**西安电子科技大学网信院**

**信息安全基础与密码学**

**综合实验**

**实 验 报 告（四）**

**ElGamal公钥密码算法**

**班级：**

**姓名：**

**学号：**

**日期：2024-11-9**

一、实验目的（包括实验环境、实现目标等等）

Python环境

实现ElGamal加解密算法

（1）大素数可用形式为p=2q+1形式的强素数，这里面q也是素数，大素数p和本原根自己生成。（关于强素数，请同学们查阅文献资料了解学习）

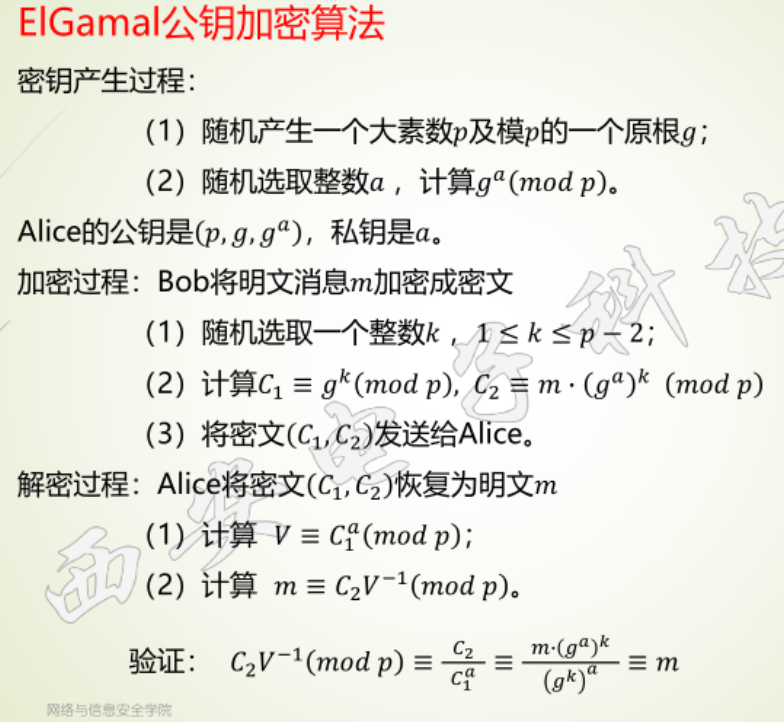
（2）解密出的结果要与明文作对比，验证解密是否正确。

（3）需显示出中间数据，包括p，g，g^a，k 及密文C=(C1, C2)

二、方案设计

（包括背景、原理、必要的公式、图表、算法步骤等等）

ElGamal的密钥生成、加解密原理如课件所示



课件中没有提到的是，p应该有一个大素数因子，这样既便于找到本原元，也保证了安全性

生成密钥是ElGamal加密算法的关键所在，这一过程，我采用如下算法步骤：

Step1：生成一个大素数，这里是大概512bits的数

Step2：验证是否是素数，如果是，则进入下一步

Step3: 找到的一个生成元

Step4: 随机选取合适的整数,计算

其中，值得强调的是找到生成元的过程，由于是极大的数，根据定义遍历群内的每一个元素，验证模幂是否为1是不可能的。因此，采用如下算法：

先随机生成一个，根据数论，如果对的每一个质因数都成立，那么就是的一个生成元。这是因为如果满足了上述条件，那么满足的最小正幂次只能为，从而覆盖了整个群。

综上，找生成元的过程可以用如下算法步骤描述：

Step1: 质因数分解

Step2: 取一个群内的元素

Step3: 验证对的每一个质因数是否都成立，如果是，则返回，生成元已经找到。如果不是，则返回Step2，重新取一个

而生成时，让，这也确保了Step1质因数分解是容易做到的。

三、方案实现

（包括算法流程图、主要函数的介绍、算法实现的主要代码等等）

主要函数：

def EEA(r0, r1)

EEA算法，求乘法逆元、gcd

def is\_primitive\_generator(g, mod, factors\_mod\_minus\_1)

判断是否是的生成元，factors\_mod\_minus\_1为的质因数分解。

def is\_probable\_prime(n, repetitions=20, mode='fermat')

实验1已经实现的，费马素性检验

def prime\_factors\_decomposition(number)

质因数分解。

主要的类：

class ElGamalCipher():

def \_\_init\_\_(self, public\_key=None, private\_key=None, generator=None, modulus=None):

@staticmethod  
def generate\_keys():

def encrypt(self, block\_size=63)

def decrypt(self, mode='ignore', block\_size=63)

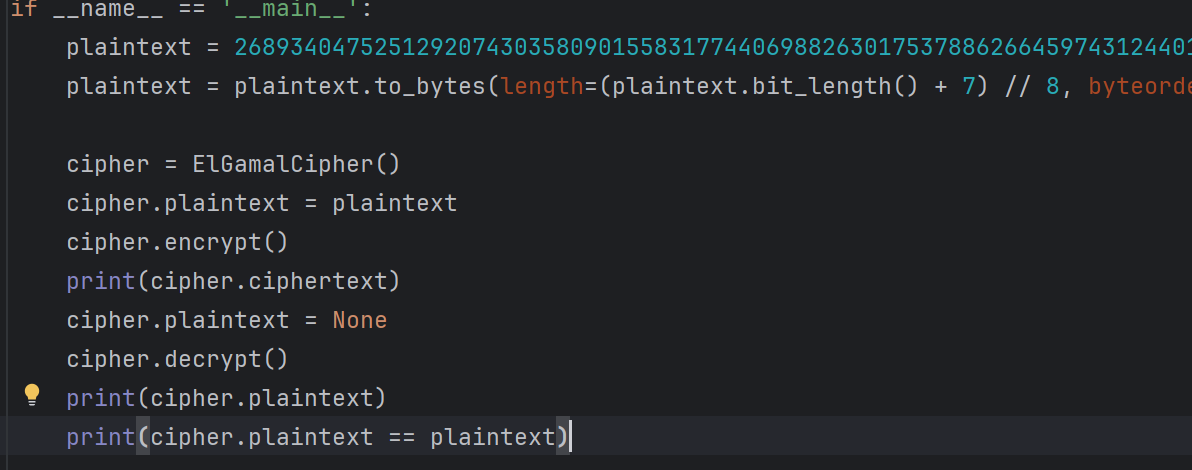
这个类主要有这几个属性：生成元g、模数mod、私钥private\_key、公钥public\_key、plaintext、ciphertext

几个类方法分别实现了：生成密钥对、加密、解密。依据的就是课件中的原理、公式。调用encrypt()后，密文会保存在ciphertext属性中，调用decrypt()后，明文保存在plaintext属性中

使用这个类进行加解密时，要求输入的plaintext是以字节序列形式表示的，因此，在main函数开始时，首先将整数形式的明文转换成字节序列。Python提供了便捷的转换方法。

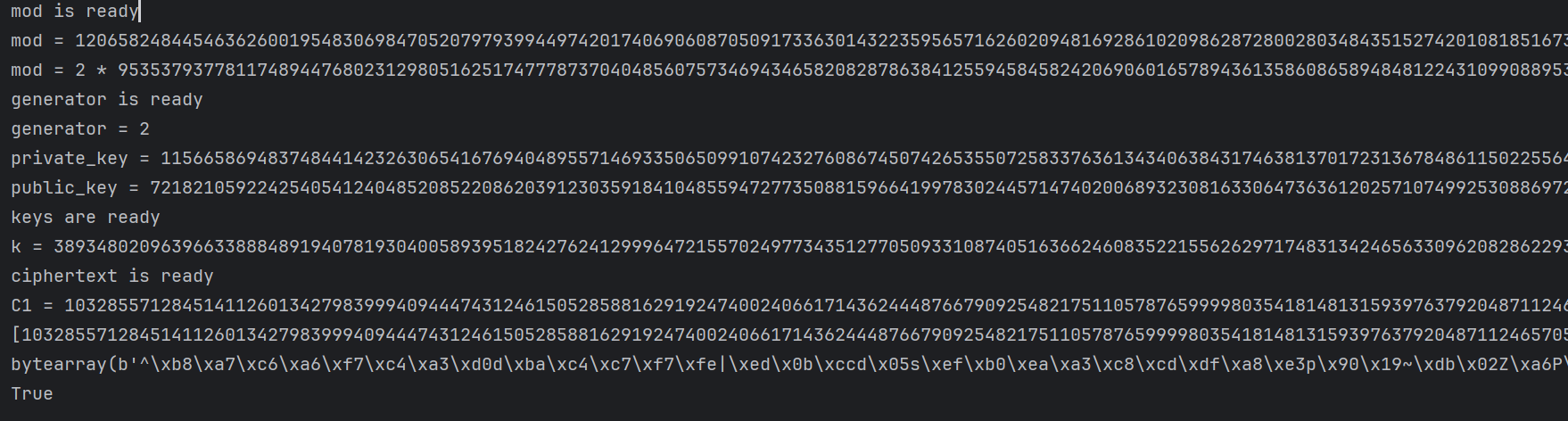
四、数据分析(包括算法测试数据的分析，运行结果截图等等)

取测试用例1:



cipher.plaintext = None

这一步为了确保结果打印出True不是因为一开始的cipher.plaintext = plaintext。而是因为decrypt()的调用确实修改了cipher.plaintext

控制台的输出展示了：

模数：

12065824844546362600195483069847052079793994497420174069060870509173363014322359565716260209481692861020986287280028034843515274201081851673124990980107693

可以表示成：

2\*9535379377811748944768023129805162517477787370404856075734694346582082878638412559458458242069060165789436135860865894848122431099088953326535159 + 1

生成元是2：

generator = 2

随机生成的私钥private\_key ，即：

11566586948374844142326306541676940489557146933506509910742327608674507426535507258337636134340638431746381370172313678486115022556438628729098952104946447

公钥public\_key，即：

7218210592242540541240485208522086203912303591841048559472773508815966419978302445714740200689323081633064736361202571074992530886972512493805904820773354

加密过程中随机生成的k：

3893480209639663388848919407819304005893951824276241299964721557024977343512770509331087405163662460835221556262971748313424656330962082862293151884906616

C1，C2：

C1：

10328557128451411260134279839994094447431246150528588162919247400240661714362444876679092548217511057876599998035418148131593976379204871124657050309203123

C2：

4916820876395118722758750738015382356084407646425145533893813927861209714326156444112931244293475697957940831647478415073563245457063193181859748241747126

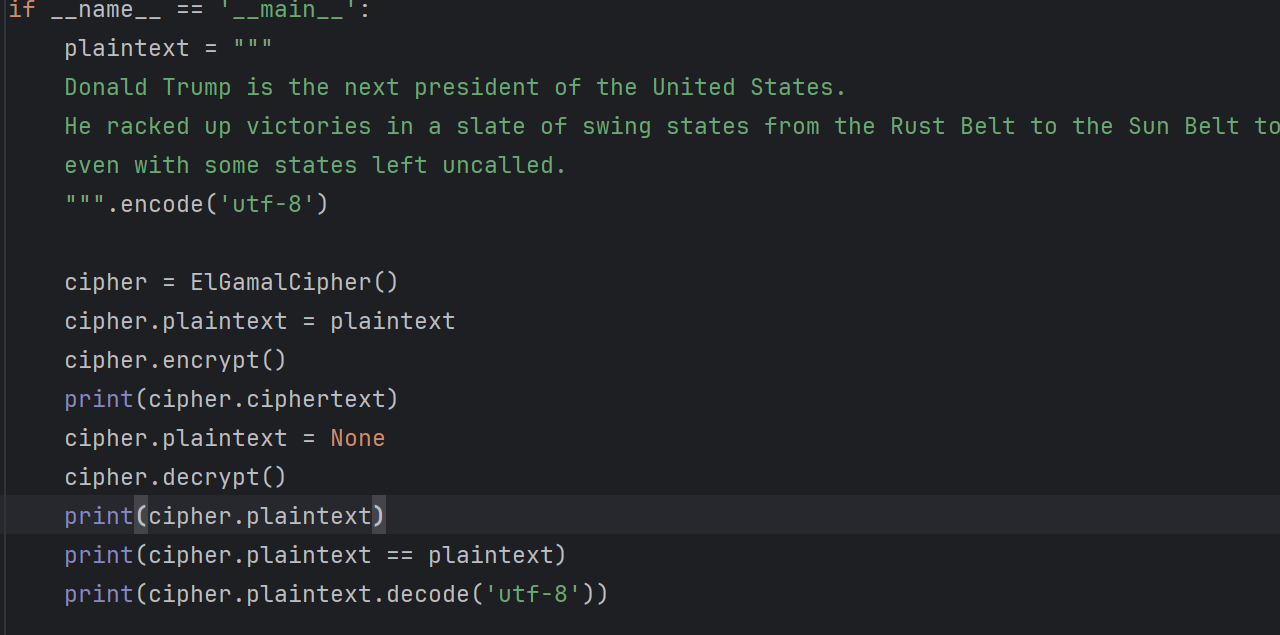
解密出的东西，以字节序列表示：

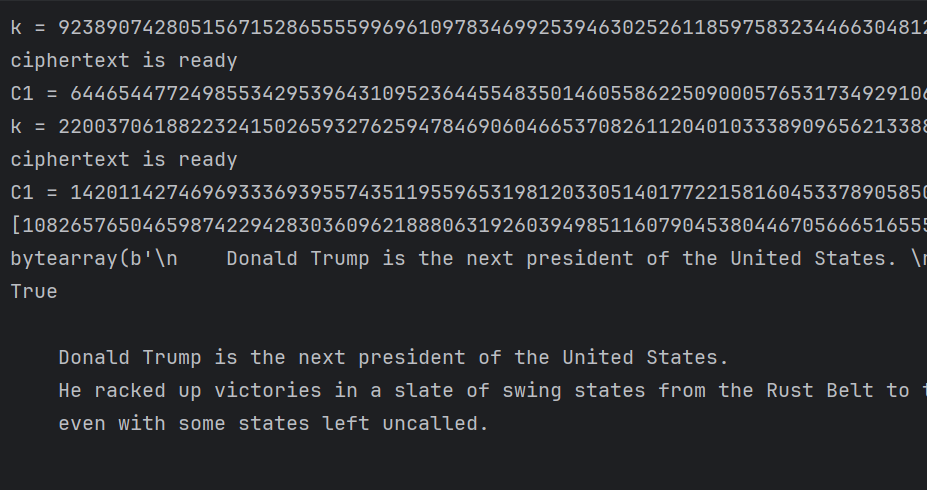
bytearray(b'^\xb8\xa7\xc6\xa6\xf7\xc4\xa3\xd0d\xba\xc4\xc7\xf7\xfe|\xed\x0b\xccd\x05s\xef\xb0\xea\xa3\xc8\xcd\xdf\xa8\xe3p\x90\x19~\xdb\x02Z\xa6P\xce}\xc3l\xf6ru\xe1\x1b\xfc\xces\xd3\xd7J\xfa')

解密出的东西是否等于原明文：

True

不仅如此，我实现的算法还可以加密字符串消息、甚至文件。加密一段字符串消息作为展示：





五、思考与总结

1. 请简述什么是本原根，给定素数P，如何求其本原根？。

满足的最小正幂次为的就是的一个生成元,又称本原根、本原元。

对于后一个问题，在“方案设计”已经提到：

Step1: 质因数分解

Step2: 取一个群内的元素

Step3: 验证对的每一个质因数是否都成立，如果是，则返回，生成元已经找到。如果不是，则返回Step2，重新取一个

给定素数, 如果是一个很大的数且质因数分解困难，那么找到模的本原元会变得非常困难。也因如此，我们在ElGamal生成密钥时，让，其中、都是素数.

1. 如果𝑘与𝑝−1不互素，可能会发生什么情况？

在ElGamal中，解密依赖于找到的逆元，即。如果不互素,有可能没有逆元，解密计算将无法正确还原出明文，从而导致解密失败。

此外，还有可能导致密钥重用的现象，因为即使随机选择了不同的 ，有可能是相同的，攻击者可以通过比较不同密文中的重复模式来推测私钥。

1. 实验过程中还遇到了什么问题，如何解决的？通过该实验有何收获？

实验过程较为顺利，没有遇到其他问题

通过该实验，我：

理解了强素数的原理，及其生成过程

深入理解本原根的概念、以及如何找到它

掌握 ElGamal 生成密钥、加解密流程、与现代密码学课程融会贯通

提高编程实现和调试能力

**实验报告提交说明**：

1. 实验报告同时提交word文档与源代码（.c或者.py）。
2. 实验报告与源代码命名规则：实验4-学号-姓名，例如：实验4-20009200400-张三。
3. 请于截止日期前在西电智课平台（学在西电）提交相关文档，逾期未提交，该部分成绩记为0分，周知。