|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 实验（2） | 实验名称 | | Shamir(t,n)密钥共享方案程序实现 | |
| 学时数 | 4时数 | | 实验类型 | V验证性 V设计性 □综合性 |
| 每组人数 | 3人 | | 难度 | 中等 |
| 覆盖知识点 | | 密钥管理，密钥分布式存储与管理 | | |
| 实验内容：   1. Shamir(t,n)密钥共享方案的程序实现 2. 共享秘密的生成与分发仿真 3. 共享密钥的重现验证   实验开展方式：   1. 以组为单位进行实验，3人1组 2. 课内外结合的方式完成实验，课内完成实验内容、实验要求等讲解，以及实验结果、实验报告检查；课内外结合完成程序设计、编制与调试   实验软硬件环境要求：  单机环境下即可实现,需要编程环境，编程语言不限制，可采用C、VC、C++、Java、python等  实验讲授内容：   1. Shamir(n,t)密钥共享方案的原理 2. 共享秘密计算的方法 3. 共享密钥重现的计算方法，拉格朗日插值方法   学生实践内容：   1. 完成shamir(n,t)密钥共享方案的程序实现 2. 进行秘密共享的分发仿真 3. 进行密钥重现验证 4. 编写实验报告 | | | | |

**《信息安全概论》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | |  | | | **姓名** |  |
| **实验题目** | Shamir(t,n)密钥共享方案程序实现 | | | | | |
| **实验时间** | **2021.4.23** | | **实验地点** | **DS3 305** | | |
| **实验成绩** |  | | **实验性质** | **□验证性 □设计性 □√综合性** | | |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | | | | |
| 一、实验目的   1. 理解Shamir(n,t)密钥共享方案的原理 2. 掌握共享秘密计算的方法 3. 掌握共享密钥重现的计算方法，拉格朗日插值方法 | | | | | | |
| 二、实验项目内容   1. Shamir(t,n)密钥共享方案的程序实现 2. 共享秘密的生成与分发仿真 3. 共享密钥的重现验证 | | | | | | |

|  |
| --- |
| 三、实验过程或算法（源程序）算法   1. 输入参数k，t，n，分别表示要加密的数，shamir(t,n)的参数      k = int(input("Input K "))      t = int(input("Input t "))      n = int(input("Input n "))  2. 输入参数dt，表示解密时的人数：      dt = int(input("Input new t to decode "))  **3. 进行加密**  （1）首先生成随机的大素数和多项式系数  （2）生成用户持有秘钥的X  （3）根据f(x) 生成用户持有秘钥的Y  def encode(k, t, n):      p, A = generate\_a(k, t)      print('A', A)      x\_list = list(range(1, n + 1))      print('X', x\_list)      y\_list = []      for x in x\_list:          y\_list.append(h(A, x, p))  return p, x\_list, y\_list  P为素数，首先生成范围在[k, k+1000]的所有素数，然后随机取值。  A为小于p的数，在[0, p-1]之间不放回的取样（不重复）。  def generate\_a(k, t=4):      # 1. create p      primes\_list = get\_primes(k + 5, k + 1000)      p = primes\_list[random.randint(0, len(primes\_list))]      # 2. generate  t-1 a      A = random.sample(range(0, p), t - 1)      A.insert(0, k)      return p, A  素数生成如下：  def get\_primes(kmin, kmax):      prime\_list = []      for i in range(kmin, kmax):          if prime(i): prime\_list.append(i)      return prime\_list  def prime(x):      if x == 0: return False      if x == 1: return False      if x % 2 == 0: return False      i = 3      while i < x \*\* 0.5 + 1:          if x % i == 0: return False          i += 2      return True  **4. 进行解密**    解密可按照如下公式直接计算如下h(0)    def decode(p, xx, yy, t):      global tmp      hx = 0      for s in range(t):          tmp = yy[s]          down = 1          for j in range(t):              if j == s:                  continue              # tmp = tmp \* (-xx[j]) % p              tmp = tmp \* (-xx[j]) % p              down \*= xx[s] - xx[j]          # tmp = tmp \* ext\_gcd(down, p)[0] % p          inv = ext\_gcd(down, p)[0]          tmp = (tmp % p \* inv % p) % p          # tmp = tmp / down          hx = (hx + tmp) % p      # return hx % p      return hx 备注 以上所有计算模p，对于加法、减法、乘法，基于模p运算的规律进行。  对于除法，转化为乘法逆元计算。  使用扩展欧几里得算法求逆元：  def ext\_gcd(a, b):      """扩展欧几里得算法      求a关于b的逆元      Args:          a (int): 数a          b (int): 数b      """      if b == 0:          return 1, 0, a      else:          x, y, gcd = ext\_gcd(b, a % b)  # 递归直至余数等于0          x, y = y, (x - (a // b) \* y)  # 辗转相除法反向推导每层a、b的因子使得gcd(a,b)=ax by成立          return x, y, gcd  将除法 a/b mod(p) 转化为 a\*inv(b,p)%p |
| **四、实验结果及分析和（或）源程序调试过程**  if \_\_name\_\_ == '\_\_main\_\_':      k = int(input("Input K "))      t = int(input("Input t "))      n = int(input("Input n "))      dt = int(input("Input new t to decode "))      # k = 2018      # t, n = 10, 15      p, xx, yy = encode(k, t=t, n=n)      # p = 2281      # xx = list(range(0, n))      # yy = [49, 31, 60, 635, 761, 528, 1604, 392, 1328, 1585, 703, 2146, 1867, 1277, 1704]      comp\_cases = []      for i, (x, y) in enumerate(zip(xx, yy)):          print(f'P\_{i + 1} ({x},{y})')          comp\_cases.append((x, y))      ans = decode(p, xx, yy, t=dt)      print("prime", p)      print('decode k', ans) （一）正确解密 参数(K, t, n, dt) = (20184376, 25, 30, 28)   （二）错误解密 参数(K, t, n, dt) = (20184376, 25, 30, 23) |