

**CNS Important Topics**

**1.Cryptographic attacks**.

**Cryptographic Attacks**

Cryptographic attacks are methods used to undermine the security of cryptographic systems.

Here are key types of attacks:

1.**Brute Force Attack**:

o **Description**: Attempts every possible key until the correct one is found.

o **Countermeasure**: Use longer key lengths.

2.**Cryptanalysis**:

o **Description**: Analyzes algorithms to find vulnerabilities and decrypt data without the

key.

o **Example**: Techniques like linear and differential cryptanalysis.

3.**Man-in-the-Middle (MitM) Attack**:

o **Description**: An attacker intercepts communication between two parties.

o **Countermeasure**: Implement strong encryption and authentication (e.g., SSL/TLS).

4.**Replay Attack**:

o **Description**: An attacker captures and reuses valid data transmissions.

o **Countermeasure**: Use nonces or timestamps to ensure message uniqueness.

5.**Side-Channel Attack**:

o **Description**: Exploits physical characteristics (e.g., timing, power usage) of a

cryptographic system.

o **Countermeasure**: Use constant-time algorithms and other physical protections.

6.**Chosen Plaintext Attack**:

o **Description**: The attacker can choose plaintexts to be encrypted and obtain the

corresponding ciphertexts.

o **Countermeasure**: Use strong encryption schemes resistant to such attacks.

7.**Chosen Ciphertext Attack**:

o **Description**: The attacker can choose ciphertexts to decrypt and learn information

about the key or plaintext.

o **Countermeasure**: Implement secure encryption methods.

8.**Birthday Attack**:

o **Description**: Exploits the probability of hash collisions in hash functions.

o **Countermeasure**: Use hash functions with larger output sizes (e.g., SHA-256).



**2.Security Services & Mechanisms.**

AN). **Security Services & Mechanisms (5 Marks)**

**Security services and mechanisms** are key to protecting data and ensuring secure communication in information systems. Here's a concise overview:

**Security Services:**

1.**Confidentiality**:   
oEnsures that sensitive data is only accessible to authorized users.

o**Mechanism**: Encryption (e.g., AES, RSA).

2.**Integrity**:   
oGuarantees data has not been altered or tampered with.

o**Mechanism**: Hash functions (e.g., SHA-256).

3.**Authentication**:   
oVerifies the identity of users before granting access.

o**Mechanism**: Passwords, biometrics, multi-factor authentication.

4.**Non-repudiation**:   
oEnsures that neither the sender nor the recipient can deny the action.

o**Mechanism**: Digital signatures, transaction logs.

5.**Access Control**:   
oRegulates who can access resources and what actions they can perform. o**Mechanism**: Role-based access control (RBAC), access control lists (ACLs).

**Security Mechanisms:**

1.**Encryption**:   
oConverts data into unreadable formats to prevent unauthorized access.

2.**Firewalls**:   
oMonitors and controls network traffic based on security rules.

3.**IDS (Intrusion Detection System)**:   
oDetects suspicious activities in a network and alerts administrators.

4.**VPN (Virtual Private Network)**:   
oCreates secure encrypted connections over the internet.

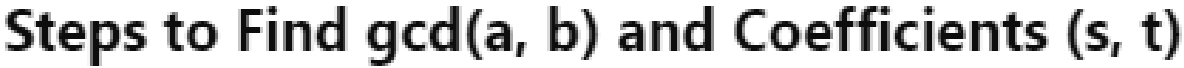
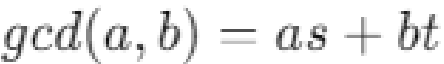
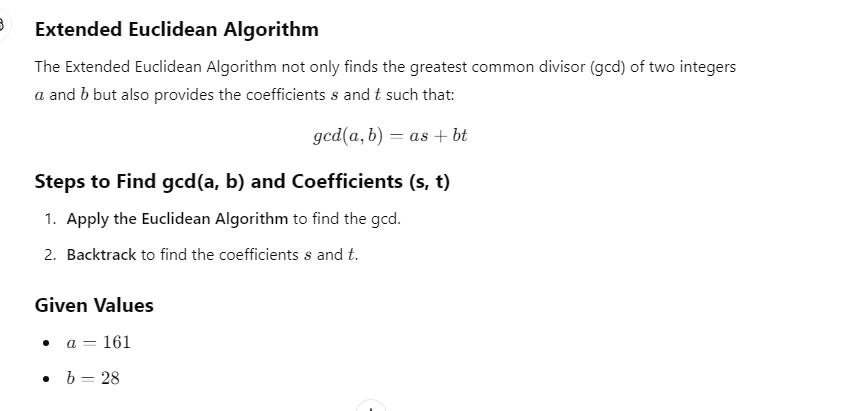
5.**Antivirus Software**:   
oDetects and removes malware from devices.

**Diagram of Security Services & Mechanisms**

+---------------------+

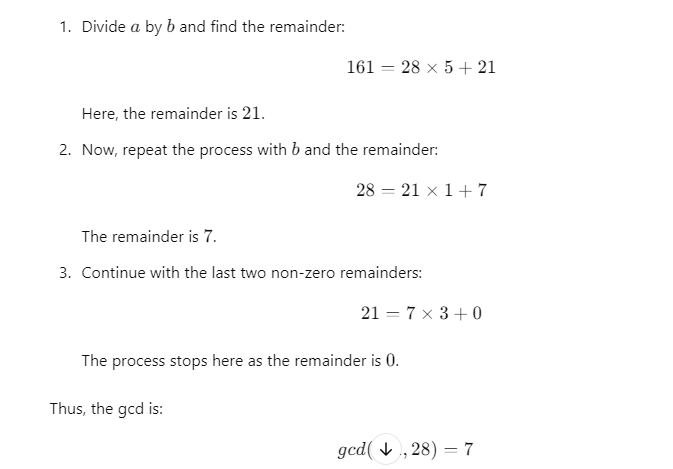
| Security Services|

+-------------------

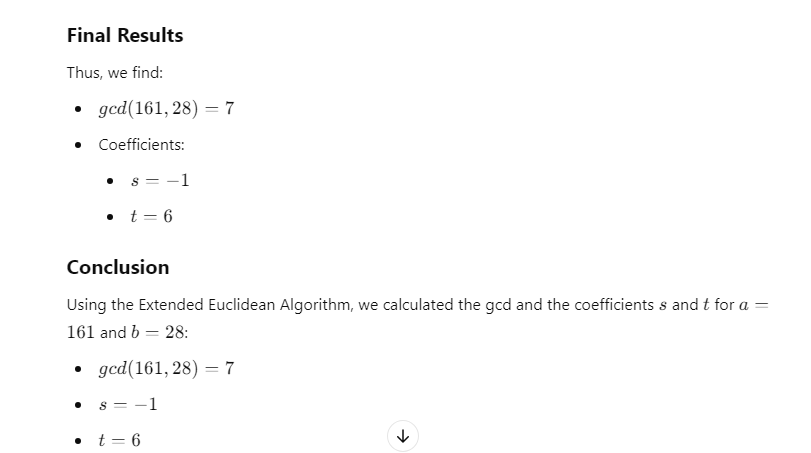
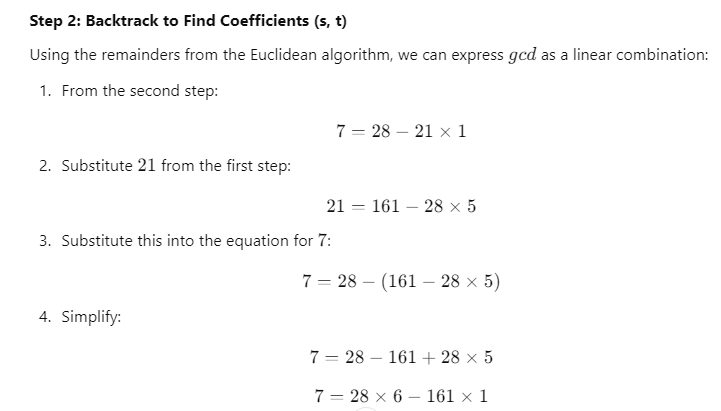


**3.Explain Extended Euclidean Algorithm. Given a=161 and b=28, find gcd(a,b) and the values of (s) and (t) using the Extended Euclidean Algorithm.**

AN).

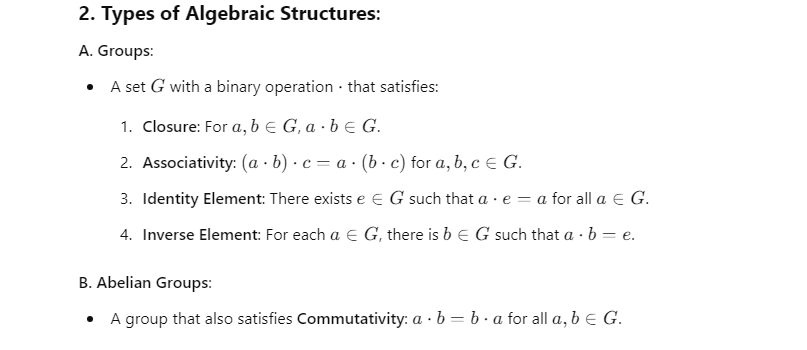


**Example: Find gcd(161, 28) and Coefficients (s, t) *Step 1: Calculate gcd Using the Euclidean Algorithm***



**4.Algebraic structures properties.**

AN).  **Algebraic Structures and Their Properties**



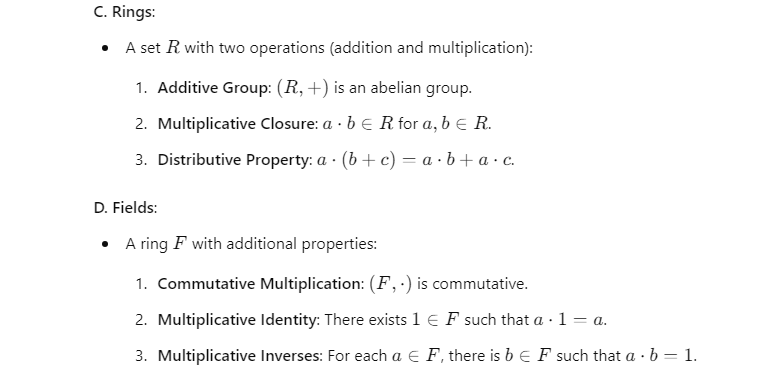
Algebraic structures are mathematical entities consisting of a set equipped with one or more operations that satisfy specific properties. Here are the fundamental types of algebraic structures and their key properties:

**1. Sets and Operations**

A set is a collection of distinct objects, and an operation is a rule that combines elements of the set.







**Structure**  **Properties**



**Structure**  **Properties**

**Group** Closure, Associativity, Identity, Inverses

**Abelian Group** All group properties + Commutativity

**Ring** Additive Group, Multiplicative Closure, Distribution

**Field** All ring properties + Commutative Multiplication, Identity, Inverses

**5.DES Permutations and Round Function.**

AN). **DES Permutations and Round Function (Short Version)**

**1. Key Permutations in DES:**

|  |  |
| --- | --- |
|    | **Initial Permutation (IP)**: Rearranges the 64-bit input before encryption.  **Inverse Initial Permutation (IP−1^{-1}−1)**: Reverses the IP at the end of encryption. **Permutation (P)**: Reorders 32-bit output from the S-boxes within each round to spread bit influence. |

**2. DES Round Function (Steps):**

1.**Expansion (E)**: Expands the 32-bit right half to 48 bits.

2.**XOR with Round Key**: The expanded right half is XORed with the 48-bit round key. 3.**Substitution (S-boxes)**: The 48-bit result is passed through 8 S-boxes, reducing it to 32 bits.

4.**Permutation (P)**: The 32-bit output is permuted to mix bits.

5.**XOR with Left Half (L)**: The permuted output is XORed with the left half of the input. 6.**Swapping**: The left and right halves are swapped for the next round.

**Diagram (Simplified)**

L(i-1) R(i-1)

| |

|--> XOR with Key ---> Expansion (E)

| |

| Substitution (S-boxes)

| Permutation (P)

|<-- XOR with L(i-1) ----|



**Conclusion**: DES uses multiple permutations, substitutions, and XOR operations within each

round to ensure strong encryption by mixing and diffusing bits effectively.

**6.AES Structure**

AN). **AES Structure (Advanced Encryption Standard)**

AES is a symmetric key encryption standard used for secure data encryption. It operates on 128-

bit blocks of data and supports key sizes of 128, 192, or 256 bits. AES consists of multiple

rounds of transformations, depending on the key size.

**Key Features of AES Structure**

**Block Size**: 128 bits (fixed)

**Key Sizes**: 128 bits (10 rounds), 192 bits (12 rounds), 256 bits (14 rounds)

**Number of Rounds**: Depends on the key size:

o 10 rounds for 128-bit key

o 12 rounds for 192-bit key

o 14 rounds for 256-bit key

**AES Transformation Steps**

Each AES round (except the final one) consists of four main transformations:

***1. SubBytes (Substitution Layer)***

Each byte in the block is replaced using a substitution table (S-box).

Provides non-linearity to AES and improves security.

***2. ShiftRows (Row Shifting)***

Each row in the 4x4 state matrix is shifted left by a certain number of positions.

Row 0: No shift

Row 1: Shift left by 1 byte

Row 2: Shift left by 2 bytes

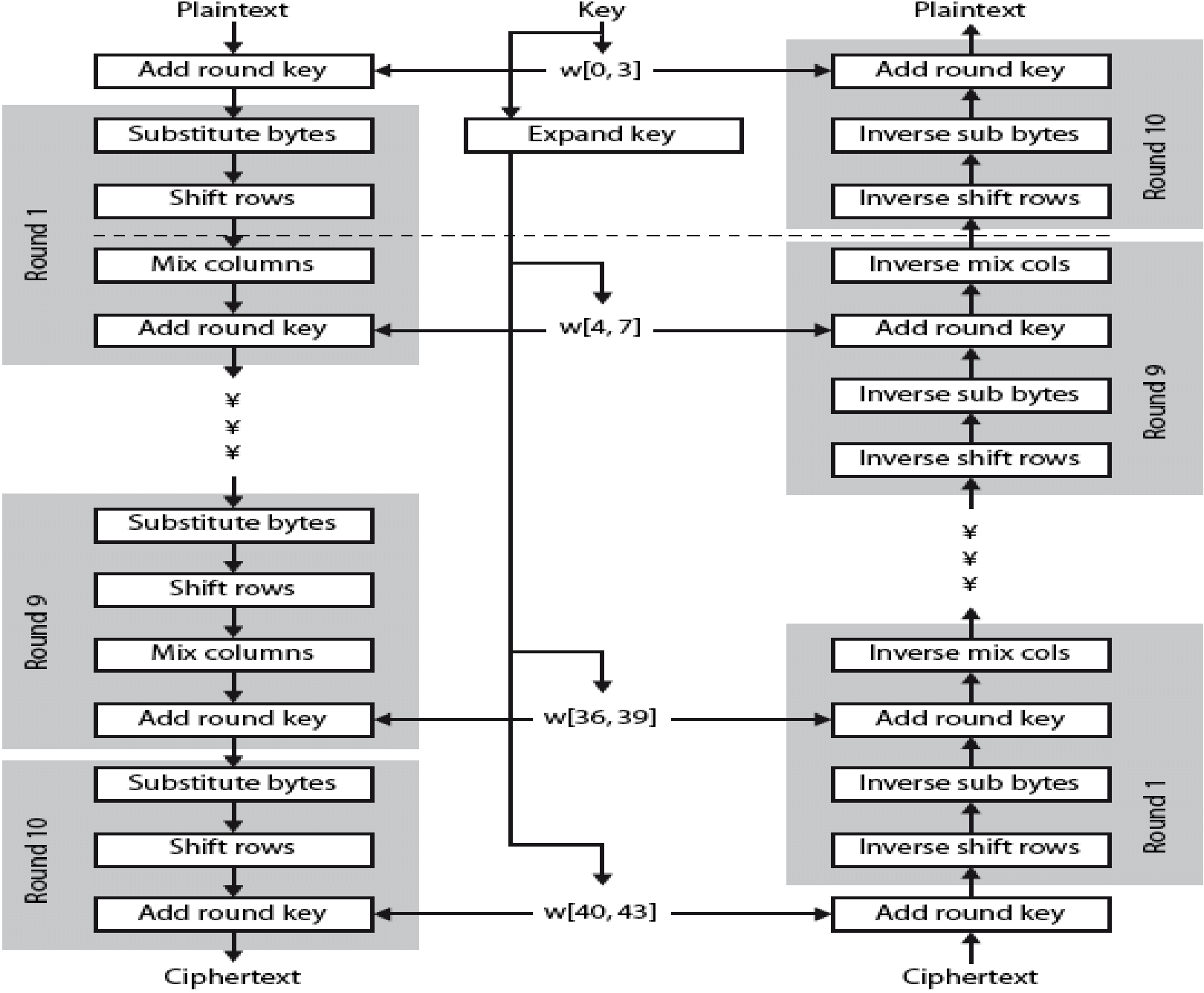
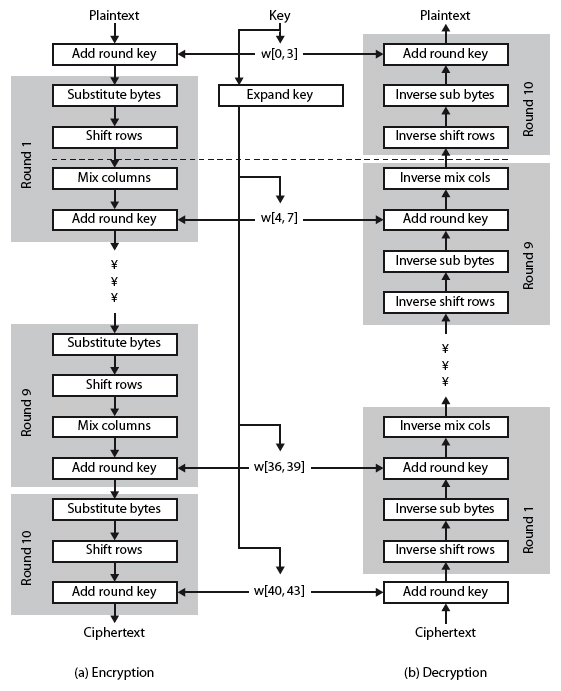
Row 3: Shift left by 3 bytes

This step introduces diffusion by rearranging the bytes.

***3. MixColumns (Column Mixing)***

Each column in the state matrix is transformed by multiplying it with a fixed polynomial in

GF(2^8).



This step ensures that each byte influences the other bytes in its column, spreading out the data

across columns.

***4. AddRoundKey (Key Addition)***

The current state is XORed with a round key derived from the original key using the **Key**

**Expansion** process.

This is the only step where the encryption key is introduced into the round.

**Initial and Final Round**

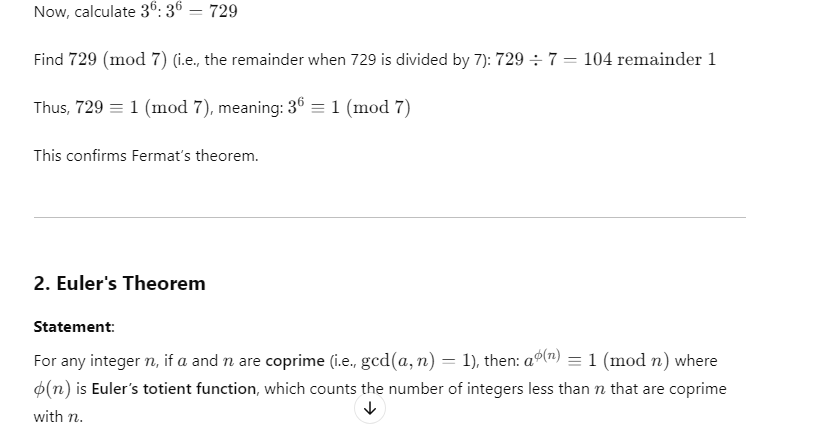
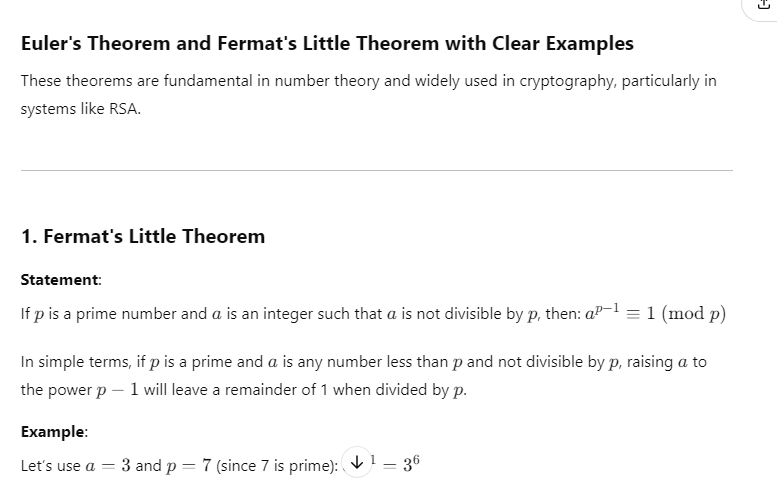
**Initial Round**: The input block is XORed with the first round key (before the first regular round).

**Final Round**: The final round excludes the **MixColumns** step and consists of only **SubBytes**,

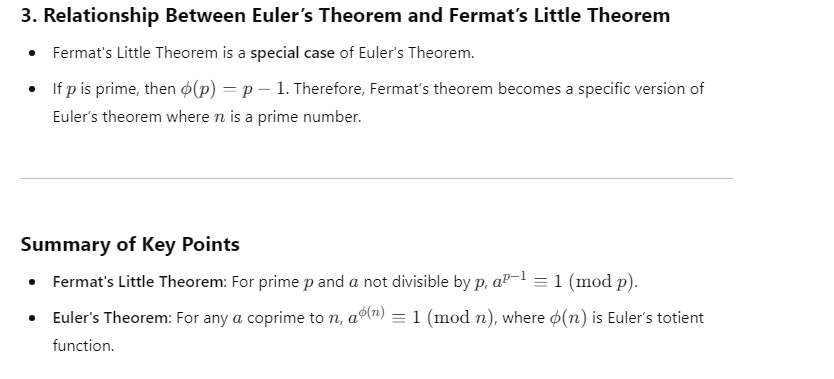
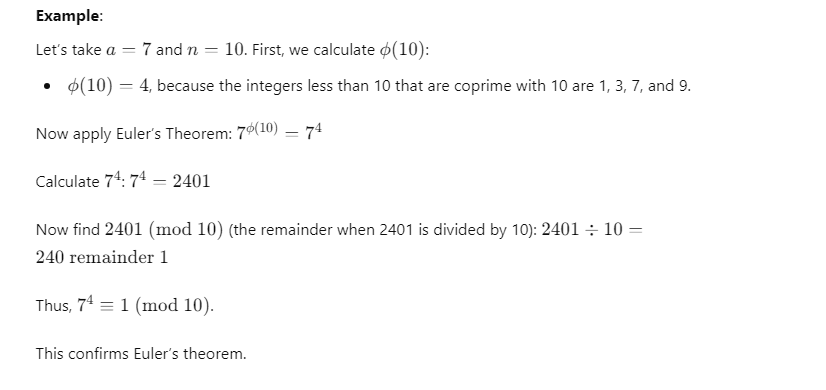
**ShiftRows**, and **AddRoundKey**.

**AES Round Structure (Diagram)**

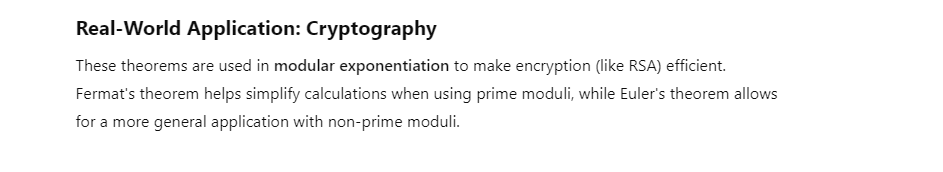
**7.Euler's theorem and Fermat's theorem with examples.**

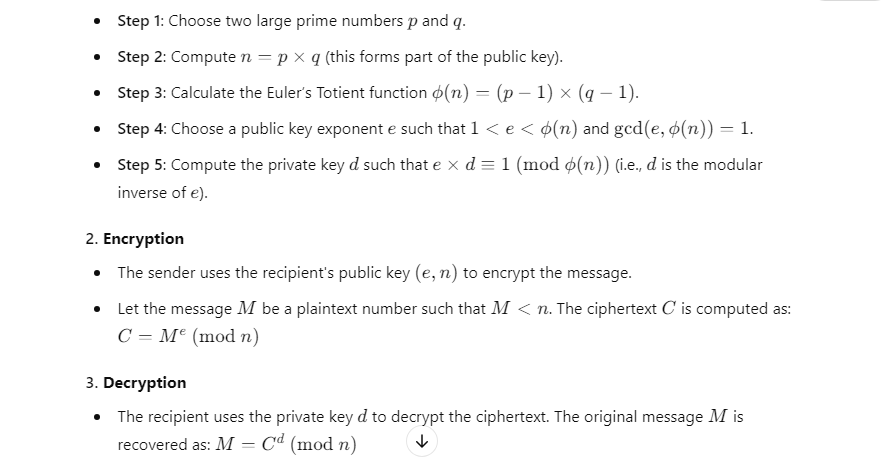


AN)







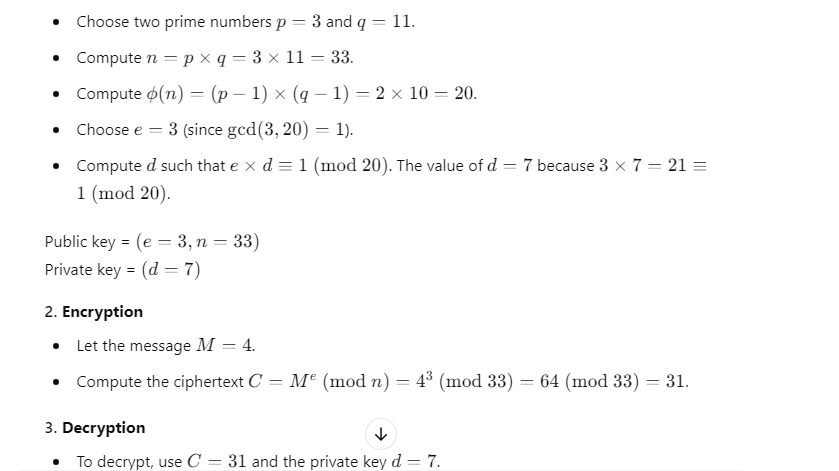
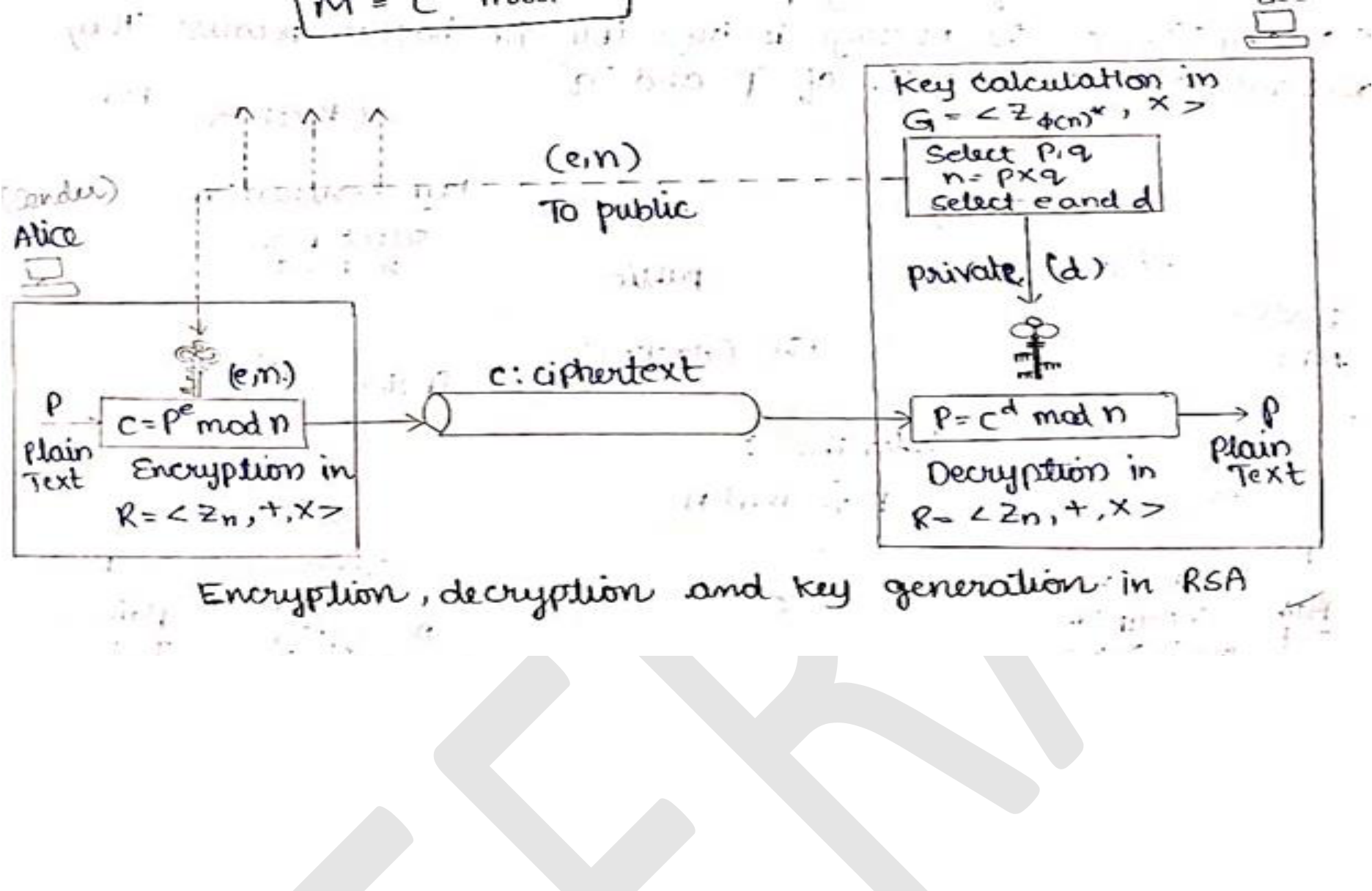
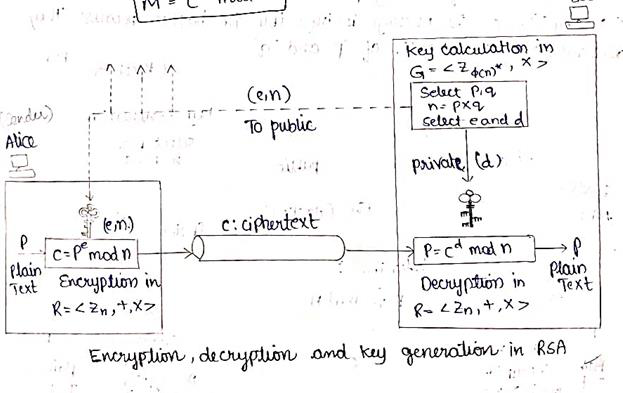


**8.RSA Algorithm with a neat diagram.**

AN). **RSA Algorithm**   
The **RSA Algorithm** is a widely-used public-key cryptosystem for secure data transmission. It is based on the mathematical difficulty of factoring large prime numbers.

**Steps in RSA Algorithm**

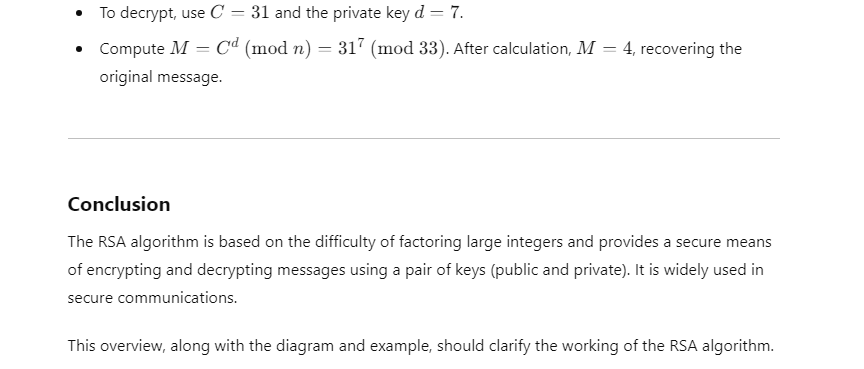
|  |  |
| --- | --- |
| **Key Generation:** |  |



**Diagram of RSA Algorithm**

**Example of RSA Algorithm**

***1. Key Generation***

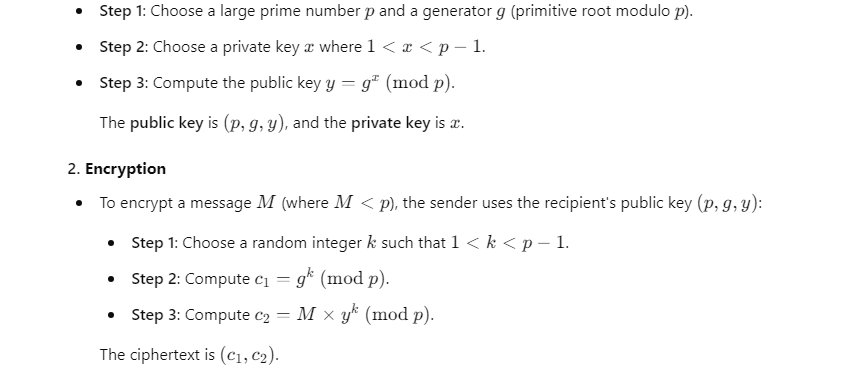


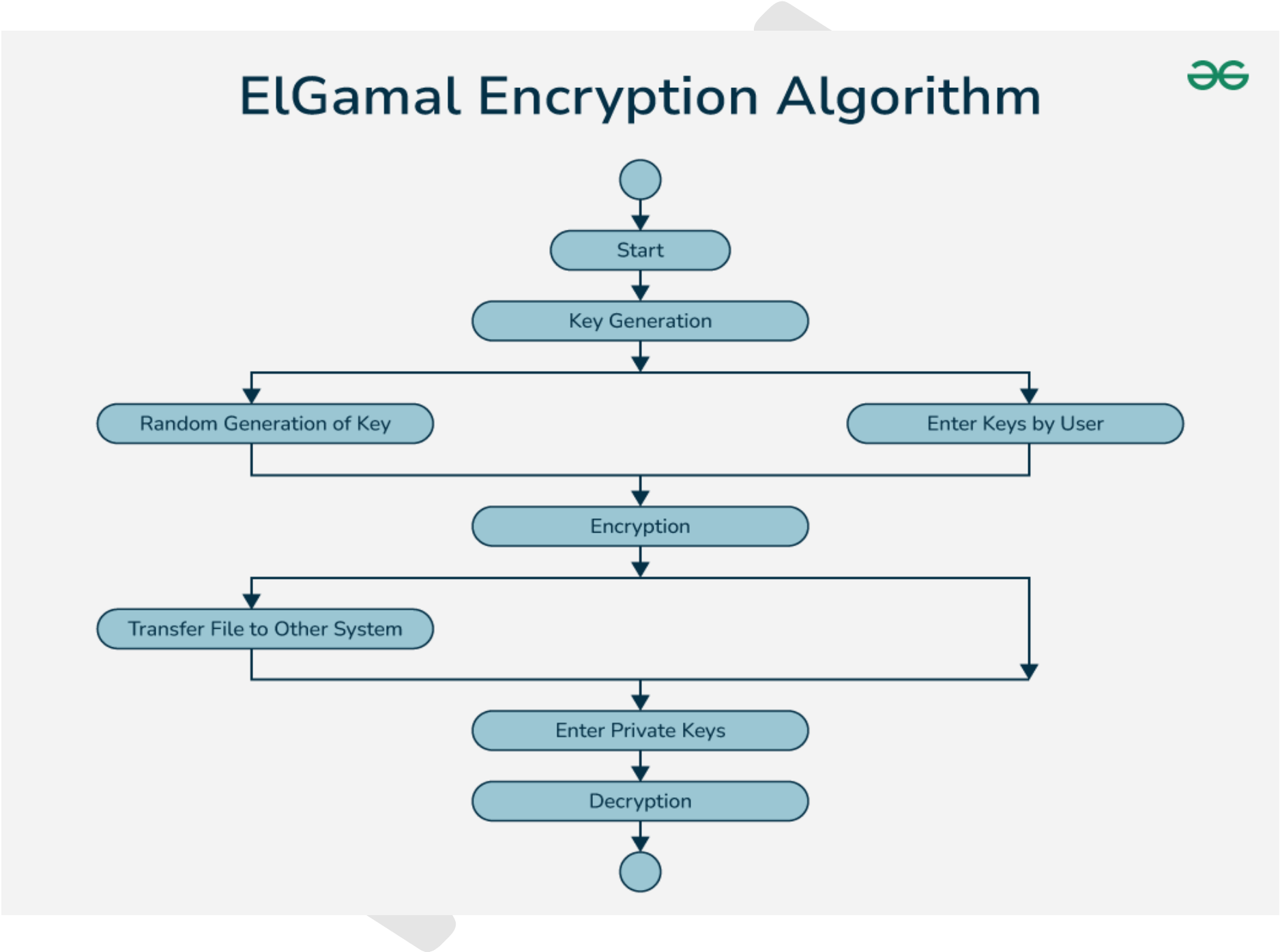
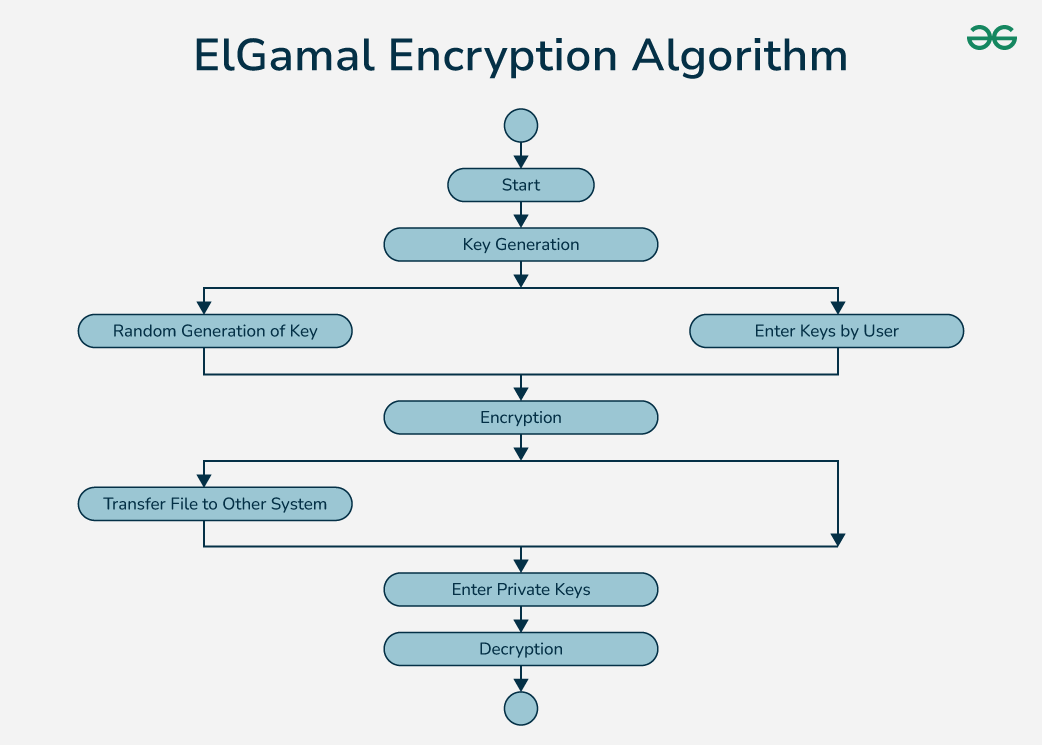
**9.ElGamal Cryptosystem.**

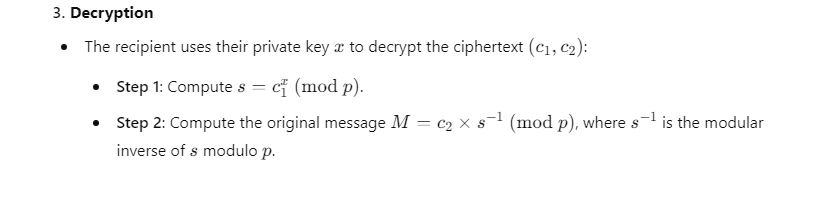
**AN). ElGamal Cryptosystem**   
The **ElGamal Cryptosystem** is a public-key encryption algorithm based on the Diffie-Hellman key exchange. It provides both encryption and digital signatures and is widely used in   
cryptographic applications.

**Steps in the ElGamal Cryptosystem**   
***1. Key Generation***









**Diagram of ElGamal Cryptosystem**

**Key Points**

|  |  |
| --- | --- |
|    | The ElGamal cryptosystem is based on the difficulty of solving the **Discrete Logarithm Problem**.  It is used in cryptographic protocols, including digital signatures and key exchanges. The encryption and decryption process relies on the modular arithmetic of large prime  numbers and generators. |

This example simplifies the cryptosystem to show the key concepts without large numbers that are typically used in real-world implementations.



**10.Applications of cryptographic hash functions.**

AN). **Applications of Cryptographic Hash Functions (Short Version)**

1.**Data Integrity Verification**: Detects alterations in data by comparing hashes.

2.**Digital Signatures**: Ensures message authenticity by signing the hash with a private key. 3.**Password Storage**: Stores hashed passwords instead of plaintext for enhanced security.

4.**Blockchain and Cryptocurrencies**: Secures data blocks, making tampering evident.

5.**Message Authentication Codes (MACs)**: Verifies integrity and authenticity of messages.

6.**Certificate Authorities**: Ensures the integrity of digital certificates.

7.**File Integrity Checking**: Creates checksums for verifying file integrity.

8.**Random Number Generation**: Enhances randomness in secure random number generators.

9.**Deduplication**: Identifies duplicate files using unique hashes.

10.**Secure Multi-Party Computation**: Facilitates secure computations among multiple parties.

These applications highlight the critical role of cryptographic hash functions in ensuring security

and integrity in various systems.