数論と代数　課題

0312017142　村尾侑哉

RSA公開鍵暗号系

コード:python

```
from sympy import nextprime
from math import gcd
import random
import math

#問題(1)1.公開鍵(n)を求める

#任意の相異なる素数p,qを問題文の条件のもと、選ぶ
a = 42 % 30 + 10 #問題(1)1.(a)より
b = 13 + 10 #問題(1)1.(b)より

p = nextprime(a) #p>aを満たす最小の素数
q = nextprime(b)
if q == p:#もしqがpと同値であれば次の素数を選ぶ
    q = nextprime(q)
print("[pとqを出力して確認]")
print("p=",p,",","q=",q) #確認: pとqの値

n = p * q #nを求める
print("n=q*q =",n) #確認:n

#最小公倍数lcm(p-1,q-1)を求めて、Lと互いに素でLより小さな任意の整数eを選ぶ
def lcm(p,q):   #最小公倍数を求める関数(Lの計算過程)
    return (p*q) // gcd(p,q)    #lcm(a,b) = ab / gcd(a,b)より

#問題(1)2.公開鍵(暗号鍵):eを生成する
L = lcm(p-1,q-1) #最小公倍数を求める
for i in range(2,L): #1<e<Lとなるiを生成
    if gcd(i,L) == 1: #互いに素ならeとする
        e = i
        break

print("[Lがp-1,q-1で割り切れているか確認します(0がならば割り切れている)]")
print("L % (p - 1) =",L % (p - 1) ,",","L % (q - 1)=",L % (q - 1)) #確認: 0が出力されれば、Lがp-1,q-1で割り切れている

#問題(1)3.秘密鍵(復号鍵):dを生成する
for i in range(2,L): #1<i<Lとなるiを生成
    if(e * i) % L == 1: # e*d≡1(mod L)のとき、dにiを代入する
        d = i
        break
```

```
if(d < 0): #負数になったときの処理
    d += L

print("ed≡1(mod L)となるd:",d) #確認：dの値
print("検算:(e*d) % L=",(e*d) % L) #確認：(e*d) % Lをして、1になればOK


#問題(2)

# Mは2つ用意する
# M：10<M<nの間でランダムに決める
M1 = random.randrange(10+1,n-1)
M2 = random.randrange(10+1,n-1)

print("Mの1つ目：",M1,"Mの2つ目",M2,"(どちらも10<M<n(",n,")である)") #2つのMの確認

#暗号化(暗号文の生成)
C1 = M1 ** e % n
C2 = M2 ** e % n

print("～暗号化(暗号文生成)～")
print("M1^e =",M1,"^",e,"=",M1 ** e )
print("M2^e =",M2,"^",e,"=",M2 ** e )
print("暗号文C1 = M1^e % n=",C1,",暗号文C2 = M2^e % n =",C2)

# 復号操作(平文の生成)
M_1 = C1 ** d % n
M_2 = C2 ** d % n

print("～復号操作(平文生成)～")
print("C1^d =",C1,"^",d,"=",C1 ** d )
print("C2^d =",C2,"^",d,"=",C2 ** d )
print("平文M_1 = C1^d % n =",M_1)
print("平文M_2 = C2^d % n =",M_2)

print("～まとめ～")
print("暗号鍵(公開):e=",e,"n=",n)
print("復号鍵:d=",d,"n=",n,"(dが秘密情報)")
print("暗号化:E")
print("復号操作:D")
print("暗号文C=E(M)=M^e mod n")
print("平文M=D(C)=C^d mod n")
print("前述の操作で、(1つ目の)平文→暗号文→平文：M1→C1→M_1：",M1,"→",C1,"→",M_1)
print("前述の操作で、(2つ目の)平文→暗号文→平文：M2→C2M_2：",M2,"→",C2,"→",M_2)
```

出力結果:

```
[pとqを出力して確認]
p= 23 , q= 29
n=q*q = 667
[Lがp-1,q-1で割り切れているか確認します(0がならば割り切れている)]
```

```
L % (p - 1) = 0 , L % (q - 1)= 0
ed≡1(mod L)となるd: 103
検算:(e*d) % L= 1
Mの1つ目:  61 Mの2つ目 43 (どちらも10<M<n( 667 )である)
～暗号化(暗号文生成)～
M1^e = 61 ^ 3 = 226981
M2^e = 43 ^ 3 = 79507
暗号文C1 = M1^e % n= 201 ,暗号文C2 = M2^e % n = 134
～復号操作(平文生成)～
C1^d = 201 ^ 103 =
169509449831383714859529472355105178559606572924399889815602332202031728580434281742946022279792855653220017814113241108183778688148082345139007199452365790332841812375567502560131992159027333452531149250642185287665839584971385501814060 1
C2^d = 134 ^ 103 =
123536198111073910632359233696300789410340426479401488917990294846329381136524159710006981148594488886516011525636226791913809061179980912767934161063843396516884670190024223181919875789561305958664360401485985512580710 4
平文M_1 = C1^d % n= 61
平文M_2 = C2^d % n = 43
～まとめ～
暗号鍵(公開):e= 3 n= 667
復号鍵:d= 103 n= 667 (dが秘密情報)
暗号化:E
復号操作:D
暗号文C=E(M)=M^e mod n
平文M=D(C)=C^d mod n
前述の操作で、(1つ目の)平文→暗号文→平文：M1→C1→M_1:  61 → 201 → 61
前述の操作で、(2つ目の)平文→暗号文→平文：M2→C2M_2:  43 → 134 → 43
```

参考サイト

1.
【Python入門】素数の生成・判定プログラムを実装してみよう！ _ 侍エンジニア塾ブログ _ プログラミング入門者向け学習情報サイト
[https://www.sejuku.net/blog/74038](2018/12/20参照)
2.
Python で公開鍵暗号アルゴリズム RSA を実装してみる - Qiita
[https://qiita.com/QUANON/items/e7b181dd08f2f0b4fdbe](2018/12/20参照)
3.
Pythonでランダムな小数・整数を生成するrandom, randrange, randintなど
[https://note.nkmk.me/python-random-randrange-randint/](2018/12/20参照)