution	**High Risk:** The secret key must be shared securely between parties before communication.	**Low Risk:** Public keys can be distributed over unsecure channels; private keys never move.
Computational Speed	Very fast and highly optimized for hardware performance. Suitable for bulk data.	Mathematically intensive and significantly slower (often 100 to 1000 times slower than symmetric).

Feature	Symmetric Key Cryptography	Asymmetric Key Cryptography
Scalability	Poor for large networks. For n users, the network requires $\frac{n(n-1)}{2}$ unique keys.	Highly scalable. For n users, the network only requires $2n$ total keys.
Core Use Cases	Data at rest encryption, bulk data transit (e.g., AES, 3DES, Blowfish).	Key exchange, digital signatures, identity authentication (e.g., RSA, ECC, Diffie-Hellman).

Combining the Two (Hybrid Cryptography)

Because symmetric encryption is fast but suffers from key distribution issues, and asymmetric encryption is secure but slow, real-world protocols (like HTTPS/TLS) use a **hybrid approach**:

1. Asymmetric cryptography is used at the start of a session to securely verify identity and exchange a temporary symmetric key.
2. That temporary symmetric key (session key) is then used to encrypt the actual bulk data payload.

4. Steganography

(a) Steganography vs. Cryptography

- **Cryptography** is the practice of scrambling data so that it becomes unreadable to unauthorized entities. It aims to protect the *content* of a message. The presence of the communication is obvious, but its meaning is hidden.
- **Steganography** is the practice of concealing the *very existence* of a message by hiding it inside an innocuous cover medium (like an image, audio file, or text).

Analogy: Cryptography is putting a letter inside an indestructible, locked safe. Steganography is writing a letter using invisible ink on the back of a grocery list.

(b) Least Significant Bit (LSB) Technique

The LSB method is a common spatial-domain steganography technique used to hide secret text data inside a digital image.

Operational Principle

A standard uncompressed digital image consists of pixels, where each pixel's color is defined by byte channels (Red, Green, Blue). A single color channel byte ranges from 00000000 to 11111111 (0 to 255 in decimal).

- Changing the **Most Significant Bit (MSB)** drastically changes the value (e.g., altering 10000000 to 00000000 changes 128 to 0).
- Changing the **Least Significant Bit (LSB)** changes the value by at most 1 (e.g., altering 10000001 to 10000000 changes 129 to 128). This minute change is completely imperceptible to the human eye.

Steps

1. Convert the secret text message into a binary string (e.g., 'A' becomes 01000001).

2. Read the cover image pixel bytes sequentially.
3. Replace the last bit (LSB) of each image byte with a single bit from the secret message string.

[Image diagram showing how LSB image steganography substitutes the last bit of pixel RGB values with bits of a secret message without changing the visible image]

Limitations of Steganography

- **Size Overhead:** Cover media must be significantly larger than the secret message payload. Hiding a large file requires an enormous image or audio track.
- **Fragility:** If the stego-image undergoes compression (like converting a BMP to a JPEG), cropping, or resizing, the modified LSB bits are often scrubbed or destroyed, corrupting the hidden payload.
- **Detection (Steganalysis):** Specialized software can analyze statistical anomalies in the pixel bit structures to flag the presence of hidden data, breaking the core objective of the techniqu

5. Cryptanalytic & Security Attacks

1. Brute-Force Attacks

The attacker tries every single possible key combination in the key space until the ciphertext decrypts into intelligible plaintext.

- *Defense:* Secure algorithms are designed so that the computing time and energy costs required to exhaust the key range outweigh the value of the decrypted data.

2. Cryptanalytic Attacks (Based on Attacker Knowledge)

Cryptanalytic attacks vary based on how much information the adversary has access to when attempting to break a cipher:

A. Ciphertext-Only Attack

- **Attacker Knowledge:** The adversary possesses *only* a sample of intercepted ciphertext. They have no access to the corresponding plaintext or the underlying encryption key.
- **Method:** The attacker uses statistical analysis (like letter frequency counts in classical ciphers) or searches for mathematical patterns. This is the hardest type of attack to execute against modern ciphers.

B. Known-Plaintext Attack (KPA)

- **Attacker Knowledge:** The adversary has access to one or more strings of ciphertext alongside their matching, original plaintexts.
- **Method:** The attacker analyzes the transformation patterns between the known pairs to deduce the secret key. For example, knowing that an encrypted network packet always begins with a standard header (like GET / HTTP/1.1) provides a known plaintext-ciphertext pair.

C. Chosen-Plaintext Attack (CPA)

- **Attacker Knowledge:** The adversary can select arbitrary plaintext messages and obtain their corresponding encrypted ciphertexts from the target system.

- **Method:** The attacker deliberately submits specific, structurally crafted plaintexts designed to expose how the algorithm manages bit transitions, aiming to leak structural details of the key.

D. Chosen-Ciphertext Attack (CCA)

- **Attacker Knowledge:** The adversary can select arbitrary ciphertexts and obtain their decrypted plaintext forms from a decryption oracle, with the exception of the specific target ciphertext they want to crack.
- **Method:** Frequently used against asymmetric cryptosystems (like RSA). By monitoring how a system behaves or errors out when decrypting manipulated ciphertexts, the attacker can mathematically deduce the private key.