Nanjing University of Information Science & Technology

Experiment (Internship) Report

|  |  |  |  |
| --- | --- | --- | --- |
| 202283890036 | 黄家睿(Jerry Huang) | Internet of Things | Zhiguo Qu(瞿治国） |

Implementation and Analysis of the AES Encryption Algorithm

1．实验目的：

1. Understanding AES Encryption and Decryption Algorithms；
2. Implement the AES encryption and decryption algorithm, and analyze its performance.

2．Experimental content：

1. Implement the AES encryption and decryption algorithm.；
2. Analyze the encryption results.
3. Experimental steps
4. Previously imported libraries

Implementing AES is quite complex, but you can use the Crypto third-party library in Python, which is a classic Python third-party library for cryptographic algorithms. To quickly install the AES encryption implementation in PowerShell, simply type pip install pycryptodome.

After importing the library, you need to import the following methods.。

from Crypto.Cipher import AES

from Crypto.Util.Padding import pad, unpad

import base64

1. Implementation of AES encryption

def aes\_encrypt(data, key):

*# AES要求密钥长度为16、24或32字节*

key = key.encode('utf-8') *# 转为字节*

cipher = AES.new(key, AES.MODE\_CBC) *# 使用CBC模式创建AES对象*

*# 对数据进行填充，使其长度为16的倍数*

data = data.encode('utf-8') *# 转为字节*

padded\_data = pad(data, AES.block\_size) *# 填充*

*# 加密数据*

encrypted\_data = cipher.encrypt(padded\_data)

*# 返回加密后的数据，包含IV（初始向量）*

*# 将IV和加密数据一起返回，方便解密时使用*

return base64.b64encode(cipher.iv + encrypted\_data).decode('utf-8')

1. AES解密的实现

def aes\_decrypt(encrypted\_data, key):

encrypted\_data = base64.b64decode(encrypted\_data) # 解码Base64数据

# 提取IV（前16个字节是IV）

iv = encrypted\_data[:16]

encrypted\_data = encrypted\_data[16:] # 剩下的是加密数据

# 创建AES对象，并使用提取的IV

key = key.encode('utf-8') # 转为字节

cipher = AES.new(key, AES.MODE\_CBC, iv) # 使用CBC模式创建AES对象

# 解密数据，并移除填充

decrypted\_data = unpad(cipher.decrypt(encrypted\_data), AES.block\_size)

return decrypted\_data.decode('utf-8')

1. Encryption and decryption effect presentation

*# 示例使用*

key = "JerryHuangandyou" *# 密钥需要是16、24或32字节（16字节为AES-128）*

data = "This is a secret message"

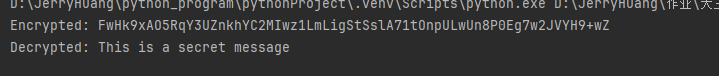
*# 加密*

encrypted = aes\_encrypt(data, key)print("Encrypted:", encrypted)

*# 解密*

decrypted = aes\_decrypt(encrypted, key)print("Decrypted:", decrypted)

The output：



1. 实验分析和总结
2. The difference between AES and Feistel

Feistel network: The basic structure of a Feistel network involves dividing a data block into two halves and performing encryption operations on one half, then swapping these two parts. Each round of encryption operates on one part of the data, achieving encryption through multiple rounds (typically 16 rounds). During decryption, the Feistel network exhibits symmetry, meaning that the encryption and decryption processes are the same, except that the order of key usage is reversed.AES (Advanced Encryption Standard), which uses a combination of substitution and permutation operations, performs all encryption steps on the entire data block. Each round of AES involves byte substitution (SubBytes), row shifting (ShiftRows), column mixing (MixColumns), and round key addition (AddRoundKey). These steps are applied to the entire data block (128 bits) rather than processing it in two parts.

Feistel network: The encryption process of a Feistel network is relatively simple, mainly achieved through substitution and permutation operations. Each round of operations is straightforward, usually implemented through basic XOR, permutation, and simple encryption functions.

AES: Each round of operations in AES is more complex. In addition to byte substitution and row shifting, AES also introduces the column mixing operation (MixColumns), which makes AES more powerful in terms of confusion and diffusion compared to the Feistel network. The column mixing operation causes the bytes in each column to influence each other, greatly enhancing the confusion of the data.

1. The advance of AES

**Substitution-permutation structure**: The substitution-permutation network design of AES makes each round of encryption operations more complex, enhancing diffusion and confusion, thereby providing stronger security.

**Column mix**: Unlike the swap operation in the Feistel network, AES complicates the data through column mix in each round, which enhances AES's ability to resist attacks.

**Higher resistance to differential attacks**: AES effectively enhances its defense against differential attacks by introducing more complex S-boxes and nonlinear transformations.

AES fully considered resistance to differential and linear attacks during its design,

employing complex byte substitution, row shifting, column mix, and round key addition operations, making AES more difficult to attack. The key length of AES-256 provides higher resistance to brute-force attacks.

1. The specific algorithmic process of AES implementation

The key expansion algorithm consists of a key schedule algorithm and a round key generation algorithm. The key schedule algorithm selects different configurations based on the length of the input key, then divides the input key into multiple words and processes these words. The round key generation algorithm generates new round keys by transforming the previous round key and combining it with the key schedule algorithm. The plaintext to be encrypted (16-byte data) is arranged into a 4x4 matrix. Then, the 16-byte key is also arranged into a 4x4 matrix. These two matrices are XORed (XOR operation for each corresponding byte), thus completing the round key addition. Each byte resulting from the round key addition is represented in hexadecimal. Using the first hexadecimal digit as the row and the second as the column, the corresponding number is looked up in the S-box table to replace the original number. The matrix after byte substitution undergoes row shifting. Specifically, the first row remains unchanged, the second row is cyclically shifted left by 1 byte, the third row by 2 bytes, and the fourth row by 3 bytes. The data after row shifting undergoes column confusion, which involves arithmetic operations in the field GF(2^8). The steps of round key addition, byte substitution, row shifting, and column confusion are repeated: depending on the length of the key (128 bits, 192 bits, or 256 bits), the AES algorithm will perform a different number of rounds (10 rounds, 12 rounds, and 14 rounds respectively). In each round, the four steps of round key addition, byte substitution, row shifting, and column confusion are executed. Finally, the AES encryption process is completed.