Cryptography Basics Assignment

1. Introduction to Cryptography

Cryptography is the practice of securing information through encoding methods to ensure confidentiality, integrity, and authenticity. The key concepts covered in this assignment are encryption, hashing, and digital signatures.

2. Types of Encryption

Symmetric Encryption (AES-256)

- Uses a single key for both encryption and decryption.
- Example: Advanced Encryption Standard (AES-256) is widely used for secure data transmission.

Asymmetric Encryption (RSA)

- Uses a pair of keys: Public Key (encryption) and Private Key (decryption).
  - Example: RSA (Rivest-Shamir-Adleman) is commonly used for secure key exchanges.

3. Hashing Algorithms

MD5 (Message Digest Algorithm 5)

- Produces a 128-bit hash value.
- Considered weak due to vulnerabilities to collision attacks.

SHA-256 (Secure Hash Algorithm 256-bit)

- Produces a 256-bit hash value.
  - Stronger than MD5, widely used in blockchain and digital signatures.

4. Tasks

Task 1: AES-256 Encryption & Decryption using OpenSSL

Step 1: Create a sample text file:

```
echo "This is a secret message." > plaintext.txt
```

Step 2: Encrypt the file using AES-256:

```
openssl enc -aes-256-cbc -salt -in plaintext.txt -out encrypted.bin -pass pass:yourpassword
```

Step 3: Verify encrypted file:

```
hexdump -C encrypted.bin | head
```

Step 4: Decrypt the file:

```
openssl enc -aes-256-cbc -d -in encrypted.bin -out decrypted.txt -pass pass:yourpassword
```

Step 5: Compare original and decrypted files:

```
diff plaintext.txt decrypted.txt
```

Task 2: RSA Key Pair Generation using OpenSSL

Step 1: Generate a private key:

openssl genpkey -algorithm RSA -out private_key.pem -aes256

Step 2: Extract the public key:

openssl rsa -in private_key.pem -pubout -out public_key.pem

Step 3: Check key details:

openssl rsa -in private_key.pem -text -noout

Task 3: Verify Hash Integrity of a File using SHA-256

Step 1: Generate SHA-256 hash:

openssl dgst -sha256 plaintext.txt

Step 2: Save hash output to a file:

openssl dgst -sha256 plaintext.txt > hash.txt

Step 3: Verify hash integrity by recomputing and comparing:

openssl dgst -sha256 -c plaintext.txt

5. Conclusion

This assignment covered fundamental cryptography concepts, including AES-256 encryption, RSA key pair generation, and SHA-256 hashing. These methods are crucial in ensuring secure communication and data integrity.

Appendix: OpenSSL Installation

If OpenSSL is not installed, use the following command to install it:

sudo apt install openssl # For Linux

brew install openssl # For macOS