Shenzhen University

Report of The Experiments

Course：Information Security and Blockchain

Topic：Hash Function and AES

Class：Wenhua Honor Class

StudentID：2022280142

Name：崔殷霖

Date：2024.9.10

Score：

1. Experiment Content

Choose one of the following 2 tasks, and complete it in any program language and with any tool.

a. Please program to implement SHA-256 and AES, and be able to demonstrate on the command line to a string, calculate its hash value as well as encrypt and decrypt it.

b. Refer to the online code for SHA-256 and AES implementations, and write a certain interface (non-DOS interface) to them respectively, to be able to finish calculating the Hash value for a file (e.g. 1.txt), and to be able to encrypt and decrypt the file.

2. The experimental code and results screenshots

(1) Code of functions

```
from flask import Flask, request, jsonify, render_template
import hashlib
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad
import base64
import os

app = Flask(__name__, static_folder='static')



def parse_ascii_to_utf8(input_string):
    # 将输入的 ASCII 码字符串转换为字节序列
    byte_sequence = bytes.fromhex(input_string.replace(" ", ""))
    # 将字节序列解码为 UTF-8 编码的字符串
    utf8_string = byte_sequence.decode('utf-8')
    return utf8_string



def sha256_hash(input_string):
    # 将 ASCII 码格式的输入文本转换为 UTF-8 编码的字符串
    utf8_string = parse_ascii_to_utf8(input_string)
    # 创建一个新的 SHA-256 哈希对象
    sha256 = hashlib.sha256()
    # 将 UTF-8 编码的字符串进行更新
    sha256.update(utf8_string.encode('utf-8'))
    # 返回哈希值的十六进制字符串表示
    return sha256.hexdigest()



def aes_encrypt(input_string, key):
    # 使用 SHA-256 从密钥生成一个 256 位的密钥
    key = hashlib.sha256(key.encode()).digest()
    # 生成一个随机的初始化向量 (IV)
    iv = os.urandom(16)
    # 创建一个新的 AES 密码实例
    cipher = AES.new(key, AES.MODE_CBC, iv)
    # 加密 UTF-8 编码的字符串，并填充至块大小
    encrypted_bytes = cipher.encrypt(pad(input_string.encode('utf-8'), AES.block_size))
    # 对加密的数据（包括 IV）进行 Base64 编码，并转换为 UTF-8 字符串
    encrypted_string = base64.b64encode(iv + encrypted_bytes).decode('utf-8')
    return encrypted_string


def aes_decrypt(encrypted_string, key):
    # 使用 SHA-256 从密钥生成一个 256 位的密钥
    key = hashlib.sha256(key.encode()).digest()
    # 对加密字符串进行 Base64 解码
    encrypted_data = base64.b64decode(encrypted_string)
    # 提取初始化向量 (IV)
    iv = encrypted_data[:16]
    # 提取加密后的数据
    encrypted_bytes = encrypted_data[16:]
    # 创建一个新的 AES 密码实例
    cipher = AES.new(key, AES.MODE_CBC, iv)
    # 解密数据并去除填充
    decrypted_string = unpad(cipher.decrypt(encrypted_bytes), AES.block_size).decode('utf-8')
    return decrypted_string


@app.route('/')
def index():
    # 渲染首页
    return render_template('index.html')


@app.route('/sha256', methods=['POST'])
def handle_sha256():
    # 获取请求中的输入字符串
    input_string = request.json.get('input_string')
    # 计算 SHA-256 哈希值
    hash_value = sha256_hash(input_string)
    # 返回 JSON 格式的哈希值
    return jsonify({"hash_value": hash_value})
```

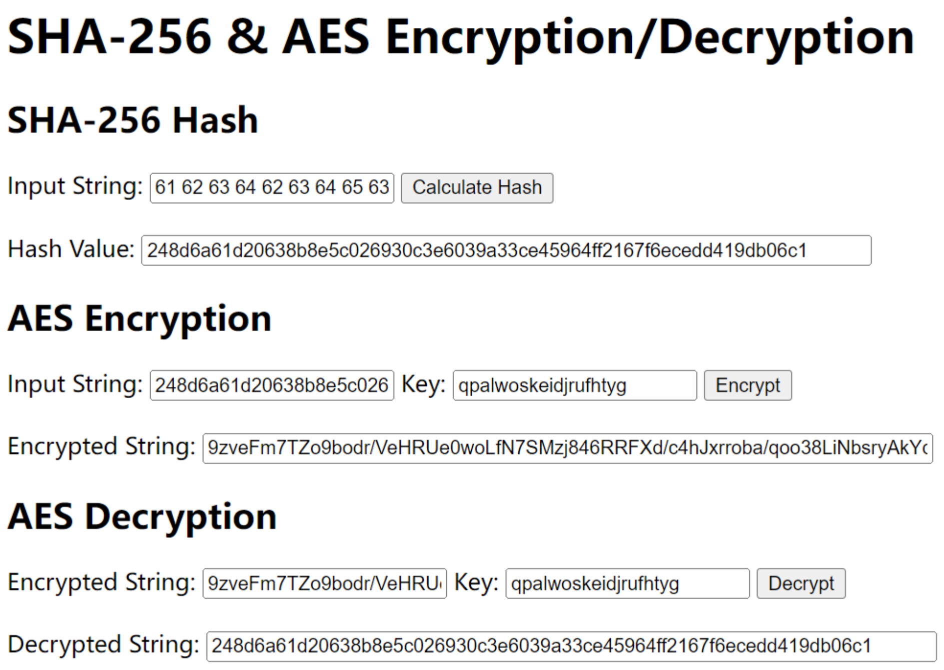
```
@app.route('/encrypt', methods=['POST'])
def handle_encrypt():
    # 获取请求中的输入字符串和密钥
    input_string = request.json.get('input_string')
    key = request.json.get('key')
    # 进行 AES 加密
    encrypted_string = aes_encrypt(input_string, key)
    # 返回 JSON 格式的加密字符串
    return jsonify({"encrypted_string": encrypted_string})


@app.route('/decrypt', methods=['POST'])
def handle_decrypt():
    # 获取请求中的加密字符串和密钥
    encrypted_string = request.json.get('encrypted_string')
    key = request.json.get('key')
    # 进行 AES 解密
    decrypted_string = aes_decrypt(encrypted_string, key)
    # 返回 JSON 格式的解密字符串
    return jsonify({"decrypted_string": decrypted_string})


if __name__ == '__main__':
    app.run(debug=True)
```

(2) Code of html browser

```
    <!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>SHA-256 & AES</title>
</head>
<body>
    <h1>SHA-256 & AES Encryption/Decryption</h1>

    <h2>SHA-256 Hash</h2>
    <label for="sha256-input">Input String:</label>
    <input type="text" id="sha256-input" placeholder="Enter string here">
    <button onclick="calculateHash()">Calculate Hash</button>
    <br><br>
    <label for="sha256-output">Hash Value:</label>
```

```
<input type="text" id="sha256-output"    readonly style="width: 500px;">

<h2>AES Encryption</h2>
<label for="aes-input">Input String:</label>
<input type="text" id="aes-input" placeholder="Enter string here">
<label for="aes-key">Key:</label>
<input type="text" id="aes-key" placeholder="Enter key here">
<button onclick="encrypt()">Encrypt</button>
<br><br>
<label for="aes-encrypted">Encrypted String:</label>
<input type="text" id="aes-encrypted"    readonly style="width: 500px;">

<h2>AES Decryption</h2>
<label for="aes-decrypted-input">Encrypted String:</label>
<input type="text" id="aes-decrypted-input" placeholder="Enter encrypted string here">
<label for="aes-decrypt-key">Key:</label>
<input type="text" id="aes-decrypt-key" placeholder="Enter key here">
<button onclick="decrypt()">Decrypt</button>
<br><br>
<label for="aes-decrypted">Decrypted String:</label>
<input type="text" id="aes-decrypted"    readonly style="width: 500px;">

<script>
    async function calculateHash() {
        const inputString = document.getElementById('sha256-input').value;
        const response = await fetch('/sha256', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
            body: JSON.stringify({ input_string: inputString }),
        });
        const data = await response.json();
        document.getElementById('sha256-output').value = data.hash_value;
    }

    async function encrypt() {
        const inputString = document.getElementById('aes-input').value;
        const key = document.getElementById('aes-key').value;
        const response = await fetch('/encrypt', {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
            },
                body: JSON.stringify({ input_string: inputString, key: key }),
            });
            const data = await response.json();
            document.getElementById('aes-encrypted').value = data.encrypted_string;
        }

        async function decrypt() {
            const encryptedString = document.getElementById('aes-decrypted-input').value;
            const key = document.getElementById('aes-decrypt-key').value;
            const response = await fetch('/decrypt', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                },
                body: JSON.stringify({ encrypted_string: encryptedString, key: key }),
            });
            const data = await response.json();
            document.getElementById('aes-decrypted').value = data.decrypted_string;
        }
    </script>
</body>
</html>
```

Result:



SHA-256 & AES Encryption/Decryption

SHA-256 Hash

Input String: 61 62 63 64 62 63 64 65 63

Hash Value: 248d6a61d20638b8e5c026930c3e6039a33ce45964ff2167f6ecedd419db06c1

AES Encryption

Input String: 248d6a61d20638b8e5c026 Key: qpalwoskeidjrufhtyg

Encrypted String: 9zveFm7TZo9bodr/VeHRUe0woLfN7SMzj846RRFXd/c4hJxrroba/qoo38LiNbsryAkY

AES Decryption

Encrypted String: 9zveFm7TZo9bodr/VeHRU Key: qpalwoskeidjrufhtyg

Decrypted String: 248d6a61d20638b8e5c026930c3e6039a33ce45964ff2167f6ecedd419db06c1