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 RSA Encryption Algorithm

1．Experimental Purpose：

1. The Algorithm flowchart of RSA
2. Implement and apply the RSA algorithm

2．Experimental content：

1. Implement the RSA encryption and decryption algorithm
2. Apply RSA to encrypt the file and test the encryption time.
3. Experimental procedure

The AES implementation is more complicated and can be implemented using the Crypto third-party library in Python, which is a classic Python third-party library for cryptography algorithms. Enter in the PowerShell：

*pip install pycryptodome*

You can quickly install the AES encryption implementation.

1. The RSA algorithm for encryption

from Crypto.PublicKey import RSA

from Crypto.Cipher import PKCS1\_OAEP

from Crypto.Random import get\_random\_bytes

# 生成RSA密钥对

def generate\_rsa\_keys(bits=2048):

key = RSA.generate(bits)

private\_key = key.export\_key()

public\_key = key.publickey().export\_key()

return private\_key, public\_key

# 使用公钥加密

def encrypt\_message(public\_key, message):

rsa\_key = RSA.import\_key(public\_key)

cipher = PKCS1\_OAEP.new(rsa\_key)

encrypted\_message = cipher.encrypt(message.encode())

return encrypted\_message

Style

1. The RSA algorithm for decryption

# 使用私钥解密

def decrypt\_message(private\_key, encrypted\_message):

rsa\_key = RSA.import\_key(private\_key)

cipher = PKCS1\_OAEP.new(rsa\_key)

decrypted\_message = cipher.decrypt(encrypted\_message).decode()

return decrypted\_message

1. The RSA algorithm results are presented

*# 示例：生成密钥、加密和解密*if \_\_name\_\_ == "\_\_main\_\_":

*# 生成RSA密钥对*

private\_key, public\_key = generate\_rsa\_keys()

*# 待加密的消息*

message = "Hello, this is a secret message!"

*# 使用公钥加密*

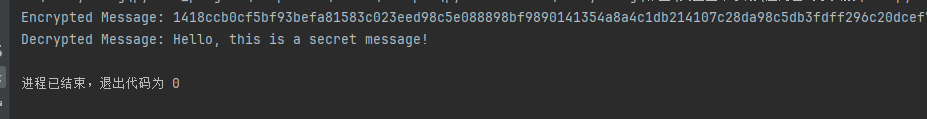
encrypted\_message = encrypt\_message(public\_key, message)

print(f"Encrypted Message: {encrypted\_message.hex()}")

*# 使用私钥解密*

decrypted\_message = decrypt\_message(private\_key, encrypted\_message)

print(f"Decrypted Message: {decrypted\_message}")



Our resulting ciphertext is for：

The results of the ciphertext is：



Hello, this is a secret message!

1. Analysis and summary of the experiments
2. Specific algorithm procedure of RSA

The RSA algorithm has three main steps: key generation, encryption, and decryption.

1. Key generation

RSA uses a pair of keys: public key and private key, public key is used for data encryption, can be disclosed to anyone, private key is used for decryption, can only be stored confidential.

The key generation process is performed as follows：

1. Select two large exponents, p and q, and compute their product
2. Calculate the Euler function of n
3. Select a small integer e, making the，e and are mutually.
4. Calculate d so that ，That is, find the module of d so that d is e.

The generation of key pairs： public key （e，n）

private key （d，n）

1. Encryption process

Encryption is done with the public key (e, n) (e, n) (e, n). Given the plaintext M, the encryption process is：

Among them, CCC is ciphertext, MMM is plaintext (plaintext must be less than nnn).

1. Decryption process

Decryption is done with the private key (d, n) (d, n) (d, n). Given the ciphertext CCC, the decryption process is：

Among them, MMM is the decrypted clear text.

1. Array signature

The RSA can also be used for digital signatures. In this case, the hash value (summary) of the message is signed with the private key, and the recipient verifies the authenticity of the signature by using the public key of the sender.