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：

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 ElGamal 公钥加密算法，加深对于离散对数的理解；
- ✧ 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

ElGamal 公钥加密算法是由 Tather ElGamal 在 1985 年提出的，它是一种基于离散对数难题的加密体系，与 RSA 算法一样，既能用于数据加密，也能用于数字签名。RSA 算法是基于因数分解，而 ElGamal 算法是基于离散对数问题。与 RSA 算法相比，ElGamal 算法哪怕是使用相同的私钥，对相同的明文进行加密，每次加密后得到的签名也各不相同，有效的防止了网络中可能出现的重放攻击。

2.原理

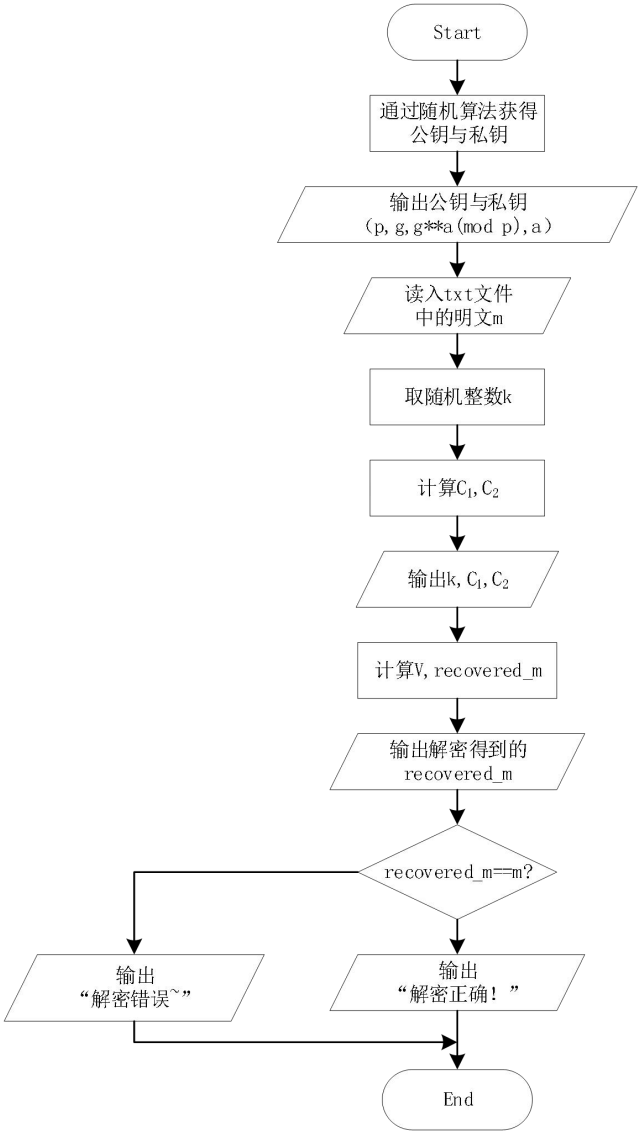
- ✧ 密钥产生过程：
  - ➢ (1)随机产生一个大素数 p 及模 p 的一个原根 g；
  - ➢ (2)随机选取整数 a，计算$g^a(\mathrm{mod\ p})$。
- ✧ Alice 的公钥是$(\mathrm{p,g},g^a)$，私钥是 a。
- ✧ 加密过程：Bob 将明文消息 m 加密成密文
  - ➢ (1)随机选取一个整数 k，$1 \le \mathrm{k} \le \mathrm{p}-2$；
  - ➢ (2)计算$\mathrm{C_1} \equiv \mathrm{g}^k(\mathrm{mod\ p})$，$\mathrm{C_2} \equiv \mathrm{m} \cdot (\mathrm{g}^a)^k(\mathrm{mod\ p})$
  - ➢ (3)将密文$(\mathrm{C_1},\mathrm{C_2})$发送给 Alice。
- ✧ 解密过程: Alice 将密文$(\mathrm{C_1},\mathrm{C_2})$恢复为明文 m
  - ➢ (1)计算 $\mathrm{V} \equiv \mathrm{C_1^a}(\mathrm{mod\ p})$；
  - ➢ (2）计算 $\mathrm{m} \equiv C_2V^{-1}(\mathrm{mod\ p})$。
- ✧ 验证：$C_2V^{-1}(\mathrm{mod\ p}) \equiv \frac{C_2}{C_1^a} \equiv \frac{m\cdot(g^a)^k}{(g^k)^a} \equiv m$

3.算法步骤

- ✧ 第一步：取一个随机的素数 q，并且这个素数满足 p=2q+1 也是一个素数，取出数 q 之后，利用费马素性检验算法检测其是不是素数即可
- ✧ 第二步：随机取一个数 g(在[2,p-2]之间),判断 g**2(mod p)与 g**q(mod p)是否为 1；只要有一个为 1，说明 g 不是 p 的原根，就重新取随机数
- ✧ 第三步：私钥 a 根据要求来取，产生公钥与私钥
- ✧ 第四步：利用公钥进行加密
- ✧ 第五步：利用私钥进行加密
- ✧ 第六步：输出要求要展示的数

三、方案实现

1.算法流程图



2. 主要函数介绍

a. rabin_miller(num)：rabin 算法，检测大整数是否为素数

```
def rabin_miller(num):
    s = num - 1
    t = 0
    while s % 2 == 0:
        s = s // 2
        t += 1
    for trials in range(5):
        a = random.randrange(2, num - 1)
        v = pow(a, s, num)
        if v != 1:
            i = 0
            while v != (num - 1):
                if i == t - 1:
                    return False
                else:
                    i = i + 1
                    v = (v**2) % num
    return True
```

b. get_root()：获取原根

```
def get_root():
    while True:
        list = []

        q = get_prime()  # 获取一个素数
        p = 2 * q + 1
        if not Isprime(p, 5):  # 判定p是否为素数，不是就结束此
次循环，开始下一次循环
            continue

        a = random.randint(2, p - 1)  # 获取一个随机整数，随机
数范围为 2 ~ p - 1
        if quick_algorithm(a, 2, p) == 1 or quick_algorithm(
                a, q, p) == 1:
            # 若a**2(mod p)或a**q(mod p)等于1，那么说明，这个
a 不是原根
            continue

        else:
            list.append(p)
```

```
            list.append(a)
            return list
```

c. get_Ga()：求 g**a(mod p)，其中 a 是一个随机整数

```
def get_Ga():
    list_key = get_root()

    a = random.randint(10**100, 10**101)  # 获得a
    Ga = quick_algorithm(list_key[1], a, list_key[0])  # 调用Format中的quick_algorithm()函数

    list_key.append(a)  # 初始化list_key
    list_key.append(Ga)

    return list_key
```

3. 算法实现主要代码

```
# 将公钥与私钥输出 公钥序数:
list_key = get_Ga()
print('公钥:\np = {}\ng = {}\ng**a(mod p) = {}\n私钥:\na = {}'.format(list_key[0], list_key[1], list_key[2], list_key[3]))

# Bob用公钥对明文m进行加密
# Bob选取一个随机的整数k
# C1(mod p) = g**k(mod p)
# C2(mod p) = (m*((g**a)**k))(mod p)

with open(r'secret2.txt', 'r', encoding='utf8') as file:
    m = int(file.read())

k = random.randint(2, list_key[0] - 2)  # 取一个随机的整数 k
C_1 = quick_algorithm(list_key[1], k, list_key[0])
C1 = C_1 % list_key[0]  # 计算C1
C2 = m * quick_algorithm(list_key[1], list_key[3] * k, list_key[0]) % list_key[0]  # 计算C2
print('k =', k)
print('C1 = {}\nC2 = {}'.format(C1, C2))

# Alice解密过程
# V = C1**a(mod p)
# m = C2*V_inverse (mod p)
# V_inverse = get_inverse(v,p)
```

```
V = quick_algorithm(C1, list_key[3], list_key[0])
V_inverse = get_inverse(V, list_key[0])

m = C2 * V_inverse % list_key[0]
recovered_m = m % list_key[0]
print('解密结果为:', recovered_m)
if recovered_m == m:
    print('解密正确！')
else:
    print('解密错误~')
```

四、数据分析

1. 输入读取“secret0.txt”

```
162  with open(r'secret0.txt', 'r', encoding='utf8') as file:
163      m = int(file.read())
```

问题 输出 JUPYTER 调试控制台 终端 Code

```
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验> cd "c:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法"
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法> python -u "c:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法\实验4-20009201350-刘倩.py"
公钥：
p = 4953718919378878600564464342458284530249533520534955879033283817977636401958371667649075568754745262014328410728443201561925354882041745502123264697367
g = 3222145698761236119566938786763440252455821598625120291857748159057738700461122741982801623632690991537078012956749755411656085761126745909976157432938
g**a(mod p) = 1609087973094054536598273403412869425226461839291385798578645706013345688807031965725709449449344627875
私钥：
a = 2251271078875376159128523796809509127867921015286534376942358720155830043135323268251287520640499924280591324122761986218390687649116692582613528381050
k = 4108680911489572654440819291471512486137515332682650859223982517442701842380655571902074137381418153529949121155134034972565992605999639862796853526 3
C1 = 4951408691851891187268580311967370241320743373914792467510770568564652901201570405326662079532867709608569778763762767126210461888705387209441651696271
C2 = 5650291625892534668608118366721336903528482567802756133394143208096495942010748367144501349719762681761828674349788166646013326932893808243160155941 52
解密结果为： 9327260388393076415930260479153046010064951650867096323260782111903507989221223155043829435161867334962529353992935468294479465522637777146290777
解密正确！
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法>
```

2. 输入读取“secret1.txt”

```
162  with open(r'secret1.txt', 'r', encoding='utf8') as file:
163      m = int(file.read())
```

问题 输出 JUPYTER 调试控制台 终端 Code

```
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验> cd "c:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法"
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法> python -u "c:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法\实验4-20009201350-刘倩.py"
公钥:
p = 388076904473962673200710299943273054328939623396031683878487621237300809722313562248769402303279711173854117646953293225955655125560457841193408041615 9
g = 186051095254843186163333054420774301160943602723798030902872515590697269311970914600918583495189632402372302086174856616251648025397857203354002852767 7
g**a(mod p) = 858039431058561073381497169385445571325936564753783143073133825529665746745735661136713893374634577 11
私钥:
a = 286841648505811873744744625038538635171153316625817311910495011803677988265417733002964386509083779765139378197715878983152088371017954355863837994712 9
k = 383510775966918756401824721368117149794629894682617399958034254953161889806483704664337298236093418023691678536012255338315532114657275602193303119 07
C1 = 250524765656354359144607953950151896703013394193831961614147164595271125673772871994937553059971590740901513593081470963840629160915080412560483641675 8
C2 = 198590494284601209126106084269167736354140688775803714355233455285775009483537479198656683112655228786367822702747334881929795447041238884192683621157 6
解密结果为: 268934047525129207430358090155831774406988263017537886266459743124401877032780923870438637078936998897356703572131607069480172048042746
解密正确!
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法>
```

3. 输入读取“secret2.txt”

```
162  with open(r'secret2.txt', 'r', encoding='utf8') as file:
163      m = int(file.read())
```

问题 输出 JUPYTER 调试控制台 终端 Code

```
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验> cd "c:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法"
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法> python -u "c:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法\实验4-20009201350-刘倩.py"
公钥:
p = 405812401330933937674924118005098140965464690944278109402002879261937344987133980119439371364453175885246159018127963865959779519612385702023833213934 3
g = 243063792959585383231199327834312350817486655147715480404136968188265438017366534007485196651296553393427674424627809221627197940942034396341727530856 7
g**a(mod p) = 440838009263214994113537198874208070756719054624058108658194244603963713868330946316035402808402023 09
私钥:
a = 225521985658662139477171644908874207832484573465935614365904660844943812877342463859977990650698253358305310164486033619473840700881898736661559493287 8
k = 382362529685396045540367561659367318399422474652490933984154817638704606103111149243705075163303628088964112621111977139123259095963852764517924112515 0
C1 = 724717520645329797779608038481063225209465706260050516727844275875399188665263989215843970735941757630293540080745207001267559174695047512516974511462
C2 = 231207977792455365128906344275844234964668633853257437743839973559896014849544766116030814409194901146407169297813469643940812336906502255623918903246 6
解密结果为: 534401672873304897938057492233840777465042901882059083916536793014854580823268775560583768951905931050906323409422304889226 02
解密正确!
PS C:\Users\Qian Liu\Desktop\3-1\信安密码学实验\实验4-ElGamal公钥密码算法>
```

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？。

答：如果使得$g^m \equiv 1(\mathrm{mod\ p})$成立的最小正幂 m 满足 $\mathrm{m} = \varphi(\mathrm{p})$，则称 g 是 p 的本原根。 其中$\varphi(\mathrm{n})$为欧拉函数。搜索发现求解本原根较为简单的一个算法为 DH 算法。若 g 满足$g^2(mod\ p) \neq 1$ 且$g^q(mod\ p) \neq 1$，则说明是 g 是 mod p 的一个原根。

DH 算法步骤如下：

- ✓ 生成一个大素数 q，使得 p=2q+1 也是素数
- ✓ 生成一个随机数 g，1<g<p-1，直到$g^2(mod\ p)$和$g^q(mod\ p)$都不等于 1
- ✓ 得到 g 是 p 的本原根

2. 如果k与$p-1$不互素，可能会发生什么情况？

答：加密过程为$\mathrm{C_1} \equiv \mathrm{g}^k(\mathrm{mod\ p})$，$\mathrm{C_2} \equiv \mathrm{m} \cdot (\mathrm{g}^a)^k(\mathrm{mod\ p})$，原根 g 满足的条件是$g^{p-1} \equiv 1(mod\ p)$。

当 p=2*q+1 时，p-1 的因子只有 2 和 q，如果 k 与 p-1 不互素，q 有很大可能也是 k 的因子。

攻击者可以根据公钥$(\mathrm{p,g},g^a)$中的 p 值求出 q，从而化简加密过程为$\mathrm{C_1} \equiv \mathrm{g}^{k/q}(\mathrm{mod\ p})$，$\mathrm{C_2} \equiv \mathrm{m} \cdot (\mathrm{g}^a)^{k/q}(\mathrm{mod\ p})$，从而降低了求解离散对数的困难程度，加大了明文泄露的风险，减弱了算法的安全性。

3. 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. 不知道如何求原根，上网搜索后解决问题；

b. 更加熟练使用 python，提升了编程能力；

c. 对 ElGamal 密钥算法的理解更加深刻，感受到数论的奇妙。