密码学综合设计实验

实验 4：RSA 加密解密算法实现

学号：031803108

姓名：苏煜程

2019 年　10 月　22 日

RSA加密解密算法实现

实验要求

- 实现寻找指定2进制位数范围的素数
  - 算法采用爱拉托斯散筛法或Miller-Rabin概率检测法
  - 输入要找的素数位数长度范围：最低的至少要5bit长，最高的16bit长
  - 输出随机找到的符合位数范围的素数
  - 提示，C语言实现的同学，素数用unsigned long表示，若编译器支持，可以用unsigned long long表示素数，这时位长范围可放宽到32bit
- 实现RSA密钥产生算法
  - 算法参见课本
  - 选取两个大素数时要注意两个大素数的二进制表示位数和不超过16bit（这样可以确保两个大素数乘积的二进制位数不超过16位,从而保证幂乘运算中间结果不超过32位）。若用unsigned long long表示素数，限制可以放宽到32bit
  - 输出公钥(e,n),私钥(d,n)
- 快速指数算法实现
  - 输入：a，m，n

    输出：d = am mod n
  - a,m,n,d都是unsigned long型或unsigned long long型, a<n
- RSA加密和解密算法的实现
  - 输入明文分组M。明文分组M类型是unsigned long型或unsigned long long型。注意，M<n。输入直接以数值形式输入
  - 实现RSA加密，输出密文
  - 实现RSA解密，输出密文
- 附加内容
  - 实现位数超过32bit的整数的加减乘除运算。提示：利用数组
  - 实现对明文进行分组，分组长度不超过n的长度(分组长度小于log2n)

实验原理

1. 密钥的产生

（1） 选取两个大素数 p 和 q ；

（2） 计算 n=p×q ，φ(n)=(p－1)(q－1)，其中 φ(n)是 n 的欧拉函数值；

（3） 随机选一整数 e,满足 1< e < φ(n)，且 gcd(φ(n),e)=1。因而在模φ(n)下，e 有逆元；

（4） 计算 d，满足 $d*e \equiv 1$ (mod φ(n))；

（5） 取公钥为{ e，n }，秘钥为{ d，n }；

（6） p，q 不再需要，可以销毁。

2. 加密

加密时首先将明文比特串分组，使得每个分组对应的十进制数小于 n，即分组长度小于$\log_2 n$。然后对每个明文分组 m，作加密运算：

$$c \equiv m^e \ (mod\ n)$$

3. 解密

对密文分组的解密运算为：

$$m \equiv c^d \ (mod\ n)$$

软件系统设计

模N大数的幂乘的快速算法

```
def fastExpMod(b, e, m):  # 底数，幂，大数N
    result = 1
    e = int(e)
    while e != 0:
        if e % 2 != 0:  # 按位与
            e -= 1
            result = (result * b) % m
            continue
        e >>= 1
        b = (b * b) % m
    return result
```

生成大素数

```
def create_prime_num(length):
    while True:
        n = random.randint(0, length)
        if n % 2 != 0:
            found = True
            for i in range(0, 10):
                if miller_rabin_test(n):
                    pass
                else:
                    found = False
                    break
            if found:
                return n
```

素性检测

```
def miller_rabin_test(n):  # p为要检验得数
    p = n - 1
    r = 0
    while p % 2 == 0:  # 最后得到为奇数的p(即m)
        r += 1
        p /= 2
    b = random.randint(2, n - 2)
    # 如果情况1    b得p次方  与1  同余  mod n
    if fastExpMod(b, int(p), n) == 1:
        return True
    # 情况2  b得（2^r  *p）次方  与-1 (n-1) 同余  mod n
    for i in range(0,20):
        if fastExpMod(b, (2 ** i) * p, n) == n - 1:
            return True
    return False
```

密钥生成

- 随机在（1，fn）选择一个E， 满足gcd（e,fn）=1

```
def selectE(fn, halfkeyLength):
    while True:
        # e and fn are relatively prime
        e = random.randint(0, fn)
        if math.gcd(e, fn) == 1:
            return e
```

- 根据选择的e，匹配出唯一的d

  ```
  def match_d(e, fn):
      d = 0
      while True:
          if (e * d) % fn == 1:
              return d
          d += 1
  ```

- 密钥生成

```
def create_keys(keylength):
    p = create_prime_num(keylength / 2)
    q = create_prime_num(keylength / 2)
    n = p * q
    fn = (p - 1)*(q - 1)
    e = selectE(fn, keylength / 2)
    d = match_d(e, fn)
    return (n, e, d)
```

- 加密与解密

```
def encrypt(M, e, n):
    return fastExpMod(M, e, n)

def decrypt( C, d, m):
    return fastExpMod(C, d, m)

def display():
    print("__________________RSA__________________")
    print("1.encrypt")
    print("2.decrypt")
    print("q.quit")
    print("Enter the key you wanna try")
    print("_____________________________________")

def encrypt_file():
    f = input("Please input your message: ")
    mess = f
    n, e, d = create_keys(1024)
    print("请妥善保管私钥（解密时需要用到）：（n:",n," ,d:",d,")")
    s = ''
    t = ""
    #s = encrypt(int(mess), e, n)
    print(mess)
    for ch in mess:
        c = chr(encrypt(ord(ch), e, n))
        t += str(hex(encrypt(ord(ch), e, n)))
        s += c
    print("Encrypt hex message :",t)
    print("Encrypt message :"+s)
    print("Encrypt Done!")

def decrypt_file():
    f = input("Please input your encrypt message: ")
    mess = f
    #n,d = input("私钥: ")
    n,d= map(int, input("输入您的私钥（n,d）:").split())
    s = ''
    for ch in mess:
        c = chr(decrypt(ord(ch), d, n))
        s += c
    print(s)
    print("Decrypt Done!")
```

实现效果

```
__________________RSA__________________
1.encrypt
2.decrypt
q.quit
Enter the key you wanna try
________________________________________
>>>1
Please input your message: abcdefg
请妥善保管私钥（解密时需要用到）：（n: 23927  ,d: 20417 )
abcdefg
Encrypt hex message : 0x4b970x43750x4d6e0x49ea0x569c0x3ed80xf1f
Encrypt message :騑羚黝隡嚜珅。
Encrypt Done!
```