Integrantes Do Grupo

Otavio Paiva/ RA: 824147017
Gabriel Prieto/ RA: 824142064
Eduardo Baptistella/ RA: 824147595

1. Criptografia Simétrica de Chaves (AES)

A AES (Standard Avançado de Criptografia) é uma forma de criptografia simétrica amplamente empregada. No exemplo a seguir, empregamos a biblioteca pycryptodome para realizar a criptografia e a descriptografia de dados através do algoritmo AES.

```
pip install pycryptodome

from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
from Crypto.Random import get_random_bytes


key = get_random_bytes(16)  # Para AES-128
cipher = AES.new(key, AES.MODE_CBC)


data = b'Criptografia simétrica com AES'


ciphertext = cipher.encrypt(pad(data, AES.block_size))
print("Texto Criptografado:", ciphertext)


decipher = AES.new(key, AES.MODE_CBC, cipher.iv)
plaintext = unpad(decipher.decrypt(ciphertext), AES.block_size)
print("Texto Descriptografado:", plaintext.decode('utf-8'))
```

2. Criptografia com Chaves Assimétricas (RSA)

O algoritmo RSA (Rivest-Shamir-Adleman) é amplamente utilizado para a criptografia de chave pública. A seguir, realizamos a codificação e decodificação utilizando RSA com a biblioteca pycryptodome.

```
pip install pycryptodome

from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
from Crypto.Random import get_random_bytes


key = RSA.generate(2048)
private_key = key
public_key = key.publickey()


cipher_rsa = PKCS1_OAEP.new(public_key)
decipher_rsa = PKCS1_OAEP.new(private_key)


data = b'Criptografia assimétrica com RSA'


ciphertext = cipher_rsa.encrypt(data)
print("Texto Criptografado:", ciphertext)


plaintext = decipher_rsa.decrypt(ciphertext)
print("Texto Descriptografado:", plaintext.decode('utf-8'))
```

Função Hash (SHA-256)

As funções de hash são empregadas para produzir um valor específico (resumo) a partir de um dado de dimensão variável. A função hash SHA-256 é frequentemente empregada.

```
pip install pycryptodome
```

```
from Crypto.Hash import SHA256

# Dados a serem "hasheados"
data = b'Função hash SHA-256'

hash_object = SHA256.new(data)

hex_digest = hash_object.hexdigest()
print("Hash SHA-256:", hex_digest)
```