Introduction & Number Theory

Q. Explain the OSI Security Architecture with a neat diagram.

The **OSI Security Architecture** is a framework that defines **security services and mechanisms** applicable to different layers of the **OSI (Open Systems Interconnection)** model. It is defined by the **ITU-T X.800** recommendation.

The purpose of this architecture is to **provide a structured way to secure data communication** across networks by identifying:

- **Security attacks**
- **Security services**
- **Security mechanisms**

Components of OSI Security Architecture:

A. Security Services:

These are services that **protect network communication and data**. OSI defines the following security services:

Service	Description
Authentication	Confirms the identity of the sender/receiver.
Access Control	Prevents unauthorized access to resources.
Data Confidentiality	Ensures that data is not disclosed to unauthorized parties.
Data Integrity	Ensures data is not altered during transmission.
Non-repudiation	Prevents sender from denying a message was sent.

B. Security Mechanisms:

These are **tools or methods** used to implement security services.

Examples:

- **Encryption** (to provide confidentiality)
- **Digital signatures** (for integrity and non-repudiation)
- **Access control lists (ACLs)** (for access control)
- **Firewalls** and **intrusion detection systems (IDS)**

- **Authentication protocols** (like passwords, biometrics)

C. Security Attacks:

OSI architecture classifies attacks into two types:

- **Passive Attacks**: Eavesdropping or monitoring data (e.g., wiretapping, traffic analysis).
  Goal: Read data **without altering** it.
- **Active Attacks**: Modify, fabricate, or interrupt data (e.g., man-in-the-middle, replay attacks).
  Goal: **Change, inject, or disrupt** communication.

Q. Differentiate between passive and active attacks with examples.

In network security, an **attack** is any attempt to compromise the **confidentiality**, **integrity**, or **availability** of data. These attacks are broadly categorized into **Passive** and **Active** types based on their behavior.

Difference Between Passive and Active Attacks:

Criteria	Passive Attack	Active Attack
Definition	An attempt to **observe** or monitor data without altering it.	An attempt to **alter, destroy, or inject** data into a communication.
Goal	Gain **unauthorized access** to information.	Cause **harm or disruption** by modifying data or operations.
Data Integrity	Not affected; data remains unchanged.	Compromised; attacker modifies or corrupts data.
Detection	**Difficult to detect** (since nothing is changed).	Easier to detect (due to visible disruptions or changes).
Examples	- Eavesdropping - Traffic analysis - Packet sniffing	- Man-in-the-middle attack - Replay attack - Denial of Service (DoS) - Masquerade attack

Criteria	Passive Attack	Active Attack
Countermeasures	Encryption, secure protocols (HTTPS, SSH)	Authentication, integrity checks, firewalls, IDS/IPS

Q. What are the different types of security services and security mechanisms?

In network and system security, **security services** are designed to protect data and communication, while **security mechanisms** are the technical means or tools that help provide those services.

These are defined in the **OSI Security Architecture (ITU-T X.800)**.

Security Services (What we want to achieve)

Security services are objectives that ensure data **confidentiality**, **integrity**, **availability**, and **authenticity**.

Here are the main types:

Security Service	Purpose
Authentication	Confirms the identity of the sender/receiver.
Access Control	Prevents unauthorized users from accessing resources.
Data Confidentiality	Ensures that data is kept secret and not disclosed to unauthorized users.
Data Integrity	Ensures that data has not been modified, inserted, deleted, or replayed.
Non-repudiation	Prevents denial of message transmission (sender cannot deny sending the message).
Availability	Ensures that systems and data are accessible to authorized users when needed.

Security Mechanisms (How we achieve it)

Security mechanisms are the **tools, protocols, or techniques** used to enforce the above services.

Security Mechanism	Function
Encryption	Converts data into unreadable format to protect confidentiality (e.g., AES, RSA).
Digital Signature	Provides authentication, integrity, and non-repudiation.
Hash Functions	Used to verify integrity of data (e.g., SHA-256, MD5).
Authentication Protocols	Mechanisms like passwords, biometrics, OTP, challenge-response.
Access Control Mechanisms	Defines who can access what (e.g., ACLs, RBAC).
Firewalls and IDS/IPS	Protect against external attacks by filtering traffic.
Message Authentication Code (MAC)	Ensures integrity and origin of message using a secret key.

Q. Explain the network security model with a diagram.

A **Network Security Model** defines how two parties (sender and receiver) can **securely communicate** over an **insecure public network**, like the Internet. It shows how **encryption, keys, and trust** are used to protect the message from attackers.

Key Elements of the Network Security Model

The basic model involves:

Component	Function
Sender (A)	The user who wants to send a secure message.
Receiver (B)	The intended recipient of the message.
Message (M)	The original plain text data to be sent.
Encryption Algorithm (E)	Converts plain text to cipher text using a key.
Decryption Algorithm (D)	Converts cipher text back to plain text.
Key (K)	Secret or public key used in encryption and decryption.
Transmission Medium	Insecure public network (e.g., Internet).

Component	Function
Attacker	Tries to intercept, modify, or steal the message during transmission.

How It Works (Text-Based Flow)

1. **Sender A** wants to send a message **M** to **Receiver B**.
2. Sender uses an **encryption algorithm (E)** and a **key (K)** to produce ciphertext:
   C = E(K, M)
3. Ciphertext **C** is sent over the insecure network.
4. **Receiver B** uses the **decryption algorithm (D)** and the same or corresponding key to retrieve the original message:
   M = D(K, C)
5. If an **attacker** tries to intercept the message, they see only encrypted text (C), which is useless without the key.

Types of Security in the Model

- **Confidentiality**: Encryption keeps message hidden from attackers.
- **Integrity**: Hashing or MAC ensures the message wasn't modified.
- **Authentication**: Digital signatures prove the sender's identity.
- **Non-repudiation**: The sender can't deny sending the message.

Q. Explain the classical encryption techniques.

Classical encryption techniques are the earliest methods used to secure messages. These techniques operate at the **character level** (letter-by-letter) and mainly fall into two categories:

1. **Substitution techniques** – Replace each letter with another letter.
2. **Transposition techniques** – Rearrange the letters without changing them.

---

1. Substitution Techniques

In these methods, **each character** in the plaintext is **replaced** with another character using a predefined rule or key.

---

a. Caesar Cipher (Shift Cipher)

- Each letter is shifted by a fixed number.
- Example: With shift = 3, A → D, B → E, etc.
- Plaintext: HELLO → Ciphertext: KHOOR

---

b. Monoalphabetic Cipher

- Each letter maps to a **unique substitute letter**, but not just by shifting.
- Example: A → M, B → Q, C → R, etc. (random mapping)
- Easy to break using **frequency analysis**.

---

c. Playfair Cipher

- Uses a 5x5 matrix with a **keyword**.
- Encrypts **letter pairs (digraphs)** instead of single letters.
- Rules:
  - Same row: replace with letters to the right.
  - Same column: replace with letters below.
  - Rectangle: swap corners.

---

d. Hill Cipher

- Uses **linear algebra** with matrix multiplication.
- Letters are represented by numbers (A=0, B=1, ..., Z=25).
- Multiply vector of plaintext letters with a key matrix (mod 26).
- Example: For "HI", key matrix [[3, 3], [2, 5]], result → encrypted letters.

---

e. Vigenère Cipher

- Uses a **keyword** to repeat and shift letters.
- More secure than Caesar.
- Example:
  - Plaintext: ATTACK
  - Key: LEMON

- Ciphertext: LXFOPV

---

2. Transposition Techniques

Here, **letters are not changed**, but their **positions are shuffled** using a key.

---

a. Keyless Transposition

- Fixed pattern rearrangement.
- Example: Write in rows, read in columns.

---

b. Keyed Transposition

- Use a **keyword** to determine the order of columns.
- Example:
  - Keyword: ZEBRA → Order: 5 3 2 4 1
  - Rearranged columns based on key order for encryption.

---

3. Steganography (Bonus Classical Technique)

- **Hiding** the message inside **non-suspicious data** like images, audio, etc.
- Unlike encryption, the existence of the message itself is hidden.

Q. Explain the advantages and limitations of classical encryption techniques.

Classical encryption techniques are the early methods used for securing messages by either **substituting** characters or **rearranging** them. Examples include **Caesar cipher, Vigenère cipher, Playfair cipher, Hill cipher**, and **transposition ciphers**.

While these techniques laid the **foundation of cryptography**, they have both **strengths and weaknesses**.

Advantages of Classical Encryption Techniques

Advantage	Explanation
Simplicity	Easy to understand, implement, and use by hand—no need for complex computation.
Low Resource Usage	Ideal for manual or low-power devices since they don't require high processing power.
Educational Value	Helps students and learners understand the **basics of cryptographic concepts** like substitution, key usage, and frequency.
Fast Encryption	Classical ciphers are quick for small amounts of data, especially when done by hand or in simple systems.
Foundation for Modern Techniques	Many modern algorithms build upon or extend classical principles (e.g., substitution-permutation networks in AES).

Limitations of Classical Encryption Techniques

Limitation	Explanation
Weak Security	Easily broken using **frequency analysis**, brute-force, or known-plaintext attacks.
Small Key Space	Limited number of possible keys (e.g., Caesar cipher has only 25 keys), making it easy to guess.
No Resistance to Modern Attacks	Cannot withstand cryptanalysis techniques used in modern computing.
Lack of Key Management	Keys are often simple and reused, making systems more vulnerable.
Not Suitable for Digital Systems	They are not designed to handle large-scale, binary, or multimedia data used in modern communication.

Q. Compare symmetric and asymmetric encryption.

Encryption is a process of converting plain text into unreadable ciphertext to protect data. It is classified into two main types:

- **Symmetric Encryption**: Uses **one single key** for both encryption and decryption.
- **Asymmetric Encryption**: Uses a **pair of keys**—a public key for encryption and a private key for decryption.

Comparison Table: Symmetric vs Asymmetric Encryption

Feature	Symmetric Encryption	Asymmetric Encryption
Number of Keys	One key (same for encryption and decryption)	Two keys (public and private)
Key Sharing	Key must be **shared securely** between sender and receiver	No need to share the **private key**; only public key is shared
Speed	**Faster** – suitable for encrypting large amounts of data	**Slower** – due to complex mathematical operations
Algorithms	AES, DES, 3DES, RC5, Blowfish	RSA, ElGamal, ECC
Security	Secure, but if key is leaked, entire system is compromised	More secure for key distribution, but computationally expensive
Key Management	Difficult – requires a **secure channel** for key exchange	Easier – public keys can be openly distributed
Use Case	Bulk data encryption, VPNs, file systems	Digital signatures, email encryption, certificate-based communication

Q. Compare monoalphabetic and polyalphabetic substitution ciphers.

Both **monoalphabetic** and **polyalphabetic substitution ciphers** are types of **classical encryption** methods. They work by **replacing characters** in the plaintext with other characters, but differ in how the substitution is applied.

Comparison Table: Monoalphabetic vs Polyalphabetic Ciphers

Feature	Monoalphabetic Cipher	Polyalphabetic Cipher
Definition	Substitutes each letter with another letter using a **single fixed mapping**.	Substitutes letters using **multiple substitution alphabets**, which change throughout the message.
Key Usage	Uses **one key** (one alphabet mapping) for the entire message.	Uses a **repeating key** or keyword that changes the mapping as the message progresses.

Feature	Monoalphabetic Cipher	Polyalphabetic Cipher
Example Ciphers	Caesar Cipher, Simple Substitution	Vigenère Cipher, Autokey Cipher
Security	**Weaker** – easily broken using **frequency analysis**.	**Stronger** – frequency analysis is much harder due to changing patterns.
Pattern	Same plaintext letter always maps to the **same cipher letter**.	Same plaintext letter can map to **different cipher letters** at different positions.
Complexity	Simple to implement and break.	More complex and harder to break manually.

Examples

- **Monoalphabetic** (Caesar Cipher, shift 3):
  HELLO → KHOOR (H→K, E→H...)
- **Polyalphabetic** (Vigenère with key "KEY"):
  HELLO → RIJVS
  (each letter encrypted using a different Caesar shift based on key)

Q. Explain Playfair cipher with encryption steps.

The **Playfair cipher** is a **polyalphabetic substitution cipher** that encrypts **pairs of letters (digraphs)** instead of single letters. It was invented by **Charles Wheatstone** in 1854 and later promoted by **Lord Playfair**.

It uses a **5×5 matrix** of letters constructed from a **keyword** to encrypt messages.

Key Rules of Playfair Cipher

Before encryption:

- **Alphabet "J" is merged with "I"**, so the 26-letter alphabet fits in a 5x5 grid.
- Repeating letters in a pair are **separated by inserting X** (e.g., "BALLOON" → "BA LX LO ON").

Encryption Steps

Let’s encrypt the message:
Plaintext: THE KEY IS HIDDEN UNDER THE DOOR
Keyword: DOMESTIC

Step 1: Prepare the 5×5 Matrix

Create the key square using the keyword **DOMESTIC** (no repeating letters), then fill in remaining letters of the alphabet (excluding "J"):

D O M E S

T I C A B

F G H K L

N P Q R U

V W X Y Z

Step 2: Prepare the Plaintext

- Remove spaces: THEKEYISHIDDENUNDERTHEDOOR
- Split into digraphs (pairs):
  TH EK EY IS HI DD EN UN DE RT HE DO OR
- Replace double letters (DD → DX):
  TH EK EY IS HI DX EN UN DE RT HE DO OR

Step 3: Encrypt Each Pair Using Rules

Rule 1: Same row → replace with letter to the right
Rule 2: Same column → replace with letter below
Rule 3: Rectangle → swap columns

Examples:

- TH: T and H → Row 2 and Row 3 → rectangle
  → T → C, H → G ⇒ **CG**
- EK: E and K → Row 1 and Row 3 → rectangle
  → E → S, K → L ⇒ **SL**
- EY: E and Y → Row 1 and Row 4 → rectangle
  → E → S, Y → Z ⇒ **SZ**
- IS: Same row (Row 2): I → C, S → T ⇒ **CT**

Continue this for all digraphs.

Q. Explain Playfair cipher with an example. *(Most frequent classical cipher in PYQs)*

The **Playfair cipher** is a type of **polyalphabetic substitution cipher** that encrypts **two letters (digraphs)** at a time instead of single letters. It uses a **5×5 matrix** of letters constructed using a **keyword**, making it **more secure** than monoalphabetic ciphers like Caesar cipher.

Key Concepts of Playfair Cipher

- The **keyword** is used to build a 5×5 matrix (only 25 letters; "J" is combined with "I").
- Plaintext is split into **pairs**. If both letters in a pair are the same (e.g., "LL"), insert **'X'** between them.
- If there is an **odd number of letters**, add an **'X'** at the end.
- **Rules to Encrypt Each Pair:**
    1. If both letters are in the **same row**, replace each with the letter to its **right**.
    2. If both are in the **same column**, replace each with the letter **below**.
    3. If in **different rows and columns**, replace each with the **letter in the same row but the column of the other letter**.

Example

Let's encrypt:
Plaintext: HELLO
Keyword: MONARCHY

Step 1: Build 5×5 Matrix

Remove duplicates from the keyword and fill in remaining letters of the alphabet (excluding J):

M O N A R

C H Y B D

E F G I K

L P Q S T

U V W X Z

Step 2: Prepare the Plaintext

- Original: HELLO
- Split into pairs: HE, LX, LO
  (We inserted X between repeating Ls)

Step 3: Encrypt Digraphs

Pair 1: HE

- H is at (2nd row, 2nd col), E is at (3rd row, 1st col) → different rows and columns
- H → C (same row as H, column of E)
- E → F (same row as E, column of H)
  → Encrypted pair: **CF**

Pair 2: LX

- L → P (same row, right of L)
- X → Z (same row, right of X)
  → Encrypted pair: **PZ**

Pair 3: LO

- L is (4th row, 1st col), O is (1st row, 2nd col) → rectangle rule
- L → P, O → M
  → Encrypted pair: **PM**

Final Ciphertext:

HELLO → CFPZPM

Q. Explain Vigenère cipher with example.

The **Vigenère cipher** is a **polyalphabetic substitution cipher** that uses a **repeating keyword** to determine the shift of each letter in the plaintext. Unlike Caesar cipher, which uses a single shift value, Vigenère uses **multiple Caesar shifts** based on letters of the keyword.

This makes it **more secure** than monoalphabetic ciphers, as the same letter in plaintext can be encrypted to **different letters** in the ciphertext.

How Vigenère Cipher Works

Components:

- **Plaintext**: The original message to encrypt.
- **Keyword**: A word used to generate a series of shifts.
- **Ciphertext**: The encrypted message.

Encryption Rule:

Each letter of the plaintext is shifted forward by a number of positions corresponding to the alphabetical position of the matching keyword letter.

Formula:
Cipher letter = (Plain letter + Key letter) mod 26

Example

Let’s encrypt:
Plaintext: ATTACKATDAWN
Keyword: LEMON

Step 1: Repeat the keyword to match plaintext length

Plaintext: A T T A C K A T D A W N

Keyword: L E M O N L E M O N L E

Step 2: Convert letters to positions (A = 0, B = 1, ..., Z = 25)

Letter	**A**	**T**	**T**	**A**	**C**	**K**	**A**	**T**	**D**	**A**	**W**	**N**
Value	0	19	19	0	2	10	0	19	3	0	22	13

| Key | L | E | M | O | N | L | E | M | O | N | L | E | | Value |11 | 4 |12 |14 |13 |11 | 4 |12 |14 |13 |11 | 4 |

Step 3: Add and apply modulo 26

Step	**0+11**	**19+4**	**19+12**	**0+14**	**2+13**	**10+11**	**0+4**	**19+12**	**3+14**	**0+13**	**22+11**	**13+4**
Cipher Values	11	23	5	14	15	21	4	5	17	13	7	17
Ciphertext	**L**	**X**	**F**	**O**	**P**	**V**	**E**	**F**	**R**	**N**	**H**	**R**

Final Ciphertext:

ATTACKATDAWN → LXFOPVEFRNHR

Decryption

To decrypt, subtract key letter value from cipher letter value:

Plaintext = (Cipher – Key) mod 26

(You can mention this briefly in exams, but focus more on encryption if question only asks for example.)

Q. Describe Hill cipher with an example.

The **Hill cipher** is a **polyalphabetic substitution cipher** that uses **linear algebra** and **matrix multiplication** to encrypt blocks of plaintext. It was invented by **Lester S. Hill** in 1929 and is one of the first ciphers based on **mathematics and matrix operations**.

Unlike other ciphers that operate on letters individually or in pairs, Hill cipher encrypts **multiple letters at once** using a **key matrix**.

Key Concepts

- Each letter is represented by a number:
  A = 0, B = 1, C = 2, ..., Z = 25
- A **key matrix** of size $n \times n$ is used to encrypt plaintext in blocks of n letters.
- All operations are done **mod 26** (since there are 26 letters in English).

Encryption Formula

Let:

- **P** = Plaintext vector
- **K** = Key matrix
- **C** = Ciphertext vector

Then:

$C = (K \times P) \bmod 26$

Example

Let's encrypt:
Plaintext: HI
Key matrix (2×2):

K = | 3 3 |

| 2 5 |

Step 1: Convert Plaintext to Numbers

Plaintext = HI
H = 7, I = 8 → P =

| 7 |

| 8 |

Step 2: Matrix Multiplication

Multiply K × P:

| 3 3 | | 7 | = | (3×7 + 3×8) | = | 21 + 24 | = | 45 |

| 2 5 | × | 8 | | (2×7 + 5×8) | = | 14 + 40 | = | 54 |

Step 3: Apply mod 26

45 mod 26 = 19 → T

54 mod 26 = 2 → C

Final Ciphertext: TC

Q. Explain transposition techniques – keyed and keyless.

Transposition techniques are classical encryption methods in which the **positions of the letters are rearranged** without changing the actual letters. Unlike substitution ciphers, which replace characters, **transposition ciphers just shuffle them**.

There are two main types:

- **Keyless Transposition Cipher**
- **Keyed Transposition Cipher**

---

1. Keyless Transposition Cipher

This method follows a **fixed rule** or pattern to rearrange characters, without using any key.

Example: Simple Columnar Transposition

Plaintext: HELLO WORLD
(Remove spaces: HELLOWORLD)
Write in rows of fixed length, say 4:

H E L L

O W O R

L D X X ← Pad with 'X' to complete the grid

Now, read column by column:

Ciphertext: HOLH EWDL LOXR → HOLHEWDLLOXR
(*Write in actual columns for clarity in your answer*)

---

2. Keyed Transposition Cipher

This method uses a **keyword** to decide the order in which columns are read. The keyword determines the **column order** by sorting its letters alphabetically.

Example:

Keyword: ZEBRA
Plaintext: ATTACKATDAWN
(Remove spaces, no punctuation)

Step 1: Assign numbers to keyword by alphabetical order:

Z E B R A → 5 3 2 4 1

Step 2: Write message in rows under keyword:

Z E B R A

5 3 2 4 1

A T T A C

K A T D A

W N X X X ← Pad with X

Step 3: Read columns in numeric order (1 → 5):

1st (A): C A X

2nd (B): T T X

3rd (E): T A N

4th (R): A D X

5th (Z): A K W

Final Ciphertext (column-wise):

CAXTTXTANADXAKW

Q. Define and differentiate between keyed and keyless transposition ciphers.

A **transposition cipher** is a classical encryption technique in which the **positions of characters are rearranged**, but the actual characters remain unchanged.

It does **not substitute** letters, but **shuffles** them based on a rule or a key.

Transposition ciphers are of two main types:

- **Keyless Transposition Cipher**
- **Keyed Transposition Cipher**

Difference Between Keyed and Keyless Transposition Ciphers

Aspect	Keyless Transposition Cipher	Keyed Transposition Cipher
Definition	A cipher that rearranges characters using a **fixed, pre-decided pattern** without any external key.	A cipher that rearranges characters using a **keyword** to determine the column order.
Use of Key	No key is used.	A **key (keyword)** is required to define the encryption order.
Complexity	Simpler to implement and understand.	More complex and secure due to varying column orders.
Example	Write characters in rows and read column-wise (fixed sequence).	Assign numbers to keyword letters, write message in grid, then read columns in keyword order.
Security	Lower; easier to break using pattern recognition.	Higher; depends on secrecy and complexity of the key.

Q. Compare transposition ciphers vs substitution ciphers.

Aspect	Substitution Cipher	Transposition Cipher
Definition	Replaces each character in the plaintext with a different symbol or letter.	Rearranges the positions of characters in the plaintext.
Characters	Characters are **changed** (e.g., A → D).	Characters are **not changed**, only their **order** is shuffled.
Letter Frequency	Alters the frequency distribution of letters.	**Preserves** the frequency of letters.
Example	Caesar cipher, Vigenère cipher, Playfair cipher.	Columnar transposition, rail fence cipher, keyed transposition.
Encryption Logic	Depends on **substitution rules** or key alphabets.	Depends on a **reordering rule or key**.
Cryptanalysis Resistance	Vulnerable to **frequency analysis**, especially monoalphabetic ones.	Vulnerable to **pattern recognition** and **anagram solving**.
Strength	Can be stronger if using polyalphabetic or modern techniques.	Can be made stronger by combining with substitution.

Q. Compare block ciphers vs stream ciphers.

In modern cryptography, encryption algorithms are broadly categorized into **block ciphers** and **stream ciphers** based on **how data is processed** during encryption.

- **Block cipher**: Encrypts data in **fixed-size blocks** (e.g., 64 or 128 bits).
- **Stream cipher**: Encrypts data **bit by bit or byte by byte**, like a continuous stream.

Comparison Table: Block Cipher vs Stream Cipher

Aspect	Block Cipher	Stream Cipher
Data Processing	Encrypts **blocks of data** (e.g., 64 or 128 bits at a time).	Encrypts **one bit or byte** at a time.
Speed	**Slower** due to block-wise processing.	**Faster**, especially for real-time applications.
Error Propagation	One error can **corrupt entire block**.	A single error affects only **one bit/byte**.

Aspect	Block Cipher	Stream Cipher
Security	Generally more secure; supports modes of operation like CBC, CTR.	Vulnerable to **key reuse attacks** if not used carefully.
Padding Required	Yes, for short messages (must fill full block).	No padding needed.
Complexity	Higher – uses modes of operation to handle long messages.	Simpler implementation.
Examples	AES, DES, Blowfish, RC5	RC4, A5/1 (used in GSM), Salsa20
Use Cases	File encryption, HTTPS, email encryption.	Streaming audio/video, wireless encryption, VoIP.

Q. What is steganography? How does it differ from encryption?

Steganography is the technique of **hiding secret information** within a **non-secret, ordinary-looking file**, such as an image, audio, video, or text file, in such a way that **no one can detect** that a secret message exists.

- The word comes from Greek: *steganos* (hidden) + *graphia* (writing).
- Unlike encryption, which hides the **content**, steganography hides the **existence** of the message.

How Steganography Works

- **Message**: The secret data to be hidden (text, image, file).
- **Cover medium**: The carrier file (image, audio, video, etc.).
- **Stego-object**: The final file that looks normal but contains hidden data.

For example:

- Hiding a text message inside the **least significant bits (LSBs)** of an image file's pixels.
- Hiding data in **inaudible parts of an audio file**.

Difference Between Steganography and Encryption

Aspect	Steganography	Encryption
Purpose	Hides the **existence** of the message.	Hides the **content** of the message.
Visibility	Message is invisible to casual observers.	Encrypted data is visible but unreadable.
Detection	Hard to detect unless steganalysis is applied.	Easy to detect that data is encrypted (but hard to read).
Security	Secrecy relies on the **cover medium and method**.	Secrecy relies on **strong encryption algorithms and keys**.
Common Use	Covert communication, watermarking, digital rights management.	Secure communication, data protection, authentication.
Tools	Steghide, OpenStego, SilentEye	AES, RSA, DES, GPG

Q. Compare steganography vs cryptography.

Both **steganography** and **cryptography** are techniques used to **protect information**, but they differ in their **approach and goals**.

- **Cryptography** focuses on making the message **unreadable** to unauthorized users.
- **Steganography** focuses on **hiding the existence** of the message.

Comparison Table: Steganography vs Cryptography

Aspect	Steganography	Cryptography
Goal	Hides the **existence** of the message.	Hides the **content** of the message.
Visibility	Message is **invisible**, embedded in cover media (e.g., image, audio).	Encrypted message is **visible** but appears as gibberish.
Security Approach	Security through **obscurity** (nobody should notice).	Security through **mathematics** and **keys**.
Detection	Hard to detect unless specifically analyzed.	Easy to detect that encryption is used (ciphertext is obvious).
Common Media Used	Images, videos, audio, text (as cover).	Plaintext, binary files, and any data.

Aspect	Steganography	Cryptography
Risk if Discovered	Once found, hidden data may be easily extracted.	Without the key, even if discovered, data remains secure.
Tools/Examples	Steghide, OpenStego, image LSB hiding.	AES, RSA, DES, SHA, GPG.
Use Cases	Covert messaging, watermarking, DRM.	Secure communication, authentication, digital signatures.

Q.