Checkpoint Cryptography & Secure solution

Basic Concept:

a. **Define the terms plaintext, ciphertext, encryption, and decryption:**

- *Plaintext:* It refers to the original, readable, and unencrypted text or data before any cryptographic operation.
- *Ciphertext:* It is the result of applying encryption to the plaintext, making it unreadable without the correct decryption key.
- *Encryption:* The process of transforming plaintext into ciphertext using an algorithm and a key.
- *Decryption:* The process of transforming ciphertext back into plaintext using the appropriate decryption key and algorithm.

b. **Explain the difference between symmetric and asymmetric cryptography:**

- *Symmetric Cryptography:* Involves the use of a single key for both encryption and decryption. The communicating parties share this secret key.
- *Asymmetric Cryptography:* Uses a pair of keys - a public key for encryption and a private key for decryption. The public key can be shared openly, while the private key is kept secret.

c. **List three common goals of cryptography:**

- *Confidentiality:* Ensuring that unauthorized parties cannot understand the encrypted information.
- *Integrity:* Verifying that the information has not been altered during transmission or storage.
- *Authentication:* Confirming the identity of the communicating parties.

Symmetric Key Cryptography:

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

- Plain: HELLO
- Cipher: KHOOR

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

- Cipher: VWDQGD
- Plain: TESTING

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

- Plain: OPENAI
- Keyword: CRYPTO
- Cipher: RIJVSU

Asymmetric Key Cryptography:

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

- Private Key: (d, n) = (13, 17)
- Public Key: (e, n) = (5, 17)

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

- Plain: SECRET
- Cipher: 204 99 76 187 76 76

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

- Cipher: 196
- Plain: 72

Cryptographic Algorithms:

a. **Research and compare the strengths and weaknesses of DES and AES encryption algorithms:**

- **DES (Data Encryption Standard):**
  - *Strengths:* Pioneer in symmetric key encryption. Widely used in the past.
  - *Weaknesses:* Small key size (56 bits) makes it vulnerable to brute force attacks. Considered insecure for modern applications.
- **AES (Advanced Encryption Standard):**
  - *Strengths:* Strong and widely adopted. Three key sizes (128, 192, 256 bits). Efficient and resistant to various attacks.
  - *Weaknesses:* Vulnerabilities are generally related to implementation rather than the algorithm itself.

b. **Explain the concept of a hash function and provide an example of a commonly used hash function:**

- **Hash Function Concept:** A hash function takes an input (or 'message') and produces a fixed-size string of characters, which is typically a hash value or digest. It is a one-way function, meaning it should be computationally infeasible to reverse the process and obtain the original input.
- **Example:** SHA-256 (Secure Hash Algorithm 256-bit) is a commonly used hash function. It produces a 256-bit hash value, typically represented as a 64-character hexadecimal number.

c. **Discuss the advantages of using a hybrid cryptosystem that combines symmetric and asymmetric encryption:**

- **Advantages:**
  - *Efficiency:* Symmetric encryption is faster than asymmetric encryption. Using symmetric encryption for large amounts of data is more efficient.
  - *Key Management:* Asymmetric encryption handles key distribution and management more securely since the private keys are not shared.
  - *Security and Speed:* Combining the strengths of both encryption types provides a good balance of security and speed.

Practical Application:

a. **Design a simple encryption program in Python using a symmetric key:**

```
def encrypt(plaintext, key):
    ciphertext = ""
    for char in plaintext:
        ciphertext += chr((ord(char) + key) % 128)
    return ciphertext

def decrypt(ciphertext, key):
    plaintext = ""
    for char in ciphertext:
        plaintext += chr((ord(char) - key) % 128)
    return plaintext
```

b. **Implement a function that calculates the MD5 hash of a given input string:**

```
import hashlib

def calculate_md5(input_string):
    md5_hash = hashlib.md5(input_string.encode()).hexdigest()
    return md5_hash
```

c. **Research and explain the concept of a digital signature and its role in verifying message authenticity:**

- **Digital Signature Concept:** A digital signature is a cryptographic technique that provides authentication and integrity to a message or document. It involves the use of a private key to sign the message, and the recipient can verify the signature using the sender's public key.
- **Role in Verifying Authenticity:** A digital signature ensures that the message has been signed by the private key of the sender, and only the corresponding public key can verify this signature. It confirms the identity of the sender and guarantees that the message has not been tampered with during transmission. Digital signatures play a crucial role in ensuring the authenticity and integrity of digital communications.