week2

babyrsa

```
(e+114514+p**k)^0x10001 mod p=(e+114514)^0x10001 mod p
```

爆破求e

```
from Crypto.Util.number import *

p=14213355454944773291
q=6184356205162070038634855117537193048606497844115920076561833974376400103329 7
c=10500213872246694649593663865603821400004347575163902508525511396508874927246 1
9068925866162502649223481924965979864527862811511564362295740651939654228 41
gift=9751789326354522940
n=p**4*q
m=0x1001
phi=(p-1)*p**3*(q-1)
e=2
while e>0:
    e+=1
    gg=pow(e+114514,0x10001,p)
    if(gg==gift):
        print(e)
        break

d=inverse(e,phi)
m=pow(c,d,n)
flag=long_to_bytes(m)
print(flag)
```

求出来e=73561

先找到第一个符合条件的e，然后再找符合e和phi互质的e+k*p

我的想法是先找到第一个符合条件的e，然后再找符合e和phi互质的e+k*p

```
n=p**4*q
phi=p**3*(p-1)*(q-1)
e=2
while e>0:
    e+=1
    now=pow(e+114514,0x10001,p)
    if(now==gift):
        print(e)
        break
while e>0:
    e+=p
    print(gcd(e,phi))
    if(gcd(e,phi)==1):

        d=inverse(e,phi)
        m=pow(c,d,n)
        flag=long_to_bytes(m)
        if b'hgame' in flag:
            print(flag)
```

但是e和phi不互质:(，把让phi=q-1，但是求出来不对，

backpack

真的是好严重的非预期，本来是想着2^20也不大，爆破就行了，结果没想到的是enc就是flag

```
from Crypto.Util.number import *

enc=871114172567853490297478570113449366988793760172844644007566824913350088148162949968812541218339
a=[3245882327, 3130355629, 2432460301, 3249504299, 3762436129, 3056281051, 3484499099, 2830291609, 3349739489, 2847095593, 3532332619, 2406839203, 4056647633, 3204059951, 3795219419, 3240880339, 2668368499, 4227862747, 2939444527, 3375243559]
for i in range(2**20):
    dec=enc^i
    flag=long_to_bytes(dec)
    print(flag)
    if(b'hgame' in flag):
        break
```

```
b'hgame{M@ster_0f ba3kpack_m4nag3ment!}\x00\x0e#'
```

直接秒了

backpack revenge

```
M=[747630795102616991263455259 79, 51725049470068950810478487507, 47190309269514609005045330671, 64955989640650139818348214927, 68559937238
S=120254819682601389900652731 4947
n = len(M)
L = matrix.zero(n + 1)

for row, x in enumerate(M):
    L[row, row] = 2
    L[row, -1] = x

L[-1, :] = 1
L[-1, -1] = S
res = L.LLL()
print(res)

```

```
[ 1 -1 -1 -1 -1  1 -1 -1  1 -1 -1 -1  1  1  1 -1 -1 -1  1  1 -1 -1  1 -1  1 -1 -1 -1  1 -1  1 -1  1 -1
1 -1  1 -1 -1 -1 -1  1 -1  1  1  1  1  0]
```

用LLL算法解决，第一行向量就是倒过来就是解

```
from Crypto.Util.number import *
import random
import hashlib
print(getPrime(96))
ans=''
bag=[747630795102616991263455259 79, 51725049470068950810478487507,
47190309269514609005045330671, 64955989640650139818348214927,
68559937238623623619114065917, 72311339170112185401496867001,
70817336064254781640273354039, 70538108826539785774361605309,
43782530942481865621293381023, 58234328186578036291057066237,
68808271265478858570126916949, 61660200470938153836045483887,
63270726981851544620359231307, 42904776486697691669639929229,
41545637201787531637427603339, 74012839055649891397172870891,
56943794795641260674953676827, 51737391902187759188078687453,
49264368999561659986182883907, 60044221237387104054597861973,
63847046350260520761043687817, 62128146699582180779013983561,
65109313423212852647930299981, 66825635869831731092684039351,
67763265147791272083780752327, 61167844083999179669702601647,
55116015927868756859007961943, 52344488518055672082280377551,
52375877891942312320031803919, 69659035941564119291640404791,
52563282085178646767814382889, 56810627312286420494109192029,
49755877799006889063882566549, 43858901672451756754474845193,
67923743615154983291145624523, 51689455514728547423995162637,
67480131151707155672527583321, 59396212248330580072184648071,
63410528875220489799475249207, 48011409288550880229280578149,
62561969260391132956818285937, 44826158664283779410330615971,
70446218759976239947751162051, 56509847379836600033501942537,
50154287971179831355068443153, 49060507116095861174971467149,
54236848294299624632160521071, 64186626428974976108467196869]
backpack=0
a=[  1  ,-1  ,-1  ,-1  ,-1  , 1  ,-1  ,-1  , 1 , -1  ,-1  ,-1  , 1   ,1  , 1 ,
-1  ,-1  ,-1   ,1  , 1  ,-1  ,-1   ,1  ,-1   ,1 , -1 , -1 , -1 ,  1 , -1  , 1
,-1  , 1  ,-1   ,1   ,1  ,-1   ,1  ,-1  ,-1  ,-1,  -1   ,1 , -1,   1   ,1,   1
,1  ,0]
for i in range(len(a)):
    if a[i]==1:
        ans='1'+ans
        backpack+=bag[i]
```

```
    if a[i]==-1:
        ans='0'+ans
        backpack+=0
print(ans)
print(backpack)
p=int(ans,2)
flag='hgame{'+hashlib.sha256(str(p).encode()).hexdigest()+'}'
print(flag)
```

检查了一下背包，对的。

```
120254819682601389900652731494 7
hgame{04b1d0b0fb805a70cda94348ec5a33f900d4fd5e9c45e765161c434fa0a49991}
```

midRSA

m0<<208就是m了

```
from Crypto.Util.number import *
n=120838778421252867808799302603972821425274682456261749029016472234934876266617
266346399909705742862458970575637664059189613618956880430078774892479256301209695
532330278722150855648119628142067607411627249527809727592760485733648456477740449
979145726062998103849874125948440719355466908199069202540040453915854 27
c=118961547254465282603128910126369011072248057317653811110746611348016137361383
017921465395766977129601435508590006599755740818071303929227578504412967513468921
191689357367045286190040251695094706564443721393216185563727951256414649625597950
957960429709583109707961019498084511008637686004730015209939219983527
e=3
m0=132921474085670873515807320829616401305433137422104094324716252817023277489632
74496942276607
m=m0<<208
print(long_to_bytes(m))
```

```
b'hgame{0ther_cas3s_0f_c0ppr3smith}\xff\xff\xff\xff\xff\x00\x00\
```

midRSA revenge

midRSA已经说了这是coopersmith，已知m高位。

```
from Crypto.Util.number import *
def phase2(high_m, n, c):
    R.<x> = PolynomialRing(Zmod(n), implementation='NTL')
    m = high_m + x
    M = m((m^5 - c).small_roots()[0])
    print(hex(int(M))[2:])
    return int(M)
e = 5
n=278143347281356719958903781547788226877138752696248431223534580596972888886405
729224862875564312417864611595132361289141766804977756196946849034980705773078102
636772802941141359297087459884069633072797670289695153058952070282821935473564 1
482741900839370115846781853510951721308892089023630028164628876169784228063328 53
553763894683600335841022582430588851748120182954601965154838192549131830794969 47
309574392848378504246991546781252139861876509894476420525317251695953355755164 78
987860294561587996570987197577082348441866563405010385256481957575695004769120 53
555990047865416002132044231458548592148974314302823330521 21
```

```
c=4562213141158670886382072030344946362447066111116217235778487290960692300679581326630186256614471315017586845026393832083328446819396981244591885718135271497722924641395307367176197417049459260756320640721253615164356311218457531865592979933552707798180577029737833915898511591140293102965517014567486989142313448351879175593054402695606133268932047481279992549021029196053703638895811367241640968795731738702808066204540874669703589986547367552570232250781470185371010
m0=999990028100335777342031068116933082326653253380390563 7
hign_m=m0<<128

m=phase2(high_m, n, c)
print(long_to_bytes(m))
```

```
6867616d657b633070707233736d6974685f5374337265307479706564 5f6d337373616733737d
b'hgame{c0ppr3smith_St3re0typed_m3ssag3s}'
```