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：

姓名：

学号：

日期：

一、实验目的

1.实验环境

- ✧ Windows11，tdm64-gcc-10.3.0，VSCode。

2.实验目标

- ✧ 通过编程实现基于中国剩余定理的（t，n）门限秘密共享方案，加深对于中国剩余定理的理解与运用；
- ✧ 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；
- ✧ 提高逻辑思维能力与实践能力。

二、方案设计

1.背景

在中国剩余定理的基础上，提出了秘密共享的设计方案。将秘密 k 分成 n 个子秘密 k_1，k_2，…，k_n，利用中国剩余定理，使得如果已知任意 t 个 k_i 的值，则很容易恢复出 k，如果已知任意 t-1 个或者更少的 k_i 值，则不能够恢复出 k，以此来达到(t，n)门限秘密共享的方案。

2.原理

✧ Ⅰ.(t,n)门限，一个秘密 k，被分隔成 n 个子秘密(k_i,d_i)

对于某个秘密 k，计算$\begin{cases} k_1 \equiv k(mod\ d_1) \\ k_2 \equiv k(mod\ d_2) \\ \qquad \dots \\ k_n \equiv k(mod\ d_n) \end{cases}$，则子秘密为$(k_i,d_i)$

其中对于 d_i 具有一定的要求:

- ◆ 1)选择 n 个整数 d_1,d_2，…，d_n，满足：
  - (1)$d_1 < d_2 < \dots < d_n$；d_i 严格递增
  - (2)$(d_i,d_j) = 1$，i≠j；d_i 两两互素
  - (3)$N=d_1\times d_2\times\cdots\times d_t$，$M=d_{n-t+2}\times d_{n-t+3}\times\cdots\times d_t$，有 N>M
- ◆ 2)N>k>M

✧ Ⅱ.n 个子秘密中任意选择 t 个，(k_{i_1},d_{i_1})，(k_{i_2},d_{i_2})，…， (k_{i_t},d_{i_t})，恢复出秘密 k：

计算$\begin{cases} x \equiv k_{i_1}(mod\, d_{i_1}) \\ x \equiv k_{i_2}(mod\, d_{i_2}) \\ \quad\ldots \\ x \equiv k_{i_t}(mod\, d_{i_t}) \end{cases}$，恢复出秘密$x \equiv k(mod\, N_1)$，$N_1 = d_{i_1}d_{i_2}\ldots d_{i_t}$

✧ Ⅲ.任选 t-1 个(d_{j_1}, k_{j_1})，$\left(d_{j_2}, k_{j_2}\right)$，…，$(d_{j_{t-1}}, k_{j_{t-1}})$：

$x \equiv k(mod\, M_1)$，$M_1 = d_{j_1}d_{j_2}\ldots d_{j_{t-1}}$

t 个子秘密能恢复出秘密，t-1 个子秘密不能。$N_1 > N > k > M > M_1$

✧ Ⅳ.中国剩余定理

设正整数 m_1，m_2，…，m_k 两两互素，对任意整数 a_1，a_2，…，a_k，一次同余方程$\begin{cases} x \equiv a_1(mod\, m_1) \\ x \equiv a_2(mod\, m_2) \\ \quad\ldots \\ \quad\ldots \\ x \equiv a_k(mod\, m_k) \end{cases}$在模 m 意义下有唯一解，该解可表示为

$$x \equiv M_1M_1^{-1}a_1 + M_2M_2^{-1}a_2 + \ldots + M_kM_k^{-1}a_k(mod\, m)$$

其中 $m=m_1m_2\cdots m_k$，$M_j=m/m_j$，$M_jM_j^{-1}\equiv 1(mod\ m_j)$，$j=1,2,\cdots,k$

3.算法步骤

a.随机生成 5 个 200 位的素数作为选择的d_i,将其放入列表 list 并升序排列，输出 list

b.计算 N，M 并输出

c.读入文件中 500 位的秘密 k

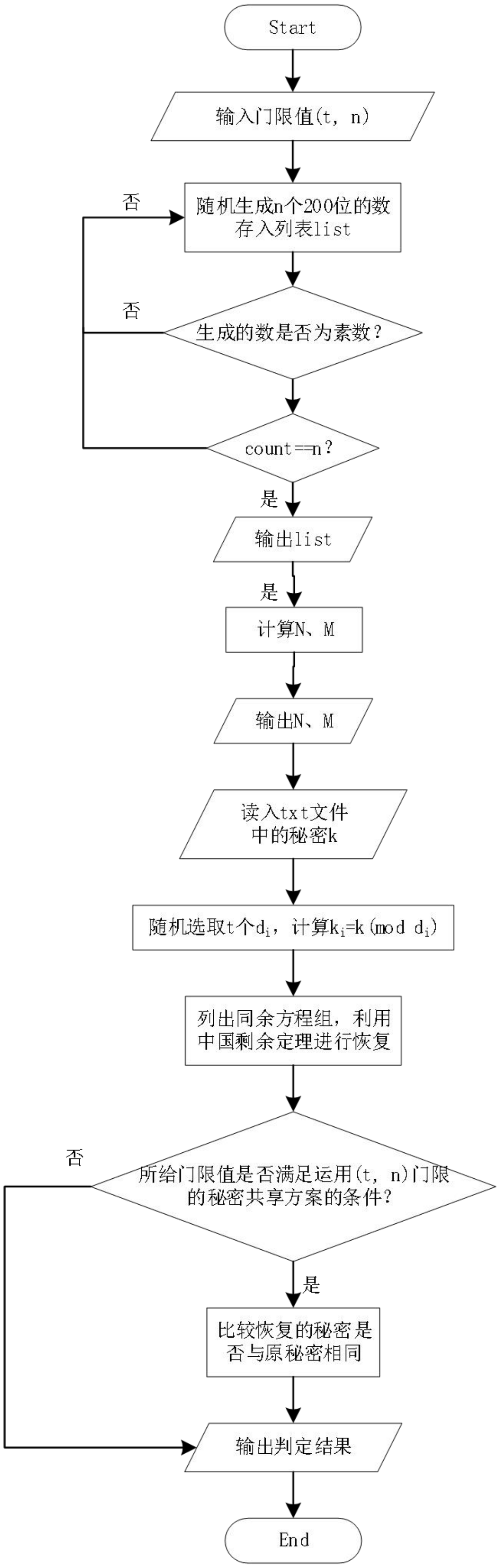
d.随机选取 3 个 d_i，计算 $k_i=k(mod\ d_i)$

e.列出同余方程组，利用中国剩余定理进行恢复

f.比较恢复的秘密是否与原秘密相同

三、方案实现

1.算法流程图


Start
输入门限值(t, n)
随机生成n个200位的数
存入列表list
否
生成的数是否为素数?
否
count==n?
是
输出list
是
计算N、M
输出N、M
读入txt文件
中的秘密k
随机选取t个d_i，计算k_i=k(mod d_i)
列出同余方程组，利用
中国剩余定理进行恢复
否
所给门限值是否满足运用(t, n)门限
的秘密共享方案的条件?
是
比较恢复的秘密是
否与原秘密相同
输出判定结果
End

2. 主要函数介绍

a. gcd(a, b)：求 a 和 b 的最大公约数，若结果为 1，则两数互质

```
def gcd(a, b):
    if a < b:  # 调整大小顺序
        a, b = b, a
    else:
        pass
    while b != 0:
        return gcd(b, a % b)
    return a
```

b. compare(list)：判断一个列表里的数是否两两互质，互质则满足 CRT 的使用条件

```
def compare(list):
    for i in range(0, len(list)):
        for j in range(i + 1, len(list)):
            if gcd(list[i], list[j]) != 1:
                print('不能直接利用中国剩余定理')
                exit()
```

c. product_m(list)：求出输入的 $m_1, m_2, \cdots, m_k$ 的乘积即 m

```
def product_m(list):
    m = 1
    for i in list:
        m *= i
    return m
```

d. get_divsion(list, m)：求 M_1, M_2， . . . ， M_k 的值，其中 $M_j=m/m_j$

```
def get_divsion(list, m):
    div = []
    for i in list:
        div.append(m // i)
    return div
```

e. get_inverse(a, m)：先求数 a 的逆，再求其模 m 的值

```
def get_inverse(a, m):
    if gcd(a, m) != 1:
        return None
    u1, u2, u3 = 1, 0, a
    v1, v2, v3 = 0, 1, m
    while v3 != 0:
        q = u3 // v3
```

```
        v1, v2, v3, u1, u2, u3 = (u1 - q * v1), (u2 - q * v2),
(u3 - q * v3), v1, v2, v3
    return u1 % m
```

f. get_x(M: int, M_inverse: int, a: int, m: int)：求 X_j，算法为：X_j = (M * M_inverse * a) % m_j

```
def get_x(M: int, M_inverse: int, a: int, m: int):
    product_x = (M * M_inverse * a) % m
    return product_x
```

g. get_solution(list_m, list_a)：算出最终答案 X = $X_1+X_2+\cdots+X_k$

```
def get_solution(list_m, list_a):
    compare(list_m)
    m = product_m(list_m)

    list_M = get_divsion(list_m, m)

    list_M_inverse = []
    list_X = []
    total = 0

    for i in range(0, len(list_M)):
        list_M_inverse.append(get_inverse(list_M[i],
list_m[i]))

    for i in range(len(list_M)):
        list_X.append(get_x(list_M[i], list_M_inverse[i],
list_a[i], m))

    for x in list_X:
        total += x

    return total % m
```

h. compare_(list, x)：检验 list 中的任一元素是否与 x 互素，若满足则返回 1

```
def compare_(list, x):  # compare_()函数 传入参数为一个列表
list 和一个数据 x
    for i in range(0, len(list) - 1):
        if gcd(list[i], x) == 1:  # 共进行 len(list)-1 次比较
            continue  # 若最大公约数为 1 ==>互素
        else:  # 否则不互素，返回 0
            return 0
```

```
    return 1  # x 与list 中任意一个元素都互素(除了本身)，则返回1
```

3. 算法实现主要代码

```
t, n = map(int, input("请输入门限值（以空格分隔）：").split())
while True:  # 死循环
    count = 1  # 用来记录已经产生了满足条件的随机数个数
    list = []
    list.append(random.randint(10**200, 10**201))  # 先产生一个随机数list[0]，并调用append()内置函数存入list 列表中
    for i in range(1, n):  # 再产生新的随机数
        list.append(random.randint(10**200, 10**201))
        if compare_(list, list[i]) != 1:
            break  # 新的随机数必须与前面产生的随机数列表中所有元素都互素
        else:
            count += 1  # 满足所有条件则将count 加1
    if count == n:  # 若产生了n 个满足条件的随机数个数，跳出最外围的死循环
        break
    else:  # 否则继续执行，直到产生n 个满足条件的大数
        continue
list.sort()  # di 升序排列
for i in range(0, len(list)):  # 用来输出产生的数据
    print('d', i, ':', list[i])
# 后面就是从list 列表中挑出t 个数据
# 列出一次同余方程组，调用CRT
N, M = 1, 1
for i in range(0, t):
    N *= list[i]
for i in range(n-t+1, n):
    M *= list[i]
print('N=', N)
print('M=', M)
with open('secret2.txt', 'r', encoding='utf-8') as f:
    k = int(f.read())
d = random.sample(list, t)  # 随机抽取t 个di
d.sort()
list_a = []  # ki
list_m = []  # di
for i in range(0, len(d)):
    list_a.append(k % d[i])
for i in range(0, len(d)):
```

```python
    list_m.append(d[i])
recovered_k = get_solution(list_m, list_a)
if N > k & k > M:
    print('还原得到的秘密为：', recovered_k)
    if (recovered_k == k):
        print('秘密还原正确！')
    else:
        print('秘密还原不正确！ ')
else:
    print(t, '个子秘密不能恢复出秘密！')
```

四、数据分析

1. 输入读取“secret1.txt”

```
问题    输出    JUPYTER    调试控制台    终端
PS C:\User[redacted]> python -u "c:\Use
r[redacted]"
请输入门限值（以空格分隔）：3 5
d 0 : 157181141720396934875592406379579526841375099459550621093914442605674697564399870276376726386768031016426967130378374894108814466664415241935208231636129070514280550275541765087233187489822298245749022
7
d 1 : 50945248167329409716020228942538291915269810251994586242843401593781398462837958634620567768708650805753944961034004108370053260294959947362712858596352887063093633746272204791666151956287685810598300
3
d 2 : 72687787430084841075357532395391832478555457721027353509413173298293054546019475288380730361193781028586104573314157853489360023425591343607714995751875441761291305770741911248893076075106854627739347
9
d 3 : 91138713169157487048829007779802575526745727020821502712067466239356154394584376807111736123078167375423412918568093232394549649140758390239415646063395960410738897873696789469334212331212837536240636
1
d 4 : 940443344424518657872050666459809896638107365313475443275915812644690843152497677203657641831719812348948838454153616131300577786014513737443750384613583845238871843710752134861882494665095872989444602
8
N= 58205707241776542487657206955074079414688192648361129759283317082063928998631740050804082603175125582368915873397427586099349461459507111301756947976509428256260057186355668304675934477262114087200008829328812141719339394602123034401055399921167495403182564034541220104633090412929736510661622813660036623560510572600333365795196508635027072791081527034809409247101861470179538791344123882883168984608425851199802530485885590830188875642633769466043657695094562481033278117928829904812255862146659867711922158342843446306133860269134067127432812129272876037704195719262247750155578079653358950220628199
M= 857107962193493889775939691435565731153677834653987574196857714734369708912459232170166594657796112307070814581489127770631363772819360502360147462583147835506947001903100828041389026438086625699593669163054102488971458928137095452359456080882380187047275221512567605261935590839560184234286598524078050437800171009708025434662517851401919756742566561991678418588084012609972528247894707072599118384108
还原得到的秘密为： 34513712707371090043400509009985458207561435293144665050156734850970676055150464106635392035854544862922520495601217956585479796115422265889272312112460676286676002639176428018428110501603858423653391913799566813168343496159580747338826480697660522927181825310390154470363345038219192650075737129895819591216748784152267758174110791876279446873375543750375483617833468974263170064305100280485303368221268987883665789183494148190239289344532645805573919693262044556121484486211305720303265273494061665
秘密还原正确！
PS C:\User[redacted]>
```

2. 输入读取“secret2.txt”

```
问题    输出    JUPYTER    调试控制台    终端
PS C:\Users[redacted]\实验3-
基于CRT的(t,n)门限秘密共享方案"
PS C:\Users[redacted] python -u "c:\User
s[redacted]"
请输入门限值（以空格分隔）：3 5
d 0 : 1193194404878822652504457722195989037013617080344179178606583616861631313338122011234813681627082117893155106142443315633495979238766586687303721324347508800579102829561472058211632523722437955127994444989529560381414517771182785168697l
d 2 : 774666749776296102684237269574529754317727904308914577357123037180271340606538215116832908669724319501094580915060597965745269268280328786553080459069456143525989705495325119977462951386345917552367903
d 3 : 797559971355606021882877810098812560366185447726439814193550086924544252602555833443031277657553420518708271863325024851502828787695284671359530479046050637057212703547214899711077477160855744530643363
d 4 : 876177874920555671877204348937450160485895522546642441424868121160680917060179617558490650983659670050159050072010502639597327198242953799510835250414852481557198714212037672618053199784935596333585l209
N= 549847850940232777577747843245133795889348323875643742266841885460208605487439516687028515054153952569777998246972707859636291682765661682680234071951798723892429764082794855543984274925634715326944337516032510073392821213417361288658872620604401751599833190845370172450542135614236530056234333134024535654027417829171249712129888104411296207187916472801927269047178816151759175827444830478401389138425212453239835770329477657416523867711398879459670336054906250248248869382427619140855532762399976148335350300030906876932053281653904287691240288029563306470117811414303954535798175899916459290573751972
M= 698804400824054137528561113151752364667746662816219488264426087243577338437220671981592134547235889713754315456411983619909721189430120141605691543622273178599962549264519157050268801567513285255604930151226251284636771437804957739433426954523622873131818811717366271967730416024171145608067735821757027410164885366883521506247737340254128627183368222940119250642323532956670214235776865635280011375867
还原得到的秘密为： 17478026666831342165607225368616063321718469751086098129424513844183358424079116995431214409628223777290633100657385400167527602481955811267190102034135644678975332538402411252204599772483146656099060144969961437110041474027934832041351813583159652034964431257857287075228427404692047523442305548984024079911564277739000132521183888591292001268001699186206357791036971642713788106200272066299890847864165018226431988593865501936053961710499420525798158420768632815038342235167689001130965331812773662
秘密还原正确！
PS C:\Users[redacted]>
```

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。

   答：不能。任选 t 个子秘密进行恢复，计算出来的模数为 N_1，而任取 t-1 个子秘密进行恢复，其模数为 M_1，$M_1 = d_{j_1}d_{j_2}...d_{j_{t-1}}$，由之前所选择的 d_i 要求可以知道，$N_1 > N > k > M > M_1$，因此模 M_1 得出的结果并非 k。

2. 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？

   a. 随机生成五个大数时不知道如何调用随机函数，上网搜索后解决问题；

   b. 更加熟练使用 python 中的 list 类型，提升了编程能力；

   c. 对中国剩余定理的理解更加深刻，感受到数论的奇妙。