




BUUCTF RSA题目全解1

原创

宁嘉  于 2020-07-04 15:27:18 发布  11111  收藏 76

分类专栏: [RSA加密](#) 文章标签: [密码学](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/MikeCoke/article/details/105967809>

版权



[RSA加密 专栏收录该内容](#)

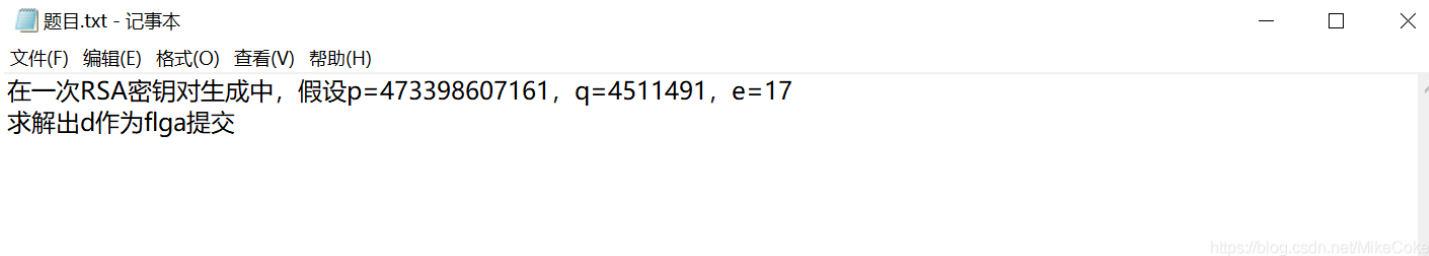
12 篇文章 9 订阅

订阅专栏

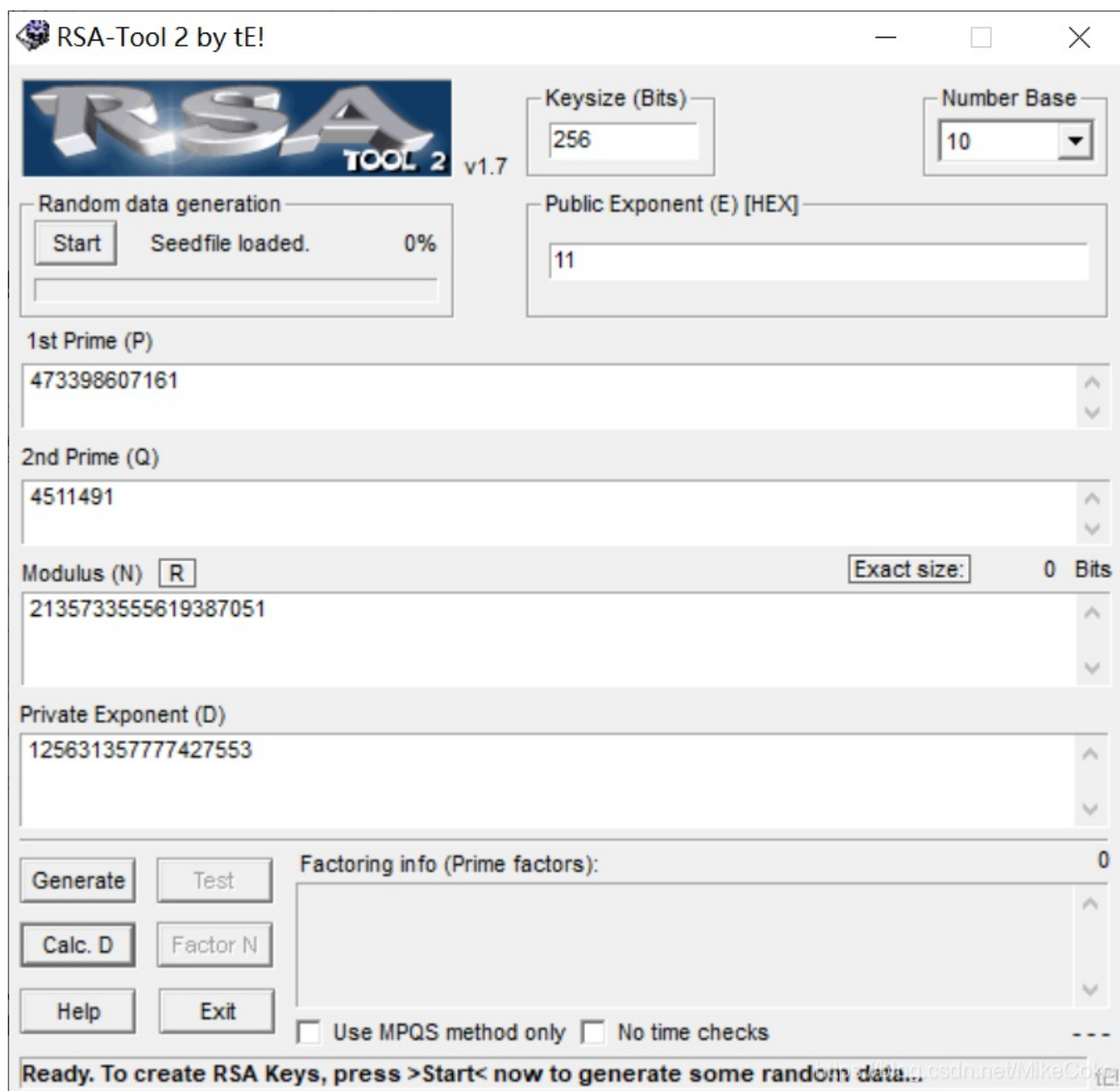
RSA write up

- 1.RSA
- 2.rsarsa
- 3.RSA1
- 4.RSA2
- 5.RSA3
- 6.RSA
- 7.RSAROLL
- 8.[BJDCTF 2nd]rsa0
- 9.Dangerous RSA
- 10.rsa2
- 11.RSA5
- 12.[GUET-CTF2019]BabyRSA

1.RSA

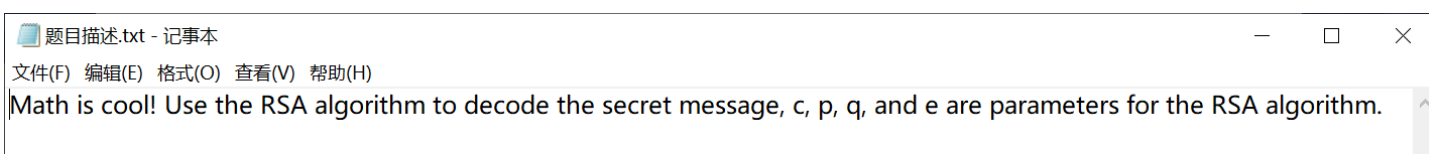


直接用工具RSA Tool2点击这篇文章有介绍怎么用



得到flag{12563135777427553}

2.rsarsa



p =
96484230290105156765905517400104265349457376392357398006439893520398525072984913995610350091634270
50370107570733633350911691280297777160200625281665378483
q =
11874843837980297032092405848653656852760910154543380907650040190704283358909208578251063047732443
992230647903887510065547947313543299303261986053486569407
e = 65537
c =
83208298995174604174773590298203639360540024871256126892889661345742403314929861939100492666605647
31664657648652621745700637684228086972858172674640158370589994176821413874225968933484073563355305
38876418476511737762518202930872128856701803674068074067659236389731613758173927377478327627516901
04423869019034

Use RSA to find the secret message

第 1 行, 第 1 列 100% Unix (LF) https:// UTF-8 Line/MikeCoke

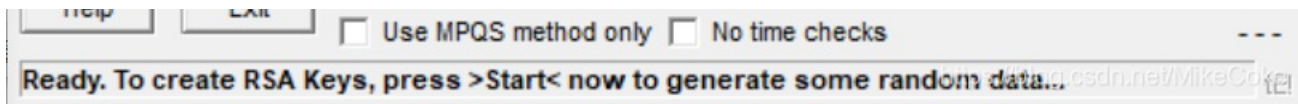
也可以直接用工具RSA Tool2:

注意: 题目中的e是十进制, 在RSA Tool2中要改为16进制, 用RSA Tool2求出私钥d, 再用python函数pow()即可求出flag

The screenshot shows the RSA-Tool 2 interface with the following settings and generated values:

- Keysize (Bits):** 256
- Number Base:** 10
- Public Exponent (E) [HEX]:** 10001 (indicated by a red arrow)
- 1st Prime (P):** 9648423029010515676590551740010426534945737639235739800643989352039852507298491399561035009163427050370107570733633350911691280297777160200625281665378483
- 2nd Prime (Q):** 11874843837980297032092405848653656852760910154543380907650040190704283358909208578251063047732443992230647903887510065547947313543299303261986053486569407
- Modulus (N) [R]:** 114573516752272714750064227635008832737477859608443481000717283425702025029279291376859256856603741797722497252841363753834114679306784379319341824813349417007577541466886971550474580368413974382926969910999462429631003527365143148445405716553105750
- Private Exponent (D):** 566320475711906605675203410288611948624114284168625070347625872299951386056498369602206199034563927521159432993353851632162337446246238488742353033096363934467363472386277930227252609864669579747530041292106804014323774449841951450098019673911966155 (indicated by a red arrow)

Buttons: Generate, Test, Calc. D, Factor N, Help, Exit. Factoring info (Prime factors): 0



```
e = 65537
p = 964842302901051567659055174001042653494573763923573980064398935203985250729849139956103500916342705037010757
0733633350911691280297777160200625281665378483
q = 118748438379802970320924058486536568527609101545433809076500401907042833589092085782510630477324439922306479
03887510065547947313543299303261986053486569407
n = p*q
#密文
C = 832082989951746041747735902982036393605400248712561268928896613457424033149298619391004926666056473166465764
8652621745700637684228086972858172674640158370589994176821413874225968933484073563355305388764184765117377625182
0293087212885670180367406807406765923638973161375817392737747832762751690104423869019034

d = 566320475711906605675203410288611948624114284168625070347625872299951386056498369602206199034563927521159432
9933538516321623374462462384887423530330963639344673634723862779302272526098646695797475300412921068040143237744
4984195145009801967391196615524488853620232925992387563270746297909112117451398527453977

#求明文
M = pow(C, d, n) #快速求幂取模运算
print(M)
```

3.RSA1

已知dp,dq

一切不思考的抄代码，都是耍流氓，[点击这篇文章](#)，去解读dp,dq

```
p = 863763376725700856709965348654109117132049150943361544753916243791124417588566780639841179052408355344515811
3502227745206205327690939504032994699902053229
q = 126406749739964727691760479371708834209270508214800105815931371353724738805956137373376306297525773461470392
84030082593490776630572584959954205336880228469
dp = 65007957022168346211090423511932615306500438410562529309309496633586250168818328407280660261502646930761093
54874099841380454881716097778307268116910582929
dq = 78347226367355344901953258038647067238057403355130388913791176043888168367455609809825679567351220196300217
5438762767516968043599582527539160811120550041
c = 247223054038873820735673164676490806626315529059602293990791079956021544181760563358006388875276141640735304
3765708507967615735020535194522298935131607648657359957604197833987226592506276431853608900731027027852615967893
7431903862892400747915525118983959970607934142974736675784325993445942031372107342103852
```

上代码：

```

p = 863763376725700856709965348654109117132049150943361544753916243791124417588566780639841179052408355344515811
3502227745206205327690939504032994699902053229
q = 126406749739964727691760479371708834209270508214800105815931371353724738805956137373376306297525773461470392
84030082593490776630572584959954205336880228469
dp = 65007957022168346211090423511932615306500438410562529309309496633586250168818328407280660261502646930761093
54874099841380454881716097778307268116910582929
dq = 78347226367355344901953258038647067238057403355130388913791176043888168367455609809825679567351220196300217
5438762767516968043599582527539160811120550041
c = 247223054038873820735673164676490806626315529059602293990791079956021544181760563358006388875276141640735304
3765708507967615735020535194522298935131607648657359957604197833987226592506276431853608900731027027852615967893
7431903862892400747915525118983959970607934142974736675784325993445942031372107342103852

```

```

import gmpy2
I = gmpy2.invert(q, p)
mp = pow(c, dp, p)
mq = pow(c, dq, q)          #求幂取模运算

m = (((mp-mq)*I)%p)*q+mq    #求明文公式

print(hex(m))              #转为十六进制

```

至于flag是16进制转文本还是10进制数，我忘记了，自己试一下吧。

4.RSA2

dp泄露

[点击查看数学原理](#)

上代码：

```

import gmpy2 as gp

e = 65537
n = 248254007851526241177721526698901802985832766176221609612258877371620580060433101538328030305219918697643619
8142009306796121098855338013353484450237516704784370730555447242806847332980515991676603036451831461614974853586
33681492129668802402065797789905550489547645118787266601929429724133167768465309665906113
dp = 90507449805234690464302513287951833069192517457305400462187725331868267505542197094355201669552856036483444
6303196939207056642927148093290374440210503657

c = 140423670976252696807533673586209400575664282100684119784203527124521188996403826597436883766041879067494280
9574102019589357373603808018454538292939974334141888387257517962617026220285872115603533628471910603065785105113
80965162133472698713063592621028959167072781482562673683090590521214218071160287665180751

for i in range(1, e):          #在范围(1, e)之间进行遍历
    if (dp*e-1)%i == 0:
        if n%(((dp*e-1)//i)+1) == 0:    #存在p, 使得n能被p整除
            p = ((dp*e-1)//i)+1
            q = n//(((dp*e-1)//i)+1)
            phi = (q-1)*(p-1)           #欧拉定理
            d = gp.invert(e, phi)       #求模逆
            m = pow(c, d, n)            #快速求幂取模运算

print(m)                        #10进制明文
print('-----')
print(hex(m)[2:])               #16进制明文
print('-----')
print(bytes.fromhex(hex(m)[2:])) #16进制转文本

```

5.RSA3

共模攻击

[点击了解数学原理](#)

```
c1=2232203527566323704164689377045193350932470191348430333807621060354261275895626286964082248647012114942448557
1361007421293675516338822195280313794991136048140918842471219840263536338886250492682739436410013436651161720725
8554848666900847887213495556620198790815011132229961233055330093259643777988927031615218528059568112195638833128
9633015629862167468435391954755812792092570684280891476219901105495581653497767526739500957534782038707348392842
506653636148277489237096952074030428745655508933372782327506569010772537497541764311429052216291198932092617792
645253901478910801592878203564861118912045464959832566051361
n=22708078815885011462462049064339185898712439277226831073457888403129378547350292420267016551819052430779004755
8466490440010241414852832864831307026160572746984736111495087988697063475019315831176327107007872280164801276773
9364992953041659868602735421642256593445901516192761360790283154285797785961259628235367932777330372700440726219
7231586324599181983572622404590354084541788062262164510140605868122410388090174420147752408554129789760902300898
0462739090078528184740307706996476473630151021189567376739413542176926960449696953085064365731425655734875835070
37356944848039864382339216266670673567488871508925311154801
e1=11187289
c2=1870201004518701555654869164239498283566926214723021273130993867522645855521042597242941844927341053538798593
1036711854265623905066805665751803269106880746769003478900791099590239513925449748814075904017471585572848473556
4905654500626647064491284158347879619472662597897859629222387011340797204142284140661930714953046123410529874556
1593002353682380149926977335718608745274750084064041936501155442118303750565346128673274098370274082267114804561
9497667184586123657285604061875653909567822328914065337797733444640351518775487649819978262363617265797982843179
630888729407238496650987720428708217115257989007867331698397
e2=9647291
```

上代码:


```

import libnum
from gmpy2 import invert
# 欧几里得算法
def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b % a, a)
        return (g, x - (b // a) * y, y)

def main():
    n = 2270807881588501146246204906433918589871243927722683107345788840312937854735029242026701655181905243077900
4755846649044001024141485283286483130702616057274698473611149508798869706347501931583117632710700787228016480127
6773936499295304165986860273542164225659344590151619276136079028315428579778596125962823536793277733037270044072
6219723158632459918198357262240459035408454178806226216451014060586812241038809017442014775240855412978976090230
0898046273909007852818474030770699647647363015102118956737673941354217692696044969695308506436573142565573487583
507037356944848039864382339216266670673567488871508925311154801
    c1 = 223220352756632370416468937704519335093247019134843033380762106035426127589562628696408224864701211494244
8557136100742129367551633882219528031379499113604814091884247121984026353633888625049268273943641001343665116172
0725855484866690084788721349555662019879081501113222996123305533009325964377798892703161521852805956811219563883
3128963301562986216746843539195475581279209257068428089147621990110549558165349776752673950095753478203870734839
284250665363614827748923709695207403042874565550893337278232750656901077253749754176431142905221629119893209261
7792645253901478910801592878203564861118912045464959832566051361
    c2 = 187020100451870155565486916423949828356692621472302127313099386752264585552104259724294184492734105353879
8593103671185426562390506680566575180326910688074676900347890079109959023951392544974881407590401747158557284847
3556490565450062664706449128415834787961947266259789785962922238701134079720414228414066193071495304612341052987
4556159300235368238014992697733571860874527475008406404193650115544211830375056534612867327409837027408226711480
4561949766718458612365728560406187565390956782232891406533779773344464035151877548764981997826236361726579798284
3179630888729407238496650987720428708217115257989007867331698397
    e1 = 11187289
    e2 = 9647291
    s = egcd(e1, e2)
    s1 = s[1]
    s2 = s[2]
    # 求模反元素
    if s1 < 0:
        s1 = -s1
        c1 = invert(c1, n)
    elif s2 < 0:
        s2 = -s2
        c2 = invert(c2, n)

    m = pow(c1, s1, n) * pow(c2, s2, n) % n
    print(libnum.n2s(m))

if __name__ == '__main__':
    main()

```

运行得到flag

6.RSA



7.RSAROLL

BUUCTF RSAROLL

8.[BJDCTF 2nd]rsa0

[BJDCTF 2nd] RSA0

9.Dangerous RSA


```
problem.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
#n:
0x52d483c27cd806550fbe0e37a61af2e7cf5e0efb723dfc81174c918a27627779b21fa3c851e9e94188eaae3d5cd6f752406a43fbecb53e
80836ff1e185d3ccd7782ea846c2e91a7b0808986666e0bdadbfb7bdd65670a589a4d2478e9adcafe97c6ee23614bcb2ecc23580f4d2e3c
c1ecfec25c50da4bc754dde6c8bfd8d1fc16956c74d8e9196046a01dc9f3024e11461c294f29d7421140732fedacac97b8fe50999117d279
43c953f18c4ff4f8c258d839764078d4b6ef6e8591e0ff5563b31a39e6374d0d41c8c46921c25e5904a817ef8e39e5c9b71225a83269693e
0b7e3218fc5e5a1e8412ba16e588b3d6ac536dce39fcdfce81eec79979ea6872793L
#e: 0x3
#c:0x10652cdfaa6b63f6d7bd1109da08181e500e5643f5b240a9024bfa84d5f2cac9310562978347bb232d63e7289283871efab83d84ff5
a7b64a94a79d34cfbd4ef121723ba1f663e514f83f6f01492b4e13e1bb4296d96ea5a353d3bf2edd2f449c03c4a3e995237985a596908adc
741f32365
so,how to get the message?
```

低加密指数攻击:

所谓低加密指数指的就是e非常小的情况下,通常为3。

这种题目通常有两种类型,一种直接爆破,另外一种为低指数广播攻击。

1.先介绍比较简单的情况。假设e=3,e很小,但是n很大。

回顾RSA加密公式: $C = M^e \% n$ (C密文, M明文)

1. 当 $M^e < n$ 时,

$C = M^e$, 所以对C开方就能得到M

2. 当 $M^e > n$ 时, 此时用爆破的方法

假设我们 M^e / n 商 k 余数为c,

所以 $M^e = k*n + C$, 对K进行爆破, 只要k满足 $k*n + C$ 能够开方就可以

Python3脚本:

```

from gmpy2 import iroot
import libnum
n = 0x52d483c27cd806550fbe0e37a61af2e7cf5e0efb723dfc81174c918a27627779b21fa3c851e9e94188eaae3d5cd6f752406a43fbec
b53e80836ff1e185d3ccd7782ea846c2e91a7b080898666e0bdadbfb7bdd65670a589a4d2478e9adcafe97c6ee23614bcb2ecc23580f4d2
e3cc1ecfec25c50da4bc754dde6c8bfd8d1fc16956c74d8e9196046a01dc9f3024e11461c294f29d7421140732fedacac97b8fe50999117d
27943c953f18c4ff4f8c258d839764078d4b6ef6e8591e0ff5563b31a39e6374d0d41c8c46921c25e5904a817ef8e39e5c9b71225a832696
93e0b7e3218fc5e5a1e8412ba16e588b3d6ac536dce39fcdfce81eec79979ea6872793

c = 0x10652cdfaa6b63f6d7bd1109da08181e500e5643f5b240a9024bfa84d5f2cac9310562978347bb232d63e7289283871efab83d84ff
5a7b64a94a79d34cfbd4ef121723ba1f663e514f83f6f01492b4e13e1bb4296d96ea5a353d3bf2edd2f449c03c4a3e995237985a596908ad
c741f32365

k = 0
while 1:
    res=iroot(c+k*n,3)
    if(res[1]==True):
        print(libnum.n2s(int(res[0])))
        break
    k=k+1

...

```

第二种写法

```

import gmpy2
from libnum import*
n = 0x52d483c27cd806550fbe0e37a61af2e7cf5e0efb723dfc81174c918a27627779b21fa3c851e9e94188eaae3d5cd6f752406a43fbec
b53e80836ff1e185d3ccd7782ea846c2e91a7b080898666e0bdadbfb7bdd65670a589a4d2478e9adcafe97c6ee23614bcb2ecc23580f4d2
e3cc1ecfec25c50da4bc754dde6c8bfd8d1fc16956c74d8e9196046a01dc9f3024e11461c294f29d7421140732fedacac97b8fe50999117d
27943c953f18c4ff4f8c258d839764078d4b6ef6e8591e0ff5563b31a39e6374d0d41c8c46921c25e5904a817ef8e39e5c9b71225a832696
93e0b7e3218fc5e5a1e8412ba16e588b3d6ac536dce39fcdfce81eec79979ea6872793

c = 0x10652cdfaa6b63f6d7bd1109da08181e500e5643f5b240a9024bfa84d5f2cac9310562978347bb232d63e7289283871efab83d84ff
5a7b64a94a79d34cfbd4ef121723ba1f663e514f83f6f01492b4e13e1bb4296d96ea5a353d3bf2edd2f449c03c4a3e995237985a596908ad
c741f32365

i = 0
while 1:
    if(gmpy2.iroot(c+i*n,3)[1]==1):      #开根号
        print(gmpy2.iroot(c+i*n,3))
        break
    i=i+1

...

```

得到 **flag{25df8caf006ee5db94d48144c33b2c3b}**

10.rsa2

得到一份py文件，观察e,n可以知道这是一道低解密指数攻击



```
34f80ba9-8d9a-43cb-b3e3-8283c50e62b3.py - C:\Users\MIKEWY...
File Edit Format Run Options Window Help
N = 1019918097775532534702767513992647401311576823292526735017921545070061584344
e = 4673191956326572130710518041030251867667613550973799291262509297684907526219

import hashlib
flag = "flag{" + hashlib.md5(hex(d)).hexdigest() + "}"

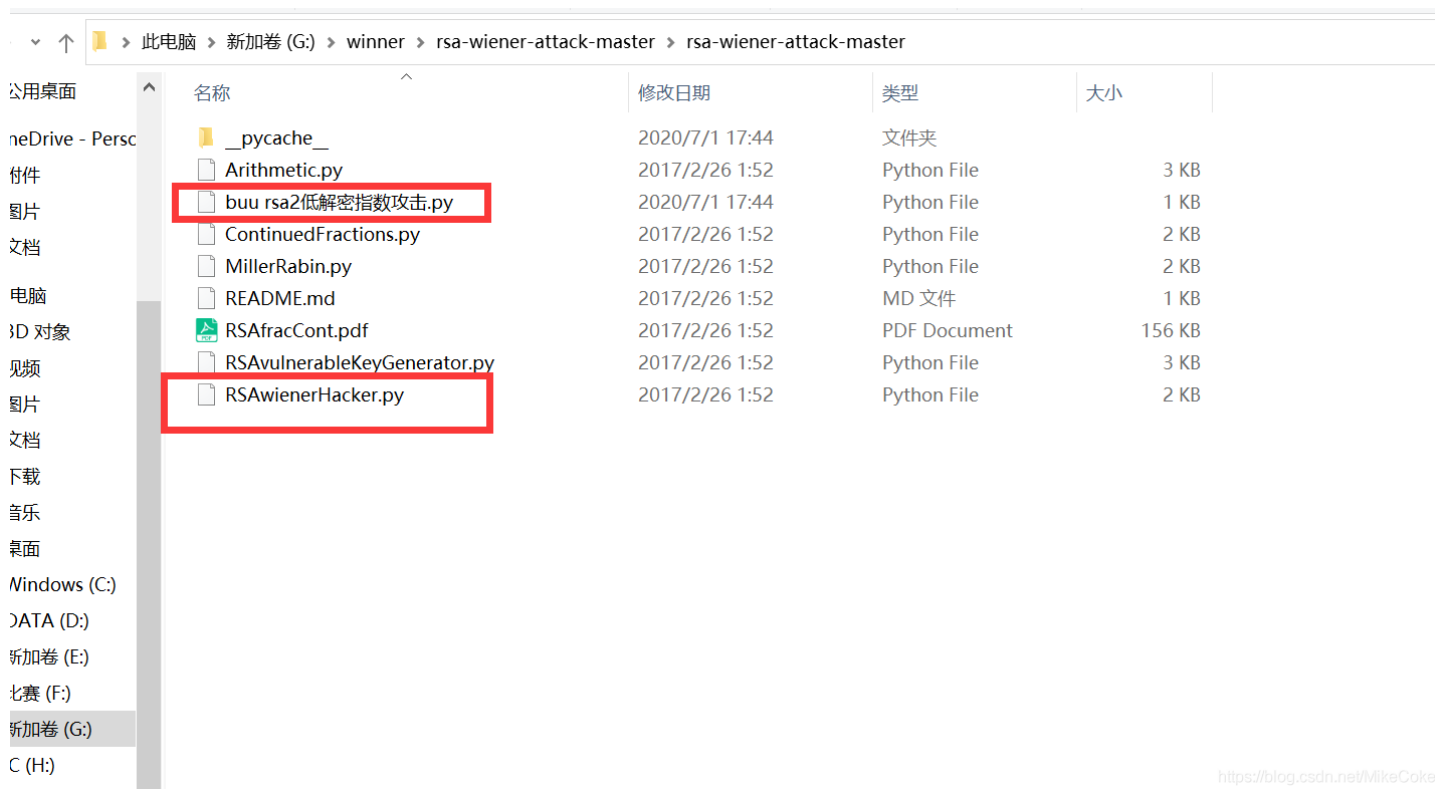
Ln: 1 Col: 0
```

[Github](#)上有公开的轮子

[rsa-wiener-attack](#)

破解脚本：求出d的值

(注意，这里要将破解脚本和rsa-wiener-attack的py文件放在同一个目录下)



名称	修改日期	类型	大小
公用桌面			
neDrive - Persc			
附件			
图片			
文档			
电脑			
3D 对象			
视频			
图片			
文档			
下载			
音乐			
桌面			
Windows (C:)			
DATA (D:)			
新加卷 (E:)			
七赛 (F:)			
新加卷 (G:)			
C (H:)			
名称			
修改日期			
类型			
大小			
文件夹	2020/7/1 17:44		
Python File	2017/2/26 1:52		3 KB
Python File	2020/7/1 17:44		1 KB
Python File	2017/2/26 1:52		2 KB
Python File	2017/2/26 1:52		2 KB
MD 文件	2017/2/26 1:52		1 KB
PDF Document	2017/2/26 1:52		156 KB
Python File	2017/2/26 1:52		3 KB
Python File	2017/2/26 1:52		2 KB

<https://blog.csdn.net/MikeCoke>

```
import RSAwienerHacker
n = 101991809777553253470276751399264740131157682329252673501792154507006158434432009141995367241962525705950046
2534001888846582624965347064387915150718858608975527366568995669157312972258172506398736433763101039921706469065
57242832893914902053581087502512787303322747780420210884852166586717636559058152544979471
e = 467319195632657213071051804103025186766761355097379929126250929768490752621920925493230823675182643786305433
3821902574482091647191369607205029199062048658171941035438512176076137422937484769514823059600540997838336974030
5816082770283909611956355972181848077519920922059268376958811713365106925235218265173085

d = RSAwienerHacker.hack_RSA(e,n)
if d:
    print(d)
```

再在py2中，用题目给的代码将求得的d进行MD5哈希

菜鸟工具 WEB 在线编辑器 SVG 在线编辑器 实例归档 菜鸟教程 输入关键字..... Q

Python 在线工具 清空 邮件反馈

```
1 # -*- coding: UTF-8 -*-
2 import hashlib
3 d = 8920758995414587152829426558580025657357328745839747693739591820283
4 flag = "flag{" + hashlib.md5(hex(d)).hexdigest() + "}"
5 print flag
6
```

flag{47bf28da384590448e0b0d23909a25a4}

<https://blog.csdn.net/yikaCola>

得到 flag{47bf28da384590448e0b0d23909a25a4}

注意：题目中给的代码要在py2中运行，d进行数的运算。
如果在py3中进行哈希的话，会报错提示你要先将d进行编码，然后一直出错

11.RSA5

得到的txt文件：

```
m = xxxxxxxx
e = 65537
===== n c =====
n = 204749188940517785333052623456018809280882844711218237540497253540724771558737788480550738433458206978866410
8684261248654125018396596600159134203156295356179333234164133430284799610841746636068813986650517968951658930563
6902137210185624650854906780037204412206309949199080005576922775773722438863762117750429327585792093447423980002
4012006133029438342128209092697138766834658173691585858222946750569789706122028854264360719502145382629210774090
7616041743669983613880116262131484560879687020683470411670776316984738722330782890857094498441697301942752979002
9089766264949078038669523465243837675263858062854739083634207
c = 974463908243330865728978769213595400782053398596897741316275722596415018912929508637393850919224969271766388
7100251950398969619560628955700621469477363403429279749926166788933727442619541728734908788054832411963458817211
6407865115606711995781642276852444202568807946265675560598210417400163534587402213304540234401004596111172015199
0412034477755851802769069309069018738541854130183692204758761427121279982002993939745343695671900015296790637464
8803373755115364247968909965266812006330868410363203958477259357447579930133528046505750681361292955913065692133
00156333650910795946800820067494143364885842896291126137320

n = 209188199606488913494382630469549022109591464078609807421659302537813187592856924925114752632342420025094190
7954564405175525131139263576341255349974450642156607472126882233732163726594222679034383985618210057553984535887
7493718334237585821263388181126545189723429262149630651289446553402190531135520836104217160268349688525168375213
4625702136128458989896943242694102024968716886499783702846610173990569039318406567573308596261837733965740564130
1736760644654019997315563046623945363723293690406370655116065029503127338561947074059351026728595790580156636250
2262757750629162937373721291789527659531499435235261620309759
c = 158196362019711855386948805051204693325821518567140708245218031218482923875568641771962297189237708100721041
5543203868251143497935308979186108741514408785567913438339689781745872654388309356760032520459615664930593035257
527403942547083635500269114586443575533821133969266951545158052745938252574301327696822347115053614052423028835
5325092206413787608006933515426338607022257726389305010215714159073481282696812241783002482726897053089112822086
8545966820050705718342066295911395607758478173798325478870304827569892142702988428255746833439967784996234219614
0864403989162117738206246183665814938783122909930082802031855
```

n = 2503325402590057272309009119214202035102128025171240430639570015203949157303275213121550823878739232052905
7955187382437487095746716398954206348941663671365464248671721923122507411526968411942808635253547168335948624820
3644461465935500517901513233739152882943010177276545128308412934555830087776128355125932914846459470221102007666
9122119923105388906543964871117053857305028435897272898296921521771347530986497814122470656606378262820551699918
2409911091657685618887697562137660663425892778402578714226336715294710872075722244668641562747970366603187163565
6314282727051189190889008763055811680040315277078928068816491

c = 418530852941687400583123078101409240719845138595567739966850183390262347839566927940488399072518433270915244
3372583701076198786635291739356770857286702107156730020004358955622511061410661058982622055199736820808203841446
7963052843946517144309186903894869205608346723161581464531837894121409390290293247560353580817544266451600332629
2433024867521610827098015704970548862026348512948095281476400286528001918512766244931832427938327776641625814227
5143923532168798413011028271543085249029048997452212503111742302302065401051458066585395360468447460658672952851
643547193822775218387853623453638025492389122204507555908862

n = 212069680973141310071834279444868019535831511514436279431137369967767871811110639579606980926968005550441991
5676567793537314959822118479228681221329461774983460769630211613674566281665811705542780331523004270069512571840
1646810484873064775005221089174056824724922160855810527236751389605017579545235876864998419873065217294820244730
7851205251265658155602290018876228375491181680816851833710923951285981250047302689102760248068085658020813668989
0403250992045378599705615049764523492552888387941964218910964900913238158667339002761476660503895101585308672116
8018787523459264932165046816881682774229243688581614306480751

c = 452103801104475844189112846846723308849388575085058898570851991115477809059713612615028904189345412667446814
1393472662337350361712212694867311622970440707727941113263832357173141775855227973742571088974593476302084111770
6257642228383662775595608870429488598921385514726806545178149166092797483655806107122598566777405184770865315922
3310717547006829190360750579943293198966370747701790461142621377023839700574373038608003195569415846655847559975
1940245039167629126576784024482348452868313417471542956778285567779435940267140679906686531862467627238401003459
101637191297209422470388121802536569761414457618258343550613

n = 228220397330493881109367781730147656636633038117912832343612306497758059239021734385539278054074631061046997
7399415837570403309347176138779985216833789852698052175361430789966901593138781992742187531630459152190159282381
4417756447695701045846773508629371397013053684553042185725059996791532391626429712416994990889693732805181947970
0714293095996149737727365562994042464247916606792538849400217288469063441988547791919517397193429087613306619104
7711993342855077424291042095249692960568615479948783992342433635374744215357167806452076314979329436078782175170
3543288696726923909670396821551053048035619499706391118145067

c = 154064985807617801086258918780085268151453720962340839366814422251550972992648086243588266869065355948536226
8737926896946843307238814978660739539642410431882087944374311235870654675393521575607834595937529965071855575969
8887852318017597503074317356745122514481807843745626429797861463012940172797612589031686718185390345389295851075
2792785161470766022701785406901478083141727989874972593300378103285234648518956218518590278236816559341047136895
3984804716308866689647366550015817904619653821077889773020957270843006765841175595986603353170046055155638099398
2706171848970460224304996455600503982223448904878212849412357

n = 215741398553414329084740647843184620184752968093272855323377069401269425753495076682892140780261026822527137
5770308155309310882321406379151848228984678019732982113950797476378026029030960088492081195984292554058396708567
0848765317877441480914852329276375776405689784571404635852204097622600656222714808541872252335877037561388406257
1817152787666528247863762622492749604671939619566909748536797952491587510784222965803675062197197387621599659588
7780618746107068907129094818194956125414431077694333485977512165018624584603172050794498783848972312789722341680
2436021278671237227993686791944711422345000479751187704426369

c = 203668561507103051245830653752976618197952422383764852649511853369960837446045934189833362851854911974260185
9503144465212328846149187902109602820369413668320344169298706956351302600186143572211798555990969267090734756359
4578265880806540396777223906955491026286843168637367593400342814725694366078337030937104035993569672959361347287
8941430271868468567729830583289197167029822221428488481177684999966175883053014830854285472673370709987674125402
2591150819684225313435590126386112150065024029674670296759422440165022016878053714165448921501914212228430811628
4129004257364769474080721001708734051264841350424152506027932

n = 253602274126666124901021611311745848192409318031964484812243052505838414395810085285359308141673383819837649
9129657563723191654764797057375826941116821930237054168478912511250502114850680964308195023762370318102569658599
8044695691322012183660424636496897073045557400768745943787342548267386564625462143150176113656264450210023925571
9459614057092766319907316021981042875285280556500504861598376122796004152594863061549475140054089075900837477589
5311548612486548672063382055913506344094252803140295195855763083350377511201071560427811432552899377108123353524
7118481765852273252404963430792898948219539473312462979849137

c = 198927725246514523410275956194827343562434356715923981726803799815027596957840879006690899199877056758999456
5864862380009027259915459012308218964502180095807686151839732543952113999565202637713236823250210862003340005134

6127757698623886142621793423225749240286511666556091787851683978017506983310073524398287279737680091787333547538
2399206077610809882436395475708183637886732495827830154756821099847152931631373244398628385744601087937141726036
7247776683135641130444688199867477950118816360066448803294363969482869898473949220069968446274892288355000265291
3518229322945040819064133350314536378694523704793396169065179

n = 227268552446323560291596917534518221633315192375476399387795177514964987131745889355665761673295764947902193
6072787716607413649612992729629699697004808287048880445656498666712938813655613701334622811898193689951068758958
5286517151323048293150257036847475424044378109168179412287889340596394755257704938006162677656581509375471102546
2613557482518690480036005200346562645219318086510385241341857329295703847059185639820656841457664279625022615224
8199419198982011057598190699843155310752554200118765570353468323177798841926833824954764133571839331229580004473
4534761692799403469497954062897856299031257454735945867491191

c = 604011979517585640754108236002353220461472385868863672482271271757275979396024634180030814973980987123431304
9629732934797569781053000686185666374833978403290525072598774001731350244744590772795701065129561898116576499984
185920661271123665356132719193665474235596884239108030605882778688561223782226811405705191803212869769471540422
7262241130398101130258622563085989273172464057465812547828711519840625384736797988376800081260539548295269868960
4477719478947595442185921480652637868335673233200662100621025061500895729605305665864693122952557361871523165300
206070325660353095592778037767395360329231331322823610060006

n = 232973337914430532973630007868353360952522908184619500545426583274845074065946327857127674599589179430955225
9422820542342820734512889974580092731914725766977381266954278283923774430518009827657884192949634596399751224421
9376701787616046235397139381894837435562662591060768476997333538748065294033141610502252325292801816812268934171
3619343999515486272677914010897039373890125865810802233130601594562388570807406995286664113030299348070112149539
8416978584471415962779201692649095528269787714161463880639768930679532834477847869208475421675342584255781889946
7945102646776342655167655384224860504086083147841252232760941

c = 541812030120837871311588946557996425787181411451504609609096015973785907682925851692036157785390392595419840
684375730368755784830230220229295916902430205737843601806700738234756698575708612424928480440868739120075888681
6720622065291565664212766111078029174189936250296906271968138303263698742497776192396033006058768659675157190797
9711591057865356278789901931013994590495802488241783373630489476543348947623457535675527514725657738702287334890
6900149634940747104513850154118106991137072643308620284663108283052245750945228995387803432128842152251549292698
947407663643895853432650029352092018372834457054271102816934

n = 288736679047156827229872342934932003069769478987112550641251159336669686787425988587224314262189144629035215
9634177113169561938226619423356167782435737980530388599380426643681060626302209790026697525043157565468691504969
3091467864820512767070713267708993899899011156106766178906700336111712803362113039613548672937053397875663144794
0180870177319490877948949037376823839161732674214034081409677130710260018747334872950075010688710446491706157098
9145185679223231552669622016184274266477858128732131874820243146650894890274531437229979956162518695523467301209
8210919745879882268512656931714326782335211089576897310591491

c = 991988046378683668498795797909152747747144499639237524407552784186550916018166654301631763496351243751032419
8702416322841377489417029572388474450075801462996825244657530286107428186354172836716502817609070590929769261932
3242753532899393025364403106286983492448720640057006445202237276709507879242960042968830329789412008833626539933
5163854586020717902247249267125663042722846185266811803531702142867595487494701519774591691819772512112223636938
2741533983023462255913924692806249387449016629865823316402366017657844166919846683497851842388058283856219900535
567427103603869955066193425501385255322097901531402103883869

n = 223246859475396537224999324694096075330654191573478139619580756890476904652664043841994836839085947873124455
2815963552783390447580189038145565380726550121732875787135273129300030343820531581679266391757906667484230774384
5261771032363928568844669895768092515658328756229245837025261744260614860746997931503548788509983868038349720225
3057309855762936752690737090223507008365100540676417537132129999543070225244958855833617073785137421625663390101
3435490786373320592184503891822446390378984188140081407458726172028387976012207090146651711826542286342037692153
6734845502100251460872499122236686832189549698020737176683019

c = 149152705020329498988282924856039518480497727774712614310395721916462418752844104783735126358044068647476738
0464005540264627910126483129930668344095814547592115061057843470131498075060420395111008619027199037019925701236
6601665630682456839757877628043595201647016916909164825910261385827055582468694961627597808784371379608230000439
8822730300387641050312137016330371160335943076453933759786686250845152815828510325181005874187968787521838416028
2506172706613359477657215420734816049393339593755489218588796607060261897905233453268671411610631047340459487937
479511933450369462213795738933019001471803157607791738538467

n = 276467464237590201110078286532640279992578476456661299077890260545943936488002361170467691127626417788656208
9244342310018961932758581138488351542491875274955962755363778503735963980112521325616300843194259372793193189819

9/2/552/68626/756184/98330291012496925/3/16030/06695/02510982283555/4085104/0226/2485/4343246464///28823142151/6
1147322574972402841640169140186890445572189203002622346528406324060672733752693010084098601931808223667358772882
0578331432610226375650378673612232134832003195001214490586955620401743059365605286793949363316349958024222476340
4338807022510136217187779084917996171602737036564991036724299
c = 219915241289572605360437712848549203931058081267001282221258567755068857219711931093613159611291908146746471
3646488708789399066089496161283820508640101888545766748891189865427023556198011117460332372128091119748828658526
9356849579263043456316319476495888696219344219866516861187654180509247881251251278919346267129904739277386289240
3943845751243311356559435138310099340233974570821846997377343888237633068053264303958499357702138175333872354863
0700889241092061166993269301816556941744588581082574960938862723123584091264465468581962093166334629759633483449
8661789016450371769203650109994771872404185770230172934013971

n = 205454874058169287317389883744750126868279337097897843918557068351362702709334012030193291369376508783861171
8777653063934257212323718805397862269728252147391797828283043216115322121619416987966954199884069138302548722085
0872075436064308499924958517979727954402965612196081404341651517326364041519250125036424822634354268773895465698
9208834392229965812263585958739939766046998306139323207205541300116712979444335150471805654844951910038875998912
8903798201021635783107832815902895322205691818936584071158867109333301311745403431362285508279581312233856244622
3041211192277089225078324682108033843023903550172891959673551
c = 142274391881910294612504766927905396546191998884873194291144145579753763086889080281408171572055798040597838
0764130557738572475853013851497296220906223057610740614240260348437562607734519088309409763601977137786633953151
1965136650567412363889183159616188449263752475328663245311059988337996047359263288837436305588848044572937759424
466586870280512424336807064729894515840552404756879590698797046333364454651204450875876217439066242796217796347
7237880295910971440051618371832326727382473654016854594644443758629921411042473815995738835078599934853517155356
9373088251552712391288365295267665691357719616011613628772175

n = 273597277115842772348971577240558527940192168452297989386558142694600463843535681385985677553925596534609494
4455787912004079679814221893925184476246127025167239954677406727534829100396255196464874205321542462025699934544
8398805278592777049668281558312871773979931343097806878701114056030041506690476954254006592555275342579529625231
1943213579046685121215395148807040469699748984120956750825853154582675910167349246462943576669242939084183455089
0211271107523204799877530360317536396405504858976931856210488365975497495556172569477975427960672635858886247919
8815999276839234952142017210593887371950645418417355912567987
c = 378852978424825502708167454087701637280784822277688792045348887824713793057829679743764792249451048376765115
0492933356093288965943741570268943861987024276610712717409139946409513963043114463933146088430004237747163422802
9592502966025706493630161515813640067958942265995847080725826969967405188876067854607758510298142803593857630910
789023019572264846204285136046305851315116701576319059122588420277284045656364315950780571100411390141750375118
105082363820780353311429510911616160851391754754434764819568054850823810901159821297849790005646102129354035735
350124476838786661542089045509656910348676742844957008857457

n = 275459376037517372487852208917357964689733297380762091440799214499672925723494245390105022875640301168312612
6819738465051104306873891142916973064013594780088598717153926721461190768757058700193382920865510082804565139161
8089603288456570334500533178695238407684702251252671579371018651675054368606282524673369983034682330578308769886
4563358187338272372945704768536735526853616891442615528957582665223930041160178493973462591192210638216632809358
2044067182560145241748733010528088952000791797911556806716159005827741837149322863123245797249428501476746989364
7892888681433965857496916110704944758070268626897045014782837
c = 140691129706088957324170399775427326657966018937624015008787868716806457987547833156935112617400597251713424
0418657106697254633281366771113566117665942461993610103890343914429488637932259163576668264517988805861757757240
9307484708171144488708410543462972008179994594087473935638026612679389759756811490524127195628741262871304427908
4812149924711828593088287781190057509289357649279672123435265034105157937172013603604379813225767980562766571403
6333270071473222484834680896399230240903770609458896417023952119358947007083979040459725299081858371786914022981
1712295005710540476356743378906642267045723633874011649259842

n = 257461620756979115602631817912164330625741785724246003368562781761127330544314632539034331282327090541416071
0089117780428581378324773506375340652467803056128449148122168195456480414145466692865754967026677565986281492438
6584148785453647316864935942772919140563506305666207816897601862713092809234429096584753263707828899780979223118
1810092936555631465267923889134625573064336642969663314699064286651274388293997030028678002699478558692620367142
5655007552019312598701194519227353173227664172800840685587159867893658532478243866874681051666015201824425300809
2470066555687277138937298747951929576231036251316270602513451
c = 173442848602754894774915258199228553267922751287197094012925456081228598298274620883900446122349675516828799
5430145842584283199551383241035532806556209876366032616326203320034733877343909570994420225249455217258950391596
593152432652366328977583152664722241920800537867331030623906674081852296232306336271542832728410803631170229642
7175249423323908424670351436315044011407270832707324642374439152638658805803087761112197189617463788429246441421

2724357382497253381947907938102310358586209906338212975756012407467615062228870609411007556770640344292069647262
7797607697962873026112240527498308535903232663939028587036724

n = 232884869341171203150369194185881362270284854941379301963237153362088493278339656938946705672179717279212438
3912996912878385301576015544677059069603758268484593713279004736321636208727786133696476089021405973277938302034
9204803205725870225429985939570141508220041286857810048164696707018663758416807708910671477407366098883430811861
9330149734093901799485777125797493522994403105436890356514653998679084288855412377761434043763334429493970632492
2370235505157179055515120386682186790853173378878497866747870767298453951243154955867246775271200451930031899920
8102076732501412589104904734983789895358753664077486894529499

c = 107382544181140765480714488449640464681416217406032143849863541891052369770710014292715606364280759704598909
5827494176252811644517116104004083335787613468974984694005261939275039468350481608119343235066945244611328563898
2551762586656329109007214019944975816434827768882704630460001209452239162896576191876324662333153835533956600295
2551583770251984269509440406432354302110110635860324677243297357859473720517590421381710541658548424729905838008
9998489323254909276640051030008358551301417122042310345229289149614180695630039654068238166836756456942781309206
4053993103537635994311143010708814851867239706492577203899024

n = 19591441383958529435987291139363466570013525783579093476572572397775404248117498177830612332358179165606891
3834404149773274901151973630303898627739403671879097137465683274105454705641777150123449476850978036907544355090
7847298246275717420562375114406055733620258777905222169702036494045086017381084272496162770259955811174440490126
5147478766613177506494887749923480050443890811016860164462192640699713706463195464297829048100630203247041384956
087615325633106997533224487106038369304448193226580150581964699853519208303687255168340576612396848790764898090
0712118052346174533513978009131757167547595857552370586353973

c = 383491709888720293198196870465911934162443229475936191955393755105349960744033323401818914197024630229938574
2548278589896033282894981200353270637127213483172182529890495903425649116755901631101665876301799865612717750360
0890851791427506646034541936420530163847145158558683687235089222717671902855211377856880756228329248292483627744
7645623282688580104696938451954938542825959156671689084460469625878363939085415303932948072620514719924718362153
5172450825979047132495439603840806501254997167051142427157381799890725323765558803808030109468048682252028720241
357478614704610089120810367192414352034177484688502364022887

n = 192542425715884301713081917578712610753585211586247457027440575560546523324959611967953696304847829302920032
3873026739646249173355771537995696969423826790898525169983470773440077531145286892433086650242957695193427922323
4676654749272932769107390976321208605516299532560054081301829440688796904635446986081691156842271268059970762004
2592190367531749099423432044327950763774321076302036217545528041244087923582200718623694432015841557118933888773
5013802323862456661655124680405472049281622665146701780250409407061489255644442591592026948586179953247338330462
2064493223627552558344088839860178294589481899206318863310603

c = 679055353399129720580456199122549310531239882518768225078019751078476522642966328422040048056303934193859978
3346724051076211265663468643826430109013245014035811178295081939958687087477312867720289964506097819762095244479
1293599988676718118197381966878846966804634586613743109946107600094742641157502049208755274344864375366235896845
194115191001702914233674249385668203154865074442022240800387911846576127391675529089811299152554611419106402299
1329724370064632569903856189236177894007766690782630247443895358893983735822824243487181851098787271270256780891
094405121947631088729917398317652320497765101790132679171889

n = 268097002511712791029749629491844111364593722676205351984214498332984480925804974853019537966191853393160643
8779809222029863042820755648280573980342027905619119436004965176741257260918768050807307465329135099825393879326
9214230457117194434853888765303403385824786231859450351212449404870776320297419712486574804794325602760347306432
9272817161603688301879449401289079710278385100795194668461761065651647309639888924002400630893977204149213989363
9992794823519508520217126472881618453265113822186224096965518559662828581405708244832174956794394627377618465769
8104465062749244327092588237927996419620170254423837876806659

c = 386213556608434013769864727123879412041991271528990528548507451210692618986652870424632219424601677524265011
0431467483097740678949850692880679525461394168194040396884547560448627846308828334960908225685805728590298006466
7130174890152813215371291330117925487987744132228591454497451972730731100233035053485786751646661247476975357785
8660075830592891403551867246057397839688329172530177187042229028685862036140779065771061933528137423019407311473
5818324058990897092517470027880320020944953796146865446729690732493097034825563860246228147310157678100429698137
52548617464974915714425595351940266077021672409858645427346

解题思路:

1.题目给了一组 e 的值, 和多组 n 的 c 的值, 共用相同的密文 m ,

(有点低加密指数广播攻击的感觉, 因为还没去了解低加密指数广播攻击的解密原理, 所以不是特别确定)

只要满足一下情况, 我们便可以考虑使用低加密指数广播攻击:

加密指数 e 非常小

一份明文使用不同的模数 n , 相同的加密指数 e 进行多次加密

可以拿到每一份加密后的密文和对应的模数 n 、加密指数 e

2.通过对不同的 n 进行 $\text{gcd}()$ 算法, 求出最大公约数, (即 p)

3.求出 P 了, 就能求出 q , 进而求出 d , 解出明文 m

python3脚本:

```
from gmpy2 import *
```

```
n0 = 20474918894051778533305262345601880928088284471121823754049725354072477155873778848055073843345820697886641
0868426124865412501839659660015913420315629535617933323416413343028479961084174663606881398665051796895165893056
3690213721018562465085490678003720441220630994919908000557692277577372243886376211775042932758579209344742398000
2401200613302943834212820909269713876683465817369158585822294675056978970612202885426436071950214538262921077409
0761604174366998361388011626213148456087968702068347041167077631698473872233078289085709449844169730194275297900
29089766264949078038669523465243837675263858062854739083634207
```

```
c0 = 97446390824333086572897876921359540078205339859689774131627572259641501891292950863739385091922496927176638
8710025195039896961956062895570062146947736340342927974992616678893372744261954172873490878805483241196345881721
1640786511560671199578164227685244420256880794626567556059821041740016353458740221330454023440100459611117201519
9041203447775585180276906930906901873854185413018369220475876142712127998200299393974534369567190001529679063746
4880337375511536424796890996526681200633086841036320395847725935744757993013352804650575068136129295591306569213
300156333650910795946800820067494143364885842896291126137320
```

```
n1 = 20918819960648891349438263046954902210959146407860980742165930253781318759285692492511475263234242002509419
0795456440517552513113926357634125534997445064215660747212688223373216372659422267903438398561821005755398453588
7749371833423758582126338818112654518972342926214963065128944655340219053113552083610421716026834968852516837521
3462570213612845898989694324269410202496871688649978370284661017399056903931840656757330859626183773396574056413
0173676064465401999731556304662394536372329369040637065511606502950312733856194707405935102672859579058015663625
02262757750629162937373721291789527659531499435235261620309759
```

```
c1 = 15819636201971185538694880505120469332582151856714070824521803121848292387556864177196229718923770810072104
1554320386825114349793530897918610874151440878556791343833968978174587265438830935676003252045961566493059303525
7527403942547083635500269114586443575533382113396926695154515805274593825257430132769682234711505361405242302883
5532509220641378760800693351542633860702225772638930501021571415907348128269681224178300248272689705308911282208
6854596682005070571834206629591139560775847817379832547887030482756989214270298842825574683343996778499623421961
40864403989162117738206246183665814938783122909930082802031855
```

```
n2 = 25033254625906757272369609119214202033162128625171246436639570615263949157363273213121556825878737923265290
5795518738243748709574671639895420634894166367136546424867172192312250741152696841194280863525354716833594862482
0364446146593550051790151323373915288294301017727654512830841293455583008777612835512593291484645947022110200766
6912211992310538890654396487111705385730502843589727289829692152177134753098649781412247065660637826282055169991
8240991109165768561888769756213766066342589277840257871422633671529471087207572224466864156274797036660318716356
56314282727051189190889008763055811680040315277078928068816491
```

```
c2 = 41853085294168740058312307810140924071984513859556773996685018339026234783956692794048839907251843327091524
4337258370107619878663529173935677085728670210715673002000435895562251106141066105898262205519973682080820384144
6796305284394651714430918690389486920560834672316158146453183789412140939029029324756035358081754426645160033262
9243302486752161082709801570497054886202634851294809528147640028652800191851276624493183242793832777664162581422
7514392353216879841301102827154308524902904899745221250311174230230206540105145806658539536046844746065867295285
1643547193822775218387853623453638025492389122204507555908862
```

```
n3 = 21206968097314131007183427944486801953583151151443627943113736996776787181111063957960698092696800555044199
1567656779353731495982211847922868122132946177498346076963021161367456628166581170554278033152300427006951257184
```

0164681048487306477500522108917405682472492216085581052723675138960501757954523587686499841987306521729482024473
0785120525126565815560229001887622837549118168081685183371092395128598125004730268910276024806808565802081366898
9040325099204537859970561504976452349255288838794196421891096490091323815866733900276147666050389510158530867211
68018787523459264932165046816881682774229243688581614306480751

c3 = 45210380110447584418911284684672330884938857508505889857085199111547780905971361261502890418934541266744681
4139347266233735036171221269486731162297044070772794111326383235717314177585522797374257108897459347630208411177
062576422283836627755956088704294885989213855147268065451781491660927974836558061071225985667740518477086531592
2331071754700682919036075057994329319896637074770179046114262137702383970057437303860800319556941584665584755997
5194024503916762912657678402448234845286831341747154295677828556777943594026714067990668653186246762723840100345
9101637191297209422470388121802536569761414457618258343550613

n4 = 22822039733049388110936778173014765663663303811791283234361230649775805923902173438553927805407463106104699
7739941583757040330934717613877998521683378985269805217536143078996690159313878199274218753163045915219015928238
1441775644769570104584677350862937139701305368455304218572505999679153239162642971241699499088969373280518194797
0071429309599614973772736556299404246424791660679253884940021728846906344198854779191951739719342908761330661910
4771199334285507742429104209524969296056861547994878399234243363537474421535716780645207631497932943607878217517
03543288696726923909670396821551053048035619499706391118145067

c4 = 15406498580761780108625891878008526815145372096234083936681442225155097299264808624358826686906535594853622
6873792689694684330723881497866073953964241043188208794437431123587065467539352157560783459593752996507185557596
9888785231801759750307431735674512251448180784374562642979786146301294017279761258903168671818539034538929585107
5279278516147076602270178540690147808314172798987497259330037810328523464851895621851859027823681655934104713689
5398480471630886668964736655001581790461965382107788977302095727084300676584117559598660335317004605515563809939
82706171848970460224304996455600503982223448904878212849412357

n5 = 21574139855341432908474064784318462018475296809327285532337706940126942575349507668289214078026102682252713
7577030815530931088232140637915184822898467801973298211395079747637802602903096008849208119598429255405839670856
7084876531787744148091485232927637577640568978457140463585220409762260065622271480854187225233587703756138840625
7181715278766652824786376262249274960467193961956690974853679795249158751078422296580367506219719738762159965958
8778061874610706890712909481819495612541443107769433348597751216501862458460317205079449878384897231278972234168
02436021278671237227993686791944711422345000479751187704426369

c5 = 20366856150710305124583065375297661819795242238376485264951185336996083744604593418983336285185491197426018
5950314446521232884614918790210960282036941366832034416929870695635130260018614357221179855599096926709073475635
945782658808065403967722390695549102628684316863736759340034281472569436607833703093710403599356967295936134728
7894143027186846856772983058328919716702982222142848848117768499996617588305301483085428547267337070998767412540
2259115081968422531343559012638611215006502402967467029675942244016502201687805371416544892150191421222843081162
84129004257364769474080721001708734051264841350424152506027932

n6 = 25360227412666612490102161131174584819240931803196448481224305250583841439581008528535930814167338381983764
9912965756372319165476479705737582694111682193023705416847891251125050211485068096430819502376237031810256965859
9804469569132201218366042463649689707304555740076874594378734254826738656462546214315017611365626445021002392557
1945961405709276631990731602198104287528528055650050486159837612279600415259486306154947514005408907590083747758
9531154861248654867206338205591350634409425280314029519585576308335037751120107156042781143255289937710812335352
47118481765852273252404963430792898948219539473312462979849137

c6 = 19892772524651452341027595619482734356243435671592398172680379981502759695784087900669089919987705675899945
6586486238000902725991545901230821896450218009580768615183973254395211399956520263771323682325021086200334000513
4612775769862388614262179342322574924028651166655609178785168397801750698331007352439828727973768009178733354753
8239920607761080988243639547570818363788673249582783015475682109984715293163137324439862838574460108793714172603
6724777668313564113044468819986747795011881636006644880329436396948286989847394922006996844627489228835500026529
13518229322945040819064133350314536378694523704793396169065179

n7 = 22726855244632356029159691753451822163331519237547639938779517751496498713174588935566576167329576494790219
3607278771660741364961299272962969969700480828704888044565649866671293881365561370133462281189819368995106875895
8528651715132304829315025703684747542404437810916817941228788934059639475525770493800616267765658150937547110254
6261355748251869048003600520034656264521931808651038524134185732929570384705918563982065684145766427962502261522
481994191989820110575981906998431553107525542001187655703534683231779884192683382495476413357183933122958000447
34534761692799403469497954062897856299031257454735945867491191

c7 = 60401197951758564075410823600235322046147238586886367248227127175727597939602463418003081497398098712343130
4962973293479756978105300068618566637483397840329052507259877400173135024474459077279570106512956189811657649998
4185920661271123665356132719193665474235596884239108030605882777868856122378222681140570519180321286976947154042
2726224113039810113025862256308598927317246405746581254782871151984062538473679798837680008126053954829526986896

0447771947894759544218592148065263786833567323320066210062102506150089572960530566586469312295255736187152316530
0206070325660353095592778037767395360329231331322823610060006

n8 = 23297333791443053297363000786835336095252290818461950054542658327484507406594632785712767459958917943095522
5942282054234282073451288997458009273191472576697738126695427828392377443051800982765788419294963459639975122442
1937670178761604623539713938189483743556266259106076847699733353874806529403314161050225232529280181681226893417
1361934399951548627267791401089703937389012586581080223313060159456238857080740699528666411303029934807011214953
9841697858447141596277920169264909552826978771416146388063976893067953283447784786920847542167534258425578188994
67945102646776342655167655384224860504086083147841252232760941

c8 = 54181203012083787131158894655799642578718141145150460960909601597378590768292585169203615778539039259541984
0684375730368755784830230220022929591690243020573784360180670073823475669857570861242492848044086873912007588868
1672062206529156566421276611107802917418993625029690627196813830326369874249777619239603300605876865967515719079
7971159105786535627878990193101399459049580248824178337363048947654334894762345753567552751472565773870228733489
0690014963494074710451385015411810699113707264330862028466310828305224575094522899538780343212884215225154929269
8947407663643895853432650029352092018372834457054271102816934

n9 = 28873667904715682722987234293493200306976947898711255064125115933666968678742598858722431426218914462903521
5963417711316956193822661942335616778243573798053038859938042664368106062630220979002669752504315756546869150496
9309146786482051276707071326770899389989901115610676617890670033611171280336211303961354867293705339787566314479
4018087017731949087794894903737682383916173267421403408140967713071026001874733487295007501068871044649170615709
8914518567922323155266962201618427426647785812873213187482024314665089489027453143722997995616251869552346730120
98210919745879882268512656931714326782335211089576897310591491

c9 = 99198804637868366849879579790915274774714449963923752440755278418655091601816665430163176349635124375103241
9870241632284137748941702957238847445007580146299682524465753028610742818635417283671650281760907059092976926193
2324275353289939302536440310628698349244872064005700644520223727670950787924296004296883032978941200883362653993
3516385458602071790224724926712566304272284618526681180353170214286759548749470151977459169181977251211222363693
8274153398302346225591392469280624938744901662986582331640236601765784416691984668349785184238805828385621990053
5567427103603869955066193425501385255322097901531402103883869

n10 = 2232468594753965372249993246940960753306541915734781396195807568904769046526640438419948368390859478731244
5528159635527833904475801890381455653807265501217328757871352731293000303438205315816792663917579066674842307743
8452617710323639285688446698957680925156583287562292458370252617442606148607469979315035487885099838680383497202
2530573098557629367526907370902235070083651005406764175371321299995430702252449588558336170737851374216256633901
0134354907863733205921845038918224463903789841881400814074587261720283879760122070901466517118265422863420376921
536734845502100251460872499122236686832189549698020737176683019

c10 = 1491527050203294989882829248560395184804977277747126143103957219164624187528441047837351263580440686474767
3804640055402646279101264831299306683440958145475921150610578434701314980750604203951110086190271990370199257012
3666016656306824568397578776280435952016470169169091648259102613858270555824686949616275978087843713796082300004
3988227303003876410503121370163303711603359430764539337597866862508451528158285103251810058741879687875218384160
2825061727066133594776572154207348160493933395937554892185887966070602618979052334532686714116106310473404594879
37479511933450369462213795738933019001471803157607791738538467

n11 = 2764674642375902011100782865326402799925784764566612990778902605459439364880023611704676911276264177886562
0892443423100189619327585811384883515424918752749559627553637785037359639801125213256163008431942593727931931898
1997275527686267756184798330291012496925737160307066957025109822835557408510470226724857434324646477728823142151
7611473225749724028416401691401868904455721892030026223465284063240606727337526930100840986019318082236673587728
8205783314326102263756503786736122321348320031950012144905869556204017430593656052867939493633163499580242224763
40433880702251013621718779084917996171602737036564991036724299

c11 = 2199152412895726053604377128485492039310580812670012822212585677550688572197119310936131596112919081467464
7136464887087893990660894961612838205086401018885457667488911898654270235561980111174603323721280911197488286585
2693568495792630434563163194764958886962193442198665168611876541805092478812512512789193462671299047392773862892
4039438457512433113565594351383100993402339745708218469973773438882376330680532643039584993577021381753338723548
6307008892410920611669932693018165569417445885810825749609388627231235840912644654685819620931663346297596334834
498661789016450371769203650109994771872404185770230172934013971

n12 = 2054548740581692873173898837447501268682793370978978439185570683513627027093340120301932913693765087838611
7187776530639342572123237188053978622697282521473917978282830432161153221216194169879669541998840691383025487220
8508720754360643084999249585179797279544029656121960814043416515173263640415192501250364248226343542687738954656
9892088343922299658122635859587399397660469983061393232072055413001167129794443351504718056548449519100388759989

1289037982010216357831078328159028953222056918189365840711588671093333013117454034313622855082795813122338562446
223041211192277089225078324682108033843023903550172891959673551

c12 = 1422743918819102946125047669279053965461919988848731942911441455797537630868890802814081715720557980405978
3807641305577385724758530138514972962209062230576107406142402603484375626077345190883094097636019771377866339531
5119651366505674123638891831596161884492637524753286632453110599883379960473592632888374363055888480445729377594
2446658687028051242433680706472989451584055240475687959069879704633333644546512044508758762174390662427962177963
4772378802959109714400516183718323267273824736540168545946444437586299214110424738159957388350785999348535171553
569373088251552712391288365295267665691357719616011613628772175

n13 = 2735972771158427723489715772405585279401921684522979893865581426946004638435356813859856775539255965346094
9444557879120040796798142218939251844762461270251672399546774067275348291003962551964648742053215424620256999345
4483988052785927770496682815583128717739799313430978068787011140560300415066904769542540065925552753425795296252
3119432135790466851212153951488070404696997489841209567508258531545826759101673492464629435766692429390841834550
8902112711075232047998775303603175363964055048589769318562104883659754974955561725694779754279606726358588862479
198815999276839234952142017210593887371950645418417355912567987

c13 = 37885297842482550270816745408770163728078482227768879204534888782471379305782967974737647922494510483767651
1504929333560932889659437415702689438619870242766107127174091399464095139630431144639331460884300042377471634228
0295925029660257064936301615158136400679589422659958470807258269699674051888760678546077585102981428035938576309
1078902301957226484620428513604630585131511167015763190591225884202772840456563643159507805711004113901417503751
1810508236382078035331114295109116161608513917547544347648195680548508238109011598212978497900056461021293540357
35350124476838786661542089045509656910348676742844957008857457

n14 = 2754593760375173724878522089173579646897332973807620914407992144996729257234942453901050228756403011683126
1268197384650511043068738911429169730640135947800885987171539267214611907687570587001933829208655100828045651391
6180896032884565703345005331786952384076847022512526715793710186516750543686062825246733699830346823305783087698
8645633581873382723729457047685367355268536168914426155289575826652239300411601784939734625911922106382166328093
5820440671825601452417487330105280889520007917979115568067161590058277418371493228631232457972494285014767469893
647892888681433965857496916110704944758070268626897045014782837

c14 = 1406911297060889573241703997754273266579660189376240150087878687168064579875478331569351126174005972517134
2404186571066972546332813667711135661176659424619936101038903439144294886379322591635766682645179888058617577572
4093074847081711444887084105434629720081799945940874739356380266126793897597568114905241271956287412628713044279
0848121499247118285930882877811900575092893576492796721234352650341051579371720136036043798132257679805627665714
0363332700714732224848346808963992302409037706094588964170239521193589470070839790404597252990818583717869140229
811712295005710540476356743378906642267045723633874011649259842

n15 = 2574616207569791156026318179121643306257417857242460033685627817611273305443146325390343312823270905414160
7100891177804285813783247735063753406524678030561284491481221681954564804141454666928657549670266775659862814924
3865841487854536473168649359427729191405635063056662078168976018627130928092344290965847532637078288997809792231
1818100929365556314652679238891346255730643366429696633146990642866512743882939970300286780026994785586926203671
4256550075520193125987011945192273531732276641728008406855871598678936585324782438668746810516660152018244253008
092470066555687277138937298747951929576231036251316270602513451

c15 = 1734428486027548947749152581992285532679227512871970940129254560812285982982746208839004461223496755168287
9954301458425842831995513832410355328065562098763660326163262033200347338773439095709944202252494552172589503915
965931524326523663289775831526647222419208005378673310306239066740818522962323063362715428327284108036311702296
4271752494233239084246703514363150440114072708327073246423744391526386588058030877611121971896174637884292464414
2127243573824972533819479079381023103585862099063382129757560124074676150622288706094110075567706403442920696472
627797607697962873026112240527498308535903232663939028587036724

n16 = 2328848693411712031503691941858813622702848549413793019632371533620884932783396569389467056721797172792124
3839129969128783853015760155446770590696037582684845937132790047363216362087277861336964760890214059732779383020
3492048032057258702254299859395701415082200412868578100481646967070186637584168077089106714774073660988834308118
6193301497340939017994857771257974935229944031054368903565146539986790842888554123777614340437633344294939706324
9223702355051571790555151203866821867908531733788784978667478707672984539512431549558672467752712004519300318999
208102076732501412589104904734983789895358753664077486894529499

c16 = 1073825441811407654807144884496404646814162174060321438498635418910523697707100142927156063642807597045989
0958274941762528116445171161040040833357876134689749846940052619392750394683504816081193432350669452446113285638
9825517625866563291090072140199449758164348277688827046304600012094522391628965761918763246623331538355339566002
9525515837702519842695094404064323543021101106358603246772432973578594737205175904213817105416585484247299058380
0899984893232549092766400510300083585513014171220423103452292891496141806956300396540682381668367564569427813092
064053993103537635994311143010708814851867239706492577203899024

```

n17 = 1959144138395852943559872911393634665700135257835790934765725723977754042481174981778306123323581791656068
91383440414977327490115197363038986277394036718790971374656832741054547056417771501234494768509780369075443550
9078472982462757174205623751144060557336202587779052221697020364940450860173810842724961627702599558111744404901
2651474787666131775064948877499234800504438908110168601644621926406997137064631954642978290481006302032470413849
5608761532563310699753322444871060383693044481932265801505819646998535192083036872551683405766123968487907648980
900712118052346174533513978009131757167547595857552370586353973
c17 = 3834917098887202931981968704659119341624432294759361919553937551053499607440333234018189141970246302299385
7425482785898960332828949812003532706371272134831721825298904959034256491167559016311016658763017998656127177503
6008908517914275066460345419364205301638471451585586836872350892227176719028552113778568807562283292482924836277
4476456232826885801046969384519549385428259591566716890844604696258783639390854153039329480726205147199247183621
5351724508259790471324954396038408065012549971670511424271573817998907253237655588038080301094680486822520287202
41357478614704610089120810367192414352034177484688502364022887

n18 = 1925424257158843017130819175787126107535852115862474570274405755605465233249596119679536963048478293029200
3238730267396462491733557715379956969694238267908985251699834707734400775311452868924330866502429576951934279223
2346766547492729327691073909763212086055162995325600540813018294406887969046354469860816911568422712680599707620
0425921903675317490994234320443279507637743210763020362175455280412440879235822007186236944320158415571189338887
7350138023238624566616551246804054720492816226651467017802504094070614892556444425915920269485861799532473383304
622064493223627552558344088839860178294589481899206318863310603
c18 = 6790553533991297205804561991225493105312398825187682250780197510784765226429663284220400480563039341938599
7833467240510762112656634686438264301090132450140358111782950819399586870874773128677202899645060978197620952444
791293599886767181181973819668788469668046345866137431099461076000947426411575020492087552743448643753662358968
4519411519100170291423367424938566820315486507444202022408003879118465761273916755290898112991525546114191064022
9913297243700646325699038561892361778940077666907826302474438953588939837358228242434871818510987872712702567808
91094405121947631088729917398317652320497765101790132679171889

n19 = 2680970025117127910297496294918441113645937226762053519842144983329844809258049748530195379661918533931606
4387798092220298630428207556482805739803420279056191194360049651767412572609187680508073074653291350998253938793
2692142304571171944348538887653034033858247862318594503512124494048707763202974197124865748047943256027603473064
3292728171616036883018794494012890797102783851007951946684617610656516473096398889240024006308939772041492139893
6399927948235195085202171264728816184532651138221862240969655185596628285814057082448321749567943946273776184657
698104465062749244327092588237927996419620170254423837876806659
c19 = 3862135566084340137698647271238794120419912715289905285485074512106926189866528704246322194246016775242650
1104314674830977406789498506928806795254613941681940403968845475604486278463088283349609082256858057285902980064
6671301748901528132153712913301179254879877441322285914544974519727307311002330350534857867516466612474769753577
8586600758305928914035518672460573978396883291725301771870422290286858620361407790657710619335281374230194073114
7358183240589908970925174700278803200209449537961468654467296907324930970348255638602462281473101576781004296981
3752548617464974915714425595351940266077021672409858645427346

```

```

n=[n0,n1,n2,n3,n4,n5,n6,n7,n8,n9,n10,n11,n12,n13,n14,n15,n16,n17,n18,n19]
c=[c0,c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,c14,c15,c16,c17,c18,c19]

```

```

for i in range(len(n)):
    for j in range(len(n)):
        if(i!=j):
            if(gcd(n[i],n[j])!=1): #对不同的n进行 欧几德得 算法, 以求出最大公约数
                print(i,j)
                print("p =",gcd(n[i],n[j]))

```

通过上面的代码求出p的值，接下来就是解出明文m:


```

from gmpy2 import*
import libnum
n4 = 22822039733049388110936778173014765663663303811791283234361230649775805923902173438553927805407463106104699
7739941583757040330934717613877998521683378985269805217536143078996690159313878199274218753163045915219015928238
1441775644769570104584677350862937139701305368455304218572505999679153239162642971241699499088969373280518194797
0071429309599614973772736556299404246424791660679253884940021728846906344198854779191951739719342908761330661910
4771199334285507742429104209524969296056861547994878399234243363537474421535716780645207631497932943607878217517
03543288696726923909670396821551053048035619499706391118145067
c4 = 15406498580761780108625891878008526815145372096234083936681442225155097299264808624358826686906535594853622
6873792689694684330723881497866073953964241043188208794437431123587065467539352157560783459593752996507185557596
9888785231801759750307431735674512251448180784374562642979786146301294017279761258903168671818539034538929585107
5279278516147076602270178540690147808314172798987497259330037810328523464851895621851859027823681655934104713689
5398480471630886668964736655001581790461965382107788977302095727084300676584117559598660335317004605515563809939
82706171848970460224304996455600503982223448904878212849412357

p = mpz(13258580638379860030542695730761256760422356262676419021133313624664372381104614933785296682872905247672
5552361132437370521548707664977123165279305052971868012755509160408641100548744046621516877981864180076497524093
201404558036301820216274968638825245150755772559259575544101918590311068466601618472464832499)

q = n4//p          #“//” 整除

phi = (p-1)*(q-1)

e = 65537
d = invert(e,phi)

m = pow(c4,d,n4)

print(libnum.n2s(m))    # "n2s" (数值转字符串)

```

得到 `flag{abcdbe5fd94e23b3de429223ab9c2fdf}`

12.[GUET-CTF2019]BabyRSA

```
BabyRsa.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
p+q :
0x1232fecb92adead91613e7d9ae5e36fe6bb765317d6ed38ad890b4073539a6231a6620584cea5730b5af83a3e80cf30141282c97be44
00e33307573af6b25e2ea
(p+1)(q+1) :
0x5248becef1d925d45705a7302700d6a0ffe5877fddf9451a9c1181c4d82365806085fd86fbaab08b6fc66a967b2566d743c626547203b3
4ea3fdb1bc06dd3bb765fd8b919e3bd2cb15bc175c9498f9d9a0e216c2dde64d81255fa4c05a1ee619fc1fc505285a239e7bc655ec6605d
9693078b800ee80931a7a0c84f33c851740
e : 0xe6b1bee47bd63f615c7d0a43c529d219
d :
0x2dde7fbaed477f6d62838d55b0d0964868cf6efb2c282a5f13e6008ce7317a24cb57aec49ef0d738919f47cdcd9677cd52ac2293ec5938
aa198f962678b5cd0da344453f521a69b2ac03647cdd8339f4e38cec452d54e60698833d67f9315c02ddaa4c79ebaa902c605d7bda32ce9
70541b2d9a17d62b52df813b2fb0c5ab1a5
enc_flag :
0x50ae00623211ba6089ddfae21e204ab616f6c9d294e913550af3d66e85d0c0693ed53ed55c46d8cca1d7c2ad44839030df26b70f22a85
67171a759b76fe5f07b3c5a6ec89117ed0a36c0950956b9cde880c575737f779143f921d745ac3bb0e379c05d9a3cc6bf0bea8aa91e4d5e
752c7eb46b2e023edbc07d24a7c460a34a9a
第 1 行, 第 1 列 100% Unix (LF) https://www.wjw.com.cn
```

解题思路:

1, 先观察题目给的条件, 给了 $p+q$, $(p+1)(q+1)$, e , d , 以及密文 C .

RSA的解密公式: $M=C^d \pmod n$

所以我们只要求出 n 即可。 ($n = p*q$)

所以 $n = (p+1)*(q+1) - (p+q) - 1$

2. 求 M 的值, 已知 C , d , n 后 用函数 `pow()`, 即可求出

上脚本:

```
import libnum
a = 0x1232fecb92adead91613e7d9ae5e36fe6bb765317d6ed38ad890b4073539a6231a6620584cea5730b5af83a3e80cf30141282c97be4400e33307573af6b25e2ea
b = 0x5248becef1d925d45705a7302700d6a0ffe5877fddf9451a9c1181c4d82365806085fd86fbaab08b6fc66a967b2566d743c626547203b34ea3fdb1bc06dd3bb765fd8b919e3bd2cb15bc175c9498f9d9a0e216c2dde64d81255fa4c05a1ee619fc1fc505285a239e7bc655ec6605d9693078b800ee80931a7a0c84f33c851740
e = 0xe6b1bee47bd63f615c7d0a43c529d219
d = 0x2dde7fbaed477f6d62838d55b0d0964868cf6efb2c282a5f13e6008ce7317a24cb57aec49ef0d738919f47cdcd9677cd52ac2293ec5938aa198f962678b5cd0da344453f521a69b2ac03647cdd8339f4e38cec452d54e60698833d67f9315c02ddaa4c79ebaa902c605d7bda32ce970541b2d9a17d62b52df813b2fb0c5ab1a5
c = 0x50ae00623211ba6089ddfae21e204ab616f6c9d294e913550af3d66e85d0c0693ed53ed55c46d8cca1d7c2ad44839030df26b70f22a8567171a759b76fe5f07b3c5a6ec89117ed0a36c0950956b9cde880c575737f779143f921d745ac3bb0e379c05d9a3cc6bf0bea8aa91e4d5e752c7eb46b2e023edbc07d24a7c460a34a9a

n = b-a-1

m = pow(c, d, n)

print(libnum.n2s(m)) # (n2s 将数值转化为字符串)
```

关于函数 `pow()` 的用法

得到 `flag{cc7490e-78ab-11e9-b422-8ba97e5da1fd}`