hgame2024 week3

Crypto

exRSA

写的时候就感觉是维纳或者连分数，毕竟e位次比较小。后面发现是拓展维纳。

```
from Crypto.Util.number import *

e1=507704823781196942747311122537087612252896744705655189912361346179268800289678
839430419291761056414976625223228157699029348523968414531087693099791896007081696
882915037687595340542080958626715317171749619833686108952370183209832228450193114
288981757581676170504495170553084932792884984815864303069336314375706322058471492
589396558796704213755780726115411791635851947796464529347197506336205069030635362
749298086100843976536583762265797795806985328805630725316750988325812294988227702
166531780725330890635567047217234617117726768806495939718692610398725955158662796
540697911819348552752097674849072846016794905528953 9
e2=125268482983490053905202769239291324634591525749986257572082592978911151336541
176482157829453325290813652738603162011307933065707777350765347721689997058956412
075353038394550740030576878103811109783209889760113261069199407991609742283118247
600463702735055110656192685576971825862592343792394104827844498157323352943956763
022264168637093400329876127151519160842918210954626258210231335604153258248853472
213914969372132463617363612708467411285575956030527136125284537099484031007112776
796412185204298788975656554820864105763799714047892122976975537482924381830655009
933750400317338254966927976993624210102715995102694 01
e3=129859407575785308105193703320636583440466888566059674749410144368727203604440
404646447909809769913939709470233983574222038732842948434011440650139114636705015
598886011451086519610983482508241666976655284176683744088145729597227890201103962
450762755535058785656035094662207102192600377838492764753972834210687160886381869
947781535428176819630595816511035635788041451561575843367126788829956856326156868
539801760476833269742838963433229815211502113175975715545424889212901581226341405
711480367328938080641190483288551340547091208778959416701664216648061867103468244
940547830257334758980812478248879675504185090382762 79
c=141417606015230184211049709802459718924625917201933541490012745209823394304182 5
926028517437075316294943355323947458928010556912909139739282924255506647305696872
907898950473108556417350199783145349691087255926287363286922011841143339530863300
198239231490707393383076174791818994158815857391930802936280447588808440607415377
391336604533440099793849237857247557582307391329320515996021820000355560514217505
643587026994918588311127143566858036653315985177551963836429728515745646807123637
193259859856630452155138986610272067480257330592146135108190083578873094133114440
050860844192259441093236787002715737932342847147399
N =
178533037338380661731104178905937044641468248863164567808733525599697426157552944
666644395293527184343995528186353527680335319480097371706975662868487108328004263
113285609241336984816535940077278770315062657063415608105880642096818091465975721
261733034631256681838378404276671018272347528237474837929445368930701880103576444
785121433320147865396985352201397844403144813714640539547698227384078081619469432
167147296858208969724670208934933490512439833900187620768128686780981724164656915
502853728464029919957943490158388682216862163965973272731101659227898143158584620
497062552540667240129258151004349538218568545297 53
a = 2/5
```

```
D = diagonal_matrix(ZZ, [int(N^(3/2)), N, int(N^(3/2+a)), int(N^(1/2)),
int(N^(a+3/2)), int(N^(a+1)), int(N^(a+1)), 1])
M = matrix(ZZ, [[1, -N, 0, N^2, 0, 0, 0,-N^3], [0, e1, -e1, -e1*N, -e1, 0, N*e1,
N^2*e1], [0, 0, e2, -e2*N, 0, N*e2, 0,N^2*e2], [0, 0, 0, e1*e2, 0, -e1*e2,-
e1*e2,-N*e1*e2],[0,0,0,0,e3,-N*e3,-N*e3,N^2*e3],[0,0,0,0,0,e1*e3,0,-N*e1*e3],[0 ,
0 , 0 , 0 , 0 , 0 , e2*e3 , -N*e2*e3],[0 , 0 , 0 , 0 , 0 , 0 , 0 , e1*e2*e3]])*D
L = M.LLL()
t = vector(ZZ, L[0])
x = t * M^(-1)
phi = int(x[1]/x[0]*e1)

d=inverse(0x10001,phi)
m=int(pow(c,d,N))
print(long_to_bytes(m))

#b"hgame{Ext3ndin9_W1en3r's_att@ck_1s_so0o0o_ea3y}"
```

HNP

```
#sage
from Crypto.Util.number import *

k = 512-32
```

```
A =
[33220085552551293368213097014829969330453797924325322515795645812110726774032449
70423357912298444457457306659801200188166569132560659008356952740599371688,
82767642602648588118452115784150233439426346135220886310211994330669242910498586 0
70459606905740357613703942631549813517284943097379011217032888226163672 66,
98722917369229744564204184636011290942272319792183859851496611327924676219407225 8
07453278354053748262937913328151764587505489427570240173828815172849916 46,
40215217451425358131536699611464574066407919358447960053440738862896684648850114 1
58877557879039278247628331581306150183266661183831286275356236390468177 99,
24569151076141700493541155834378165089870615699969211988778938492838766214386066 9
52596557490584021813819164202001474086538804476667616708172536787956 586,
32185011565208485728614588311238226897020352425148035050491017799962317508750363 4
45643226000868613614146092012148222629084280910973827817708509290674042 10,
35634059873983750763276334440364921630049587148286858462028186103204393063969124 2
54203910701170698755837868193231733429511725940466520172975528135015571 59,
49147090456938630385982251245345150489933107702861050707255136674359837898475472 2
51800248243214587612623908174878616755954665135389013734221492361339263 54,
10800566112999947911006702454427389510409658644419749067440812458744391509925306 9
94806187389406032718319773665587324010542068486131582672363925769248595266,
62336492005220979079812873108919481313890969103913793527503733950362212632592877 3
03750125472285168431802401410814952521508326573371280916234455399842732 4,
49184210976284306138012655258705610412300110298188512910868629705086215290744976 0
16787749212859127455898405104596775220748875761520153569845925896498444 31,
74457333572158473700706961366536897487180280803648122639477857473532589369689781 8
34715497061663642431489721542150552248579188349377075550532461848220956 02,
93335347550492256275302842493884386940026026450479338654531598367966671989660581 7
79885001840734543861840809347275372005754575989761216673738014413959324 40,
50108548031799704458387915753211279112783116352300766390234115711484889034006101 2
12486173077738726127432289988929862002027134965703754472552586309321588 22,
60006450684625698196484610701405575211448010134901066323568363250025464008714639 5
72285811439545910053985332522184299704861154905355840717862608187731663 24,
80072609091246693818620349015561112457805059870829908043808147972003222289424326 7
39399446930624701782568673666023316123631764083563046416724594565179785 60,
10179739175373883376929532026389135792129233730601278687507041429438945598523995 7
00184622359660605910932803141785598758326254886448481046307666042835829725,
83900727677173957019262897794330556728638803360318370091191034486752323629422236 3
31293283091181582738359615674365912349227839533733197678358772668495452 92,
78750119115629678746761136806939292302838668414756411628546652931113444677094244 0
86231983709427970999646254475127971381928530091268888532835260344110075 13,
52937728110200125010201247752147701932346552103193430586486754111152104536807530 7
00428218350826196343415006808923230021189535577461169180936617694646420 68,
26137972794267745403064619313191936579998921298448321596587717173871202467956896 7
82312753714995565223960615918824314263108419747134199740458830216139877 05,
96581260121332178041266300052360735134852153908129779746600290535226652825509650 4
02882560749452468507446945195433587772529296615616362411615759370615217 11,
29825352208449776217751394063575288760193493856348117954802306779823456971835862 0
36690949980399956839739397216448875439074949638249680421993539451203675 05,
10728998487819184935718049085039753931103776226208275539816029240134007878264324 6
49856603941527986879666759668612584740013089816001783898130863881485464 1,
12099313059087422847381131486982370469901243530313464095320180880761807004891291 8
04661666467791624881306204359760787372887040249371735144790545692080686 5,
22530406527717962842662542617198057681027406530974463258697838122011711441507688 7
58859637293249157148127191382477841947526392826771234473619861170863008 9,
86500072721542830573506643115058875358412687674245450169014189895556208690911456 5
12164487232002409141438827746166789687255239143109653568756812072952424 34,
96287478291075846500141560799281088016871580290862217308839997490445328464896661 1
```

```
54739930054421928591719318827959737741313099000212873190592161059396707 57,
10846936951522093706092027908131679912432689712451920718439096706435533926996215 7
66191967052667966065917006691565771695772798711202812180782901250249613072,
16068656512279887366641270216786892999890454399983366035622329088634057784745209 1
51707667718113363196557927465909817406178235648135731184100649760819892 37,
62390636575917210977350494096108729412140786993301368265929585492124818029739731 0
43745485551849079292550315705253430075184343576904804299810167811102496 12,
18553659163871146205810299397070537010624767452355786835580637966047444480502781 3
89543595069228759675375673595756623942975799583721074842763609205677304 58]
ls = [2150646508, 1512876052, 2420557546, 2504482055, 892924885, 213721693,
2708081441, 1242578136, 717552493, 3210536920, 2868728798, 1873446451, 645647556,
2863150833, 2481560171, 2518043272, 3183116112, 3032464437, 934713925, 470165267,
1104983992, 194502564, 1621769687, 3844589346, 21450588, 2520267465, 2516176644,
3290591307, 3605562914, 140915309, 3690380156, 3646976628]
N =
113062992417749500532695471032846374144078351257772452040693675676910219288647732
075487310515928535152062323659011697780480841465208290323393282639135580 53

Ai = []
Bi = []
for ai, li in zip(A[1:], ls[1:]):
    alpha = ai * pow(A[0], -1, N) % N
    Ai.append(alpha)
    Bi.append((alpha * ls[0] - li) * pow(2^32+1, -1, N) % N)


K=2^k
M = Matrix(ZZ,33,33)
for i in range(len(Ai)):
    M[i,i]=N
    M[-2,i]=Ai[i]
    M[-1,i]=Bi[i]
M[-2,-2]=1
M[-1,-1]=K

L = M.LLL()
ss = list(L[0])
assert ss[-1] == K
ss = ss[-2:-1] + ss[:-2]
B = [ZZ(hi + si * (2^32+1)) for hi, si in zip(ls, ss)]
x=B[0]*(pow(A[0],-1,N))
print(long_to_bytes(int(x)))

#b'\xff\xff\xff\xff\xff\xff_hgame{H1dd3n_Numb3r_PrObl3m_has_diff3rent_s1tuatiOn}\
xff\xff\xff\xff'
```

matrix_equation

一把过，爽!

```
#sage

import hashlib

k1=73715329877215340145951238343247156282165705396074786483256699817651255709671
```

```
k2=6136197066226986973827032852389776540844390719831363241006845422371782427683 7

m=Matrix(ZZ,[[2^256,0,0],[k1,1,0],[k2,0,1]])
m=m.LLL()

temp,q,r=m[0,0],m[0,1],m[0,2]

p=(temp-q*k1-r*k2)//2^256
flag='hgame{'+hashlib.sha256(str(p+q+r).encode()).hexdigest()+'}'
print(flag)

#hgame{3633c16b1e439d8db5accc9f602f2e821a66e6d80a412e45eb3e1048dffbb0e2}
```