Check point 2 :

Basic consept :

a. Definitions:

Plaintext: The original, readable, and unencrypted data or message.

Ciphertext: The result of applying encryption to the plaintext, it is the unreadable and encrypted form of the data or message.

Encryption: The process of converting plaintext into ciphertext using an algorithm and a key.

Decryption: The process of converting ciphertext back into plaintext, using a key to reverse the encryption.

b. Difference between symmetric and asymmetric cryptography:

Symmetric Cryptography: In symmetric key cryptography, the same key is used for both encryption and decryption. Both the sender and the receiver share a common secret key. It is efficient in terms of computational speed, but the challenge is securely distributing and managing the shared key.

Asymmetric Cryptography: In asymmetric key cryptography, a pair of public and private keys is used. The public key is shared openly, while the private key is kept secret. Data encrypted with the public key can only be decrypted with the corresponding private key, and vice versa. Asymmetric cryptography provides a solution to the key distribution problem in symmetric cryptography but is generally slower computationally.

c. Three common goals of cryptography:

Confidentiality: Ensuring that unauthorized individuals cannot access or understand the information. This is achieved through encryption, where only those with the proper key can decrypt and access the original data.

Integrity: Verifying that the information has not been altered or tampered with during transmission or storage. Cryptographic hash functions and digital signatures are commonly used to ensure data integrity.

Authentication: Verifying the identity of the parties involved in a communication. Cryptographic techniques such as digital signatures and certificates are used to establish and confirm the identities of entities in a secure manner.

Symmetric Key Cryptography:

a. Encrypt the plaintext "HELLO" using a Caesar cipher with a shift of 3:

Plaintext: HELLO. Caesar cipher with a shift of 3:

H + 3 = K

E + 3 = H

L + 3 = O

L + 3 = O

O + 3 = R

The encrypted ciphertext is "KHOOR."

b. Decrypt the ciphertext "VWDQGD" using a Caesar cipher with a shift of 3:

Ciphertext: VWDQGD. Caesar cipher with a backward shift of 3:

V - 3 = S

W - 3 = T

D - 3 = A

Q - 3 = N

G - 3 = D

The decrypted plaintext is "STAND."

c. Encrypt the plaintext "OPENAI" using a Vigenère cipher with the keyword "CRYPTO":

Plaintext: OPENAI. Keyword: CRYPTO (repeated to match the length of the plaintext)

Vigenère cipher:

O + C = P

P + R = S

E + Y = I

N + P = O

A + T = G

I + O = P

The encrypted ciphertext is "PSIOGP."

Asymmetric Key Cryptography:

a. Generate a key pair using RSA encryption with a prime modulus of 17 and a public exponent of 5:

For RSA key generation, we typically choose two distinct prime numbers, p and q. In this case:

- p = 17 (prime)
- q = 17 (prime)
- n = p * q = 17 * 17 = 289
- φ(n) = (p-1) * (q-1) = 16 * 16 = 256

Now, choose a public exponent (e) such that 1 < e < φ(n) and e is coprime with φ(n). Here, e = 5 is a valid choice.

The public key is (e, n) => (5, 289).

To find the private exponent (d), calculate d ≡ e^(-1) mod φ(n). In this case, d = 77 is a valid choice.

The private key is (d, n) => (77, 289).

b. Encrypt the plaintext "SECRET" using the recipient's public key: (e=5, n=221):

- Convert each character to its ASCII value:
    - S: 83, E: 69, C: 67, R: 82, E: 69, T: 84
- Encrypt each ASCII value individually using the public key: ciphertext = (plaintext^e) mod n
    - S: 83^5 mod 221 = 196
    - E: 69^5 mod 221 = 41
    - C: 67^5 mod 221 = 4
    - R: 82^5 mod 221 = 160
    - E: 69^5 mod 221 = 41
    - T: 84^5 mod 221 = 130

The encrypted ciphertext is "196 41 4 160 41 130."

c. Decrypt the ciphertext "196" using the recipient's private key: (d=53, n=221):

- Decrypt using the private key: plaintext = (ciphertext^d) mod n
    - 196^53 mod 221 = 83

The decrypted plaintext is "83," which corresponds to the ASCII value of the character 'S'.

Cryptographic Algorithms:

a. Comparison of DES and AES:

DES (Data Encryption Standard):

- Strengths:
    - Well-established and widely used in the past.
    - Fast and efficient in software implementations.
- Weaknesses:
    - Small key size (56 bits) makes it susceptible to brute force attacks.
    - Vulnerable to advances in cryptanalysis, and modern computing power can break it within a reasonable time.

AES (Advanced Encryption Standard):

- Strengths:
    - Strong and widely accepted security.
    - Supports key sizes of 128, 192, and 256 bits, providing a high level of security.
    - Efficient and fast in both hardware and software implementations.
- Weaknesses:
    - No significant weaknesses have been found when used properly.

b. Hash Function:

- Concept: A hash function takes an input (or 'message') and returns a fixed-size string of bytes, which is typically a hash value. It is designed to be fast and deterministic, producing a unique hash for different inputs. Hash functions are commonly used for data integrity verification, password storage, and digital signatures.

Example: SHA-256 (Secure Hash Algorithm 256-bit):

- SHA-256 takes an input and produces a 256-bit hash value.
- It is widely used for various cryptographic applications and is considered secure.
- Example: The SHA-256 hash of the message "Hello, World!" is "a591a6d40bf420404a011733cfb7b190d62c65bf0bcda32b57b277d9ad9f146e. "

c. Advantages of Hybrid Cryptosystem:

- Efficiency: Symmetric encryption is faster than asymmetric encryption. Using symmetric encryption for data encryption and asymmetric encryption for key exchange combines efficiency with security.
- Key Management: Symmetric encryption requires a shared key, which can be securely exchanged using asymmetric encryption. This avoids the challenges of securely sharing symmetric keys.
- Scalability: Asymmetric encryption is computationally expensive, especially for large amounts of data. A hybrid approach allows for a balance between security and performance.
- Forward Secrecy: Even if an attacker compromises the symmetric key used for data encryption, the compromise doesn't affect the security of past communications as new keys are generated for each session.

In summary, a hybrid cryptosystem leverages the strengths of both symmetric and asymmetric encryption to provide a secure and efficient solution for various cryptographic applications.

Practical Application:

c. Digital Signature:

A digital signature is a cryptographic technique that provides authenticity, integrity, and non-repudiation for a message or document. Here's a brief explanation of the concept:

- **Creation:** The sender uses their private key to create a unique digital signature for the message. This signature is appended to the message.
- **Verification:** The recipient uses the sender's public key to verify the signature. If the signature is valid, it confirms that the message was signed by the private key corresponding to the public key, and the message hasn't been altered.
- **Role in Authenticity:** Digital signatures provide a way to verify that a message comes from a specific sender and hasn't been tampered with during transmission. They ensure the authenticity of the sender and the integrity of the message.
- **Non-Repudiation:** With a valid digital signature, the sender cannot later deny sending the message since only their private key could have produced the signature.