南京信息工程大学 实验（实习）报告

|  |  |  |
| --- | --- | --- |
| 202283890036 | 黄家睿 | Internet of Things |

RSA加密算法的实现与分析

1．实验目的：

1. 掌握RSA算法流程
2. 实现并应用RSA算法

2．实验内容：

1. 实现RSA加解密算法
2. 应用RSA对文件进行加密，并测试加密时间。
3. 实验步骤

AES实现较为复杂，可以使用Python中的Crypto第三方库来实现，这是一个经典的用于密码学算法的Python第三方库。在PowerShell中输入：

*pip install pycryptodome*

即可快速安装AES加密实现。

1. RSA算法加密

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. RSA算法解密

# 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. RSA算法结果展示

*# 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*

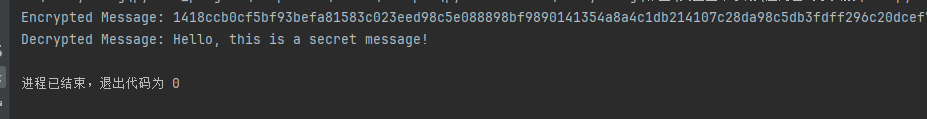
encrypted\_message = encrypt\_message(public\_key, message)

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}")



我们的得到的密文为：

对密文解密的结果为：



Hello, this is a secret message!

1. 实验分析和总结
2. RSA的具体算法过程

RSA算法有三个主要步骤：密钥生成，加密，解密

1. 密钥生成

RSA使用一对密钥：公钥和私钥，公钥用于数据的加密，可以公开给任何人，私钥用于解密，只能保密存储。

密钥的生成过程如下：

1. 选择两个大指数p和q，并计算他们的乘积
2. 计算n的欧拉函数
3. 选择一个小的整数 e，使得，并且e和互质。
4. 计算d，使得 ，即找到d使得 d是e的模逆。

密钥对的生成：公钥（e，n）

私钥（d，n）

1. 加密过程

加密是通过公钥 (e,n)(e, n)(e,n) 完成的。给定明文 M，加密过程是：

其中，CCC 是密文，MMM 是明文（明文必须小于 nnn）。

1. 解密过程

解密是通过私钥 (d,n)(d, n)(d,n) 完成的。给定密文 CCC，解密过程是：

其中，MMM 是解密后的明文

1. 数组签名

RSA也可以用于数字签名。在这种情况下，消息的哈希值（摘要）用私钥进行签名，接收方使用发送方的公钥验证签名的真实性。