UNIVERSIDADE SÃO JUDAS TADEU

Emilly dos santos ferreira
RA: 825153657

Sistemas computacionais e segurança

Professor Robson Calvetti

Sao Paulo - 2025

• Atividade: Implementação de Algoritmos de Criptografia

1. Criptografia com Chaves Simétricas (Exemplo: AES)
2. Criptografia com Chaves Assimétricas (Exemplo: RSA)
3. Função Hash (Exemplo: SHA-256)

Optei pela lingugem de programação: Python

Criptografia com Chaves Simétricas (AES - Advanced Encryption Standard)

📌 Código Implementado (Python)

Usamos a biblioteca cryptography e o algoritmo Fernet para garantir segurança.

Instalação da Biblioteca

Antes de rodar o código, instale a biblioteca necessária:
pip install cryptography

```
from cryptography.fernet import Fernet

# Gerar chave secreta para criptografia e descriptografia
chave = Fernet.generate_key()
cipher = Fernet(chave)

# Mensagem que será protegida
mensagem = "Texto Secreto"

# Criptografar a mensagem
mensagem_criptografada = cipher.encrypt(mensagem.encode())

# Descriptografar a mensagem
mensagem_descriptografada = cipher.decrypt(mensagem_criptografada).decode()

# Exibir resultados
print("🔹 Mensagem original:", mensagem)
print("🔹 Texto criptografado:", mensagem_criptografada)
print("🔹 Texto descriptografado:", mensagem_descriptografada)
```

Criptografia com Chaves Assimétricas (RSA - Rivest-Shamir-Adleman)

Código Implementado (Python)

Usamos a biblioteca PyCryptodome.

Instalação da Biblioteca

Antes de rodar o código, instale a biblioteca necessária:
pip install pycryptodome

```
from Crypto.PublicKey import RSA
from Crypto.Cipher import PKCS1_OAEP
import base64

# Gerar par de chaves RSA (privada e pública)
chave_privada = RSA.generate(2048)
chave_publica = chave_privada.publickey()

# Criar um objeto de criptografia com a chave pública
cipher_rsa = PKCS1_OAEP.new(chave_publica)

# Mensagem a ser criptografada
mensagem = "Segredo RSA".encode()

# Criptografar a mensagem com a chave pública
mensagem_criptografada = cipher_rsa.encrypt(mensagem)
mensagem_criptografada_b64 = base64.b64encode(mensagem_criptografada)

print("🔹 Texto criptografado (RSA):", mensagem_criptografada_b64)

# Criar objeto de descriptografia com a chave privada
cipher_rsa_dec = PKCS1_OAEP.new(chave_privada)

# Descriptografar a mensagem
mensagem_descriptografada =
cipher_rsa_dec.decrypt(base64.b64decode(mensagem_criptografada_b64))

print("🔹 Texto descriptografado:", mensagem_descriptografada.decode())
```

Função Hash (SHA-256 - Secure Hash Algorithm 256 bits)

Código Implementado (Python)

Usamos a biblioteca nativa hashlib.

```
import hashlib

# Texto a ser transformado em hash
mensagem = "SenhaSuperSegura"

# Criar um hash SHA-256 da mensagem
hash_sha256 = hashlib.sha256(mensagem.encode()).hexdigest()

# Exibir o hash gerado
print("🔹 Hash SHA-256:", hash_sha256)
```