**广州大学学生实验报告**

**开课学院及实验室：**网络空间安全学院 **2023年 5 月9 日**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **学院** | 网络空间安全学院 | **年级/专业/班** | 网安211 | **姓名** |  | **学号** |  |
| **实验课程名称** | 密码技术及应用 | | | | | **成绩** |  |
| **实验项目名称** | 公钥密码实验—RSA算法 | | | | | **指导**  **老师** | 李树栋 |

1. 实验目的
2. 实现RSA加解密
3. 实现费马因子分解
4. 练习小指数迭代攻击方法（选做）
5. 实验内容
6. 使用python实现RSA加解密算法，根据参数p=61,q=53,m=’hello world’随机选取e生成公私钥对和密文c，再对密文解密还
7. 使用费马因子分解法分解整数：476714679652321667
8. 使用小指数迭代攻击法破解以下密文

e = 3

c =

358367237930161611287553617821502821834060085243898907996647845293765170791369942269257353667444037801340579758680509083796235778230457983351909760841969489695686448841352419071149675085236615042625125

n =

16503642207789401613904637446670329919611766947715053323104219193209988406515756550342556551087269744552311616952010203456790322170970296166677505605996894842559532450550835236993676533810157091757910377447156599161839816759569000052558651907868440945641567209771534186995020539019344287451849086813001914316093471981888660658969896738024934729735528534566540708968896374055591535315302278504854219775669117057573452687716997892400605761254842309025919547236250830974219452915644544246195460317093268045928951431217025117317053023012828445449507248638685265783199196296512529426531595699588709043560235664398614480367

1. 实验过程及结果

3.1

# python实现RSA加密

import gmpy2

import random

import math

from Crypto.Util.number import bytes\_to\_long

from Crypto.Util.number import long\_to\_bytes

def randchose\_e(phi):

    while True:

        e = random.randint(2, phi-1)

        if math.gcd(e, phi) == 1:

            break

    print("公钥e:", e)

    return e

# p, q 是两个大质数

p = 61

q = 53

# n 是RSA模数

n = p\*q

# 求欧拉函数 φ(n)

phi = (p-1)\*(q-1)

# 随机生成的公钥e

e = randchose\_e(phi)

# 用扩展的欧几里得算法计算 e 对于 φ(n) 的模逆元 d

d = gmpy2.invert(e, phi)

# 计算明文 m 的 RSA 加密结果

m = bytes\_to\_long(b'hello world')

c = pow(m, e, n)

print("公钥 (n, e):", (n, e))

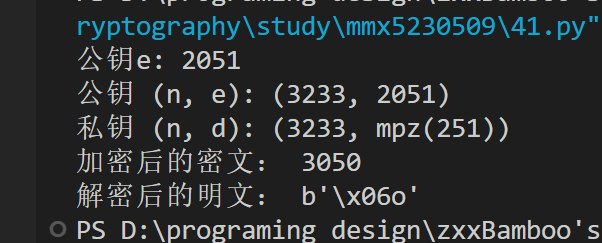
print("私钥 (n, d):", (n, d))

print("加密后的密文：", c)

# 解密

m\_decr = pow(c, d, n)

print("解密后的明文：", long\_to\_bytes(m\_decr))



3.2

import gmpy2

def fermat\_factorization(n):

    a = gmpy2.isqrt(n)

    b2 = gmpy2.square(a) - n

    while not gmpy2.is\_square(b2):

        a += 1

        b2 = gmpy2.square(a) - n

    p = a + gmpy2.isqrt(b2)

    q = a - gmpy2.isqrt(b2)

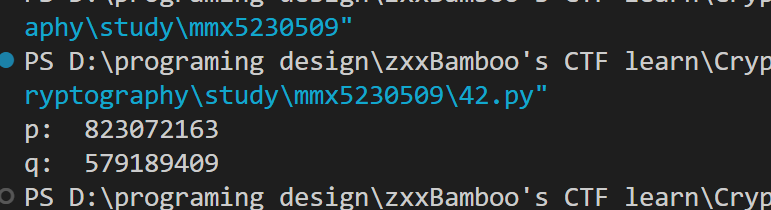
    return int(p), int(q)

n = 476714679652321667

p, q = fermat\_factorization(n)

print("p: ", p)

print("q: ", q)



3.3

import gmpy2

# 设置公钥指数

e = 3

# 从文件读取密文和模

with open('c.txt', 'r') as f:

    c = int(f.read())

with open('n.txt', 'r') as f:

    n = int(f.read())

# 检查是否满足小指数攻击的条件

if gmpy2.iroot(c, e)[1]:

    # 计算密文的e次方根

    m = gmpy2.iroot(c, e)[0]

    # 将解密后的消息从整数转换为字符串

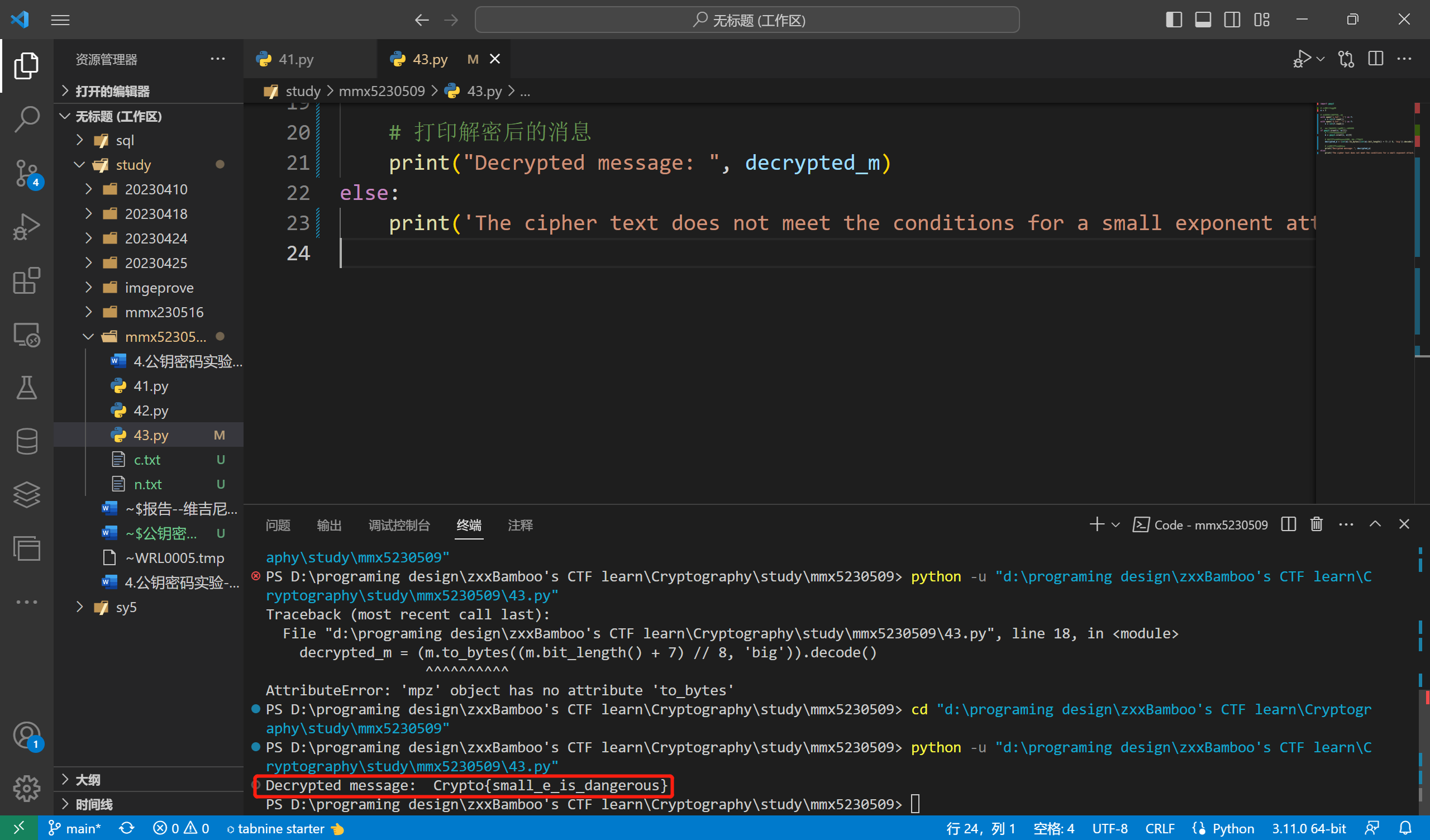
    decrypted\_m = (int(m).to\_bytes((int(m).bit\_length() + 7) // 8, 'big')).decode()

    # 打印解密后的消息

    print("Decrypted message: ", decrypted\_m)

else:

    print('The cipher text does not meet the conditions for a small exponent attack.')



1. 实验总结

遇到问题：

Traceback (most recent call last):

File "d:\programing design\zxxBamboo's CTF learn\Cryptography\study\mmx5230509\43.py", line 18, in <module>

decrypted\_m = (m.to\_bytes((m.bit\_length() + 7) // 8, 'big')).decode()

^^^^^^^^^^

AttributeError: 'mpz' object has no attribute 'to\_bytes'

因为**gmpy2**库的**mpz**类没有**to\_bytes**方法。可以使用Python内置的**int**类型来解决这个问题。将**mpz**对象转换为**int**对象，然后进行字节转换：

这个实验主要包含以下几个部分：

1. RSA加解密： 在这部分，我实现了RSA的加密和解密过程。我首先定义了两个素数p和q，然后计算出RSA的模n和phi(n)。然后，我使用这些参数生成了RSA的公钥和私钥。之后，我选择了一个消息m进行加密，并对加密后的密文进行解密，得到的结果与原始消息相同，从而验证了RSA加密和解密的正确性。

2. 费马因子分解： 在这部分，我使用费马因子分解法对一个大整数进行因子分解。费马因子分解法的基本思想是将一个整数表示为两个平方数之差。虽然费马因子分解法并不总是最有效的方法，但它仍然是一种有趣且具有教育意义的因子分解方法。

3. 小指数迭代攻击： 在这部分，我尝试对一个使用小公钥指数加密的RSA密文进行攻击。这种攻击的基本思想是，如果原始消息m小于模n，那么m的e次方不会大于n的e次方，因此m^e mod n = m^e。所以，我可以通过计算密文的e次方根来获取原始消息m。我使用gmpy2库的iroot函数来计算密文的e次方根。

通过这个实验，我不仅学习了RSA的工作原理，还了解了如何对使用小公钥指数的RSA加密进行攻击，以及如何使用费马因子分解法对整数进行因子分解。这些都是密码学中非常重要的概念。