**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（三）**

**基于中国剩余定理的秘密共享方案**

**班级：2018039**

**姓名：叶振雷**

**学号：20009101064**

**日期：2022.11.12**

一、实验目的（包括实验环境、实现目标等等）

实验环境：pycharm+anoconda+windows11

1.通过算法编程，熟悉使用python库的基本函数操作

2.通过编程实现基于中国剩余定理的秘密共享方案，加深对于中国剩余定理的理解与运用

3.体会密码学与数论的紧密联系,将数论的知识运用于密码学的方案设计中

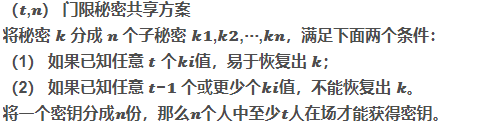
4.提高逻辑思维能力与实践能力

二、方案设计

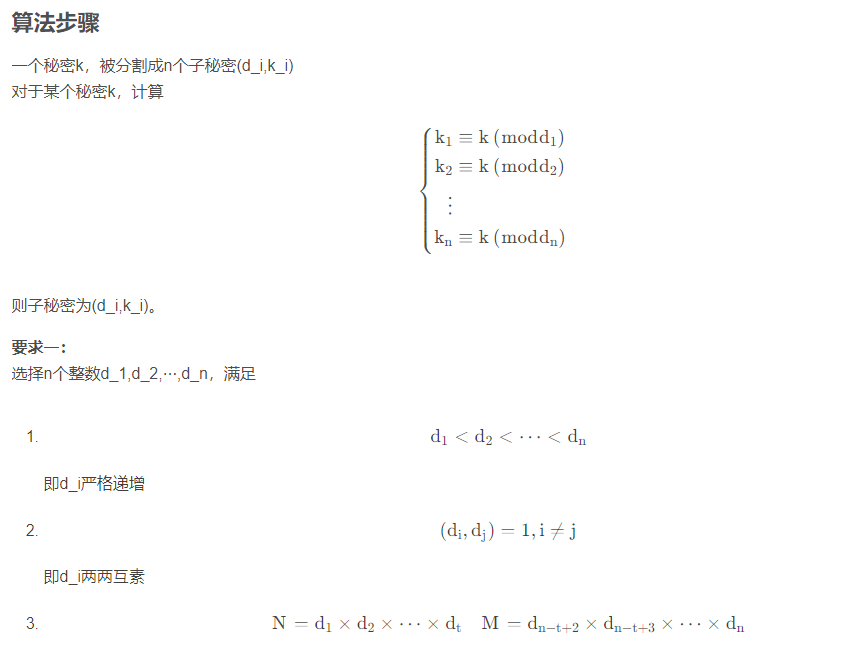
（包括背景、原理、必要的公式、图表、算法步骤等等）

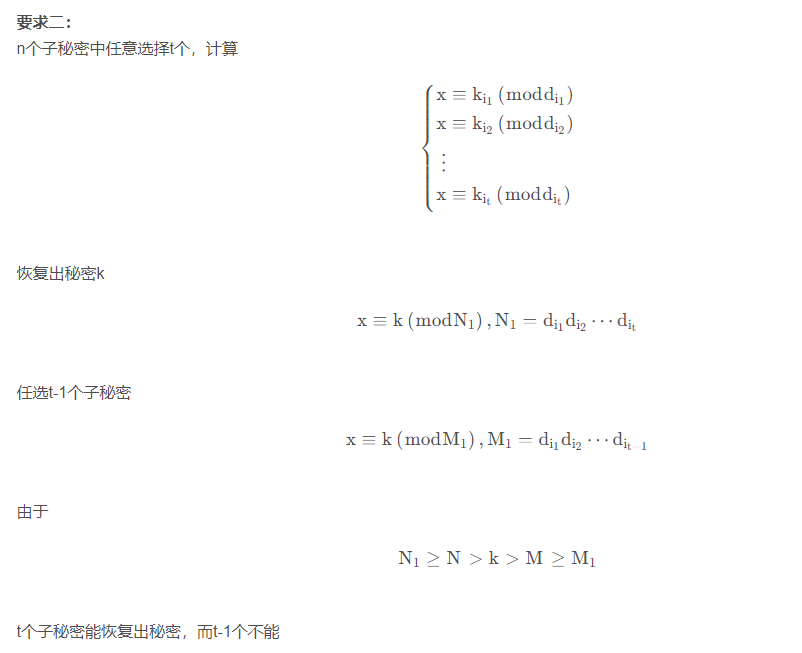
秘密共享是将秘密以适当的方式拆分，拆分后的每一个子秘密由不同的参与者管理，单个参与者无法恢复秘密信息，只有若干个参与者一同协作才能恢复秘密消息。并且，当其中某些参与者出问题时，秘密仍可以恢复。

并且秘密共享是现代密码学的重要分支,也是保证信息安全的重要方法之一,已成为信息安全领域的研究热点。秘密共享为密钥管理和数据保密提供了一种全新的思想。秘密共享技术不仅可以防止由于单个保管者的权利过于集中而导致的权威欺骗,而且能够保证重要数据的安全性和健壮性。 可验证秘密共享允许参与者验证秘密份额的真实性,较好地解决了秘密分发者的欺骗以及参与者之间的相互欺诈问题。在密钥管理、身份认证、电子商务等领域中具有重要的理论意义和实际应用价值。



算法步骤如下

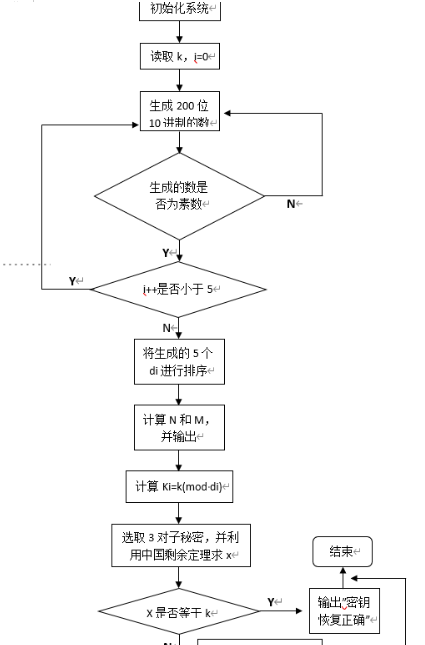




三、方案实现

（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）

算法流程图：



主要函数介绍：

（1）gcd()

函数原型：gcd(a,b)

参数类型：int

功能：用来计算两个大数的最大公约数， 即z=gcd(x,y)。

（2）mutual\_prime\_check()

函数原型：mutual\_prime\_check(list)

参数类型：列表

函数功能：判断列表元素是否互素

（3）find\_mod\_reverse()

函数原型：find\_mod\_reverse(a,m)

参数类型：int

功能：计算模逆

（4）crt()

函数原型：crt(equation\_num,a\_list,m\_list)

参数类型： int,list,list

功能：中国剩余定理解方程。

1. find\_d1()

函数原型：find\_d1(n,t)

参数类型：int

功能：生成随机分割秘密

（6）encrypt()

函数原型：encrypt(k,t,n)

参数类型：int int int

功能：加密

（7）decrypt

函数原型：decrypt(t,k\_d)

参数类型：int int

功能：解密

（8）main()

函数原型：main()

参数类型：主函数

主要代码：

中国剩余定理：

# 中国剩余定律  
def crt(equation\_num: int, a\_list: list, m\_list: list):  
 for i in range(len(a\_list)):  
 print("a", i, ":", a\_list[i])  
 for i in range(len(m\_list)):  
 print("m", i, ":", m\_list[i])  
 if not mutual\_prime\_check(m\_list): # 互素检查  
 print("不能直接利用中国剩余定理")  
 print()  
 return  
 # 求乘积  
 m = prod(m\_list)  
 # 求Mj  
 mj = [m // \_m for \_m in m\_list]  
 # 求Mj的逆  
 inv\_mj = [find\_mod\_reverse(m // \_m, \_m) for \_m in m\_list]  
 xj = [(mj[j] \* inv\_mj[j] \* a\_list[j]) % m for j in range(equation\_num)]  
 res = sum(xj) % m  
 return res

主函数：

def main():  
 t = int(input("请输入t："))  
 n = int(input("请输入n："))  
 d\_seq = find\_d1(n, t)  
 for d\_idx in range(n):  
 print(f"d{d\_idx + 1} = {d\_seq[d\_idx]}")  
 print()  
  
 for case\_i in range(2):  
 print(f"{'#' \* 36} 测试样例【{case\_i + 1}】{'#' \* 36}")  
 with open(f"./secret{case\_i + 1}.txt", "r") as f:  
 k = int(f.read())  
 print("[+]测试加密：", end="\n\n")  
 print("k = ", k)  
 res\_enc = encrypt(k, t, n, d\_seq)  
 for i in range(n):  
 print(f"(k{i + 1}, d{i + 1})：{res\_enc[i]}")  
 print()  
 print("[-]测试解密：", end="\n\n")  
 # 取前t组用于解密（从res\_enc中任取t组均可）  
 res\_dec = decrypt(t, random.sample(res\_enc, t))  
 print("正确秘密k：", k)  
 print("还原秘密k：", res\_dec)  
 if res\_dec == k:  
 print("秘密还原正确！")  
 else:  
 print("秘密还原错误！")

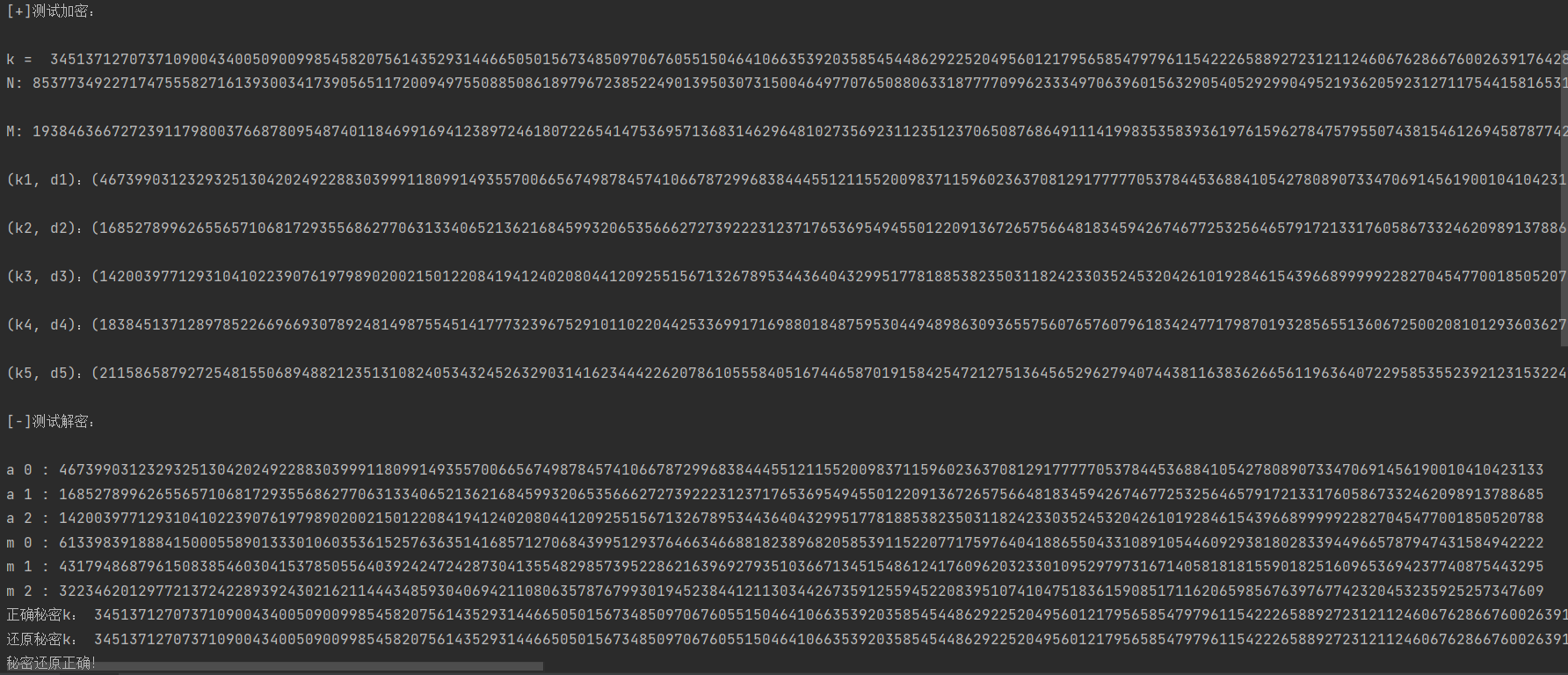
1. 数据分析(包括算法测试数据的分析，运行结果截图等等)

进过计算，样例数据在501位，我们根据输入的n和t，生成d数组，即分割的秘密数组模数，需要满足互素并且前n位大于后n-1位（大致满足即可）

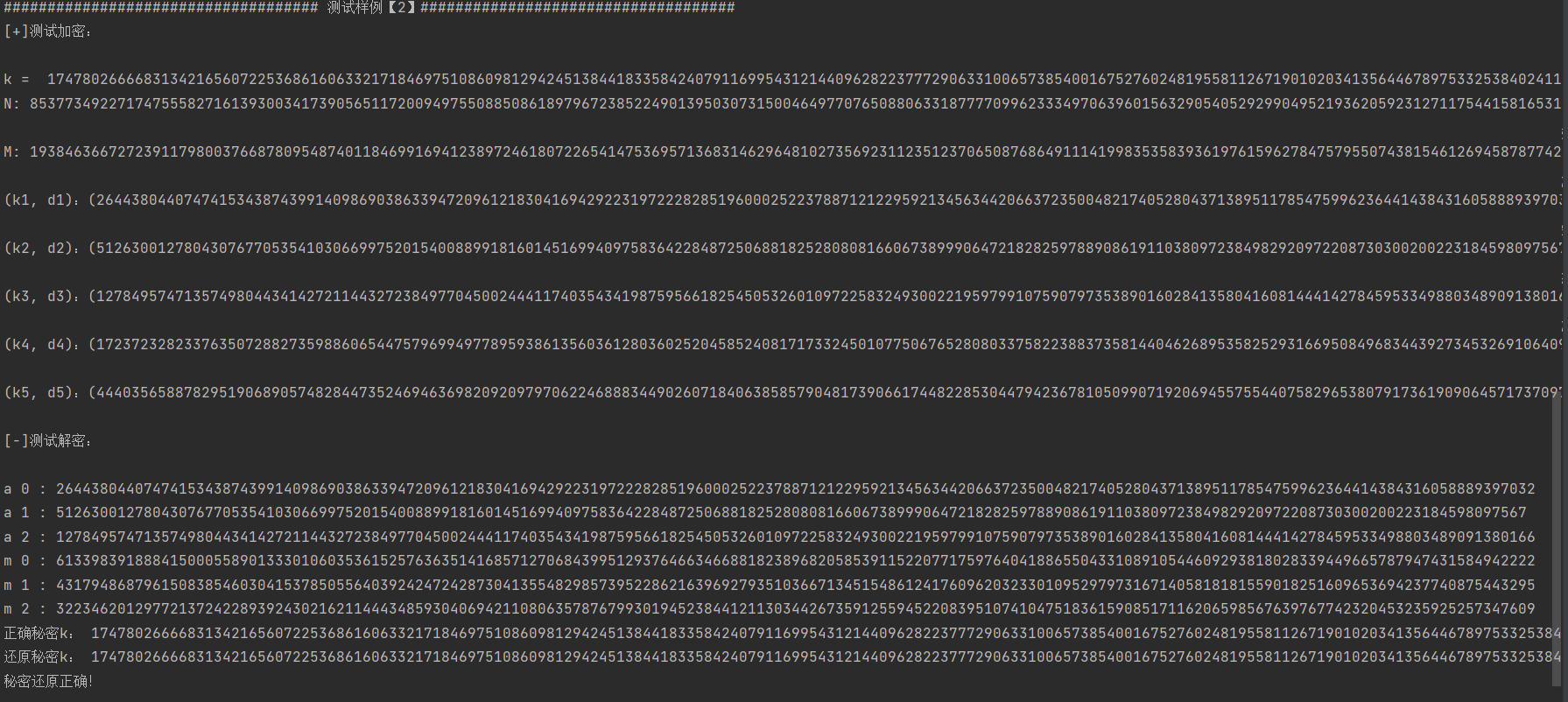
写出代码如下：

# 根据选取的n和t 生成d数组  
def find\_d1(n, t):  
 d = [] # 初始化d数组  
 x = 0  
 for i in range(500):  
 if (i \* t > 502) & (i \* (t - 1) < 502):  
 x = i  
 break  
 print("x:", x)  
 temp = random.randint(pow(10, x), pow(10, x+1))  
 d.append(temp)  
 i = 1  
 while i < n:  
 temp = random.randint(pow(10, x), pow(10, x+1))  
 tem = d.copy()  
 tem.append(temp)  
 if mutual\_prime\_check(tem): # judge1是判断是否互素的函数  
 d.append(temp)  
 i = i + 1  
 return d

样例1：



样例2：



五、思考与总结

1. 在基于中国剩余定理的(*t*, *n*)秘密共享方案中，少于*t*个子秘密，是否能够正确恢复出秘密？请简述原因。

不能，因为如果少于t个子秘密最终会导致，k>最终模数，导致部分损失，从而还原不出原秘密

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？

秘密共享是将秘密以适当的方式拆分，拆分后的每一个子秘密由不同的参与者管理，单个参与者无法恢复秘密信息，只有若干个参与者一同协作才能恢复秘密消息。并且，当其中某些参与者出问题时，秘密仍可以恢复。

本实验中难点在于分割子秘密模数的过程，我们需要根据t和n生成满足要求的一组数：前t个大于后t-1个，两两互素，便随机生成了一组数，进行加密。通过本次实验我体会了密码学与数论的紧密联系,将数论的知识运用于密码学的方案设计中，并且提高了逻辑思维能力与实践能力。

**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3-学号-姓名，例如：实验3-20009200400-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。