アルゴリズム論特講 （塩田）

2006 年 5 月 25 日の課題

課題

- 関数定義部 crypto060525.py と RSA 暗号の暗号化・復号化雛形プログラム hina0605025.py をダウンロードせよ。
- hina0605025.py の鍵生成部分を完成せよ。
- 法のビット数を取り替えて実行し、反復２乗法と単純乗算との計算量の違いを実感せよ。

提出期限：未定 （ 512 号室ポストまで ）

雛形

```
#!/bin/env python
#
# hina060525.py

from sys import *
from math import *
from random import *
from crypto060525 import *    # crypto060525.py を引用

########## RSA 暗号の暗号化関数・復号化関数 ##########
# 反復２乗法を用いた場合
# 関数 powermod の定義は crypto060525.py を見よ

# RSA 暗号の公開鍵 e,n を用いて平文 x を暗号文に変換する
def encrypt(x,e,n):
    return powermod(x,e,n)

# RSA 暗号の公開鍵 n と秘密鍵 d を用いて暗号文 y を復号する
def decrypt(y,d,n):
    return powermod(y,d,n)

########## RSA 暗号の暗号化関数・復号化関数 ##########
# 反復２乗法を用いない愚かな場合

# RSA 暗号の公開鍵 e,n を用いて平文 x を暗号文に変換する
def slowencrypt(x,e,n):
    z = 1
    while e > 0:
        z = modmul(x,z,n)
        e = e-1
    return z

# RSA 暗号の公開鍵 n と秘密鍵 d を用いて暗号文 y を復号する
def slowdecrypt(y,d,n):
    z = 1
```

```
    while d > 0:
        z = modmul(y,z,n)
        d = d-1
    return z

############### ここからメインルーチン ###############

print
print '-----'
k = input('素因子 p のビット数を指定してください ( 1 以下で終了 ) : ')
if k < 2:
    exit(0)

# 素数 p の生成（約 k ビット）
p = randbit(k)
while solovaystrassen(p)==0:
    p+=1

# p と異なる素数 q の生成 （約 1.1*k ビット）
q = randbit(round(1.1*k))
while solovaystrassen(q)==0 or p==q:
    q+=1

# n の生成 ( p と q の積 )
n = ##### ここを埋めよ #####

# m の生成 ( n のオイラー関数 )
m = ##### ここを埋めよ #####

# 暗号化指数 e の生成 ( m と互いに素な乱数 )
e = rand(m)
while gcd(e,m) != 1:
    e = rand(m)

# 復号化指数 d の生成 ( mod m での e の逆数 )
d = modinv(e,m)

# 鍵の表示
print
print '公開鍵 :'
print 'n = %d ( %d ビット )' % (n,bitlen(n))
print 'e = %d' % e
print
print '秘密鍵 :'
print 'p = %d' % p
print 'q = %d' % q
print 'm = %d' % m
print 'd = %d' % d

# 暗号化・復号化テスト
print
print '暗号化・復号化テスト（反復２乗法を用いた場合):'
print
for i in range(0,3):
    x = rand(n)
    y = encrypt(x,e,n)
    z = decrypt(y,d,n)
    print '   平文  ',x
    print '-> 暗号文',y
    print '-> 復号文',z
    print
```

```
print 'Hit Any Key'
stdin.readline()

print '暗号化・復号化テスト（反復２乗法を用いない場合):'
print
for i in range(0,3):
    x = rand(n)
    y = slowencrypt(x,e,n)
    z = slowdecrypt(y,d,n)
    print '   平文  ',x
    print '-> 暗号文',y
    print '-> 復号文',z
    print
```

実行例

```
素因子 p のビット数を指定してください（ 1 以下で終了 ）: 10

公開鍵 :
n = 1371443 ( 21 ビット )
e = 853837

秘密鍵 :
p = 733
q = 1871
m = 1368840
d = 528013

暗号化・復号化テスト（反復２乗法を用いた場合):

   平文   1255499
-> 暗号文 852
-> 復号文 1255499

   平文   284571
-> 暗号文 774286
-> 復号文 284571

   平文   1299873
-> 暗号文 1361884
-> 復号文 1299873

Hit Any Key

暗号化・復号化テスト（反復２乗法を用いない場合):

   平文   848019
-> 暗号文 29032
-> 復号文 848019

   平文   290168
-> 暗号文 882217
-> 復号文 290168

   平文   152858
-> 暗号文 1000981
-> 復号文 152858
```