

实 验 报 告

姓名 张诺宇 班级 0211213 学号 028121308 组别 实验日期 2023.04.23

课程名称 数字内容安全 同实验者 指导教师 李冬梅 成绩

实验1 RSA加密算法

1. 实验目的

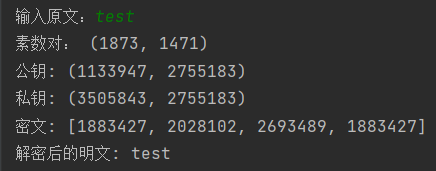
了解数据加密的原理，掌握公钥加密算法。

1. 实验内容

编程实现RSA加解密算法。

1. 实验过程

import random  
import math  
  
  
# 生成一个素数  
def generate\_prime\_number():  
 while True:  
 prime = random.randint(2 \*\* 10, 2 \*\* 11)  
 if is\_prime(prime):  
 return prime  
  
  
# 判断一个数是否为素数  
def is\_prime(n):  
 if n < 2:  
 return False  
 for i in range(2, int(math.sqrt(n)) + 1):  
 if n % i == 0:  
 return False  
 return True  
  
  
# 计算欧拉函数的值  
def euler\_func(p, q):  
 return (p - 1) \* (q - 1)  
  
  
# 计算最大公因数  
def gcd(a, b):  
 while b:  
 a, b = b, a % b  
 return a  
  
  
# 计算模反元素  
def mod\_inverse(e, phi):  
 d = 0  
 x1 = 0  
 x2 = 1  
 y1 = 1  
 temp\_phi = phi  
  
 while e > 0:  
 temp1 = temp\_phi // e  
 temp2 = temp\_phi - temp1 \* e  
 temp\_phi = e  
 e = temp2  
  
 x = x2 - temp1 \* x1  
 y = d - temp1 \* y1  
  
 x2 = x1  
 x1 = x  
 d = y1  
 y1 = y  
  
 if temp\_phi == 1:  
 return d + phi  
  
  
# 生成密钥  
def generate\_keypair():  
 p = generate\_prime\_number()  
 q = generate\_prime\_number()  
 n = p \* q  
 phi = euler\_func(p, q)  
 e = random.randint(1, phi - 1)  
 while gcd(e, phi) != 1:  
 e = random.randint(1, phi - 1)  
 d = mod\_inverse(e, phi)  
 return ((p, q), (e, n), (d, n))  
  
  
# 加密  
def encrypt(public\_key, message):  
 e, n = public\_key  
 cipher\_text = [pow(ord(char), e, n) for char in message]  
 return cipher\_text  
  
  
# 解密  
def decrypt(private\_key, cipher\_text):  
 d, n = private\_key  
 plain\_text = [chr(pow(char, d, n)) for char in cipher\_text]  
 return ''.join(plain\_text)  
  
  
message = input("输入原文：")  
prime, public\_key, private\_key = generate\_keypair()  
cipher\_text = encrypt(public\_key, message)  
print('素数对：', prime)  
print('公钥:', public\_key)  
print('私钥:', private\_key)  
print('密文:', cipher\_text)  
plain\_text = decrypt(private\_key, cipher\_text)  
print('解密后的明文:', plain\_text)



1. 实验总结

在实验过程中一开始没有注意到RSA算法要求的大素数的要求，选取了比较小的素数，发现存在比较容易暴力破解的可能。然后我实现了RSA算法的三个关键步骤：生成密钥、加密和解密。在生成密钥的步骤中，我使用了随机生成素数的方法来保证生成的密钥的安全性。在加密和解密的过程中，我使用了Python内置函数pow来计算幂余。

通过这次实验，我更加深入地理解了RSA算法的原理和实现方法，并掌握了Python编写加密算法的基本技能。同时，我也认识到加密算法在保护信息安全方面的重要性。