hgame2023 week1 wp

Crypto

ezmath

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

enc=b"\xce\xf1\x94\x84\xe9m\x88\x04\xcb\x9ad\x9e\x08b\xbf\x8b\xd3\r\xe2\x81\x17g\
x9c\xd7\x10\x19\x1a\xa6\xc3\x9d\xde\xe7\xe0h\xed/\x00\x95tz)1\\\t8:\xb1,U\xfe\xde
c\xf2h\xab`\xe5'\x93\xf8\xde\xb2\x9a\x9a"
D = 114514

def solve_pell(N, numTry = 1000):
    cf = continued_fraction(sqrt(N))
    for i in range(numTry):
        denom = cf.denominator(i)
        numer = cf.numerator(i)
        if numer^2 - N * denom^2 == 1:
            return numer, denom
    return None, None

def pad(x):
    return x+b'\x00'*(16-len(x)%16)

x,y=solve_pell(D)
KEY=pad(long_to_bytes(y))[:16]
cipher= AES.new(KEY,AES.MODE_ECB)
m=cipher.decrypt(enc)
print(m)
```

就是解佩尔方程，用连分数找特解

b'hgame{G0od!_Yo3_k1ow_C0ntinued_Fra3ti0ns!!!!!!!}\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'

ezPRNG

```
]=
['1111110110111011110000101011010001000111111001111110100101000011110111111100010
0001111101101111000010010001011010111101111000100101000000111111011011101010110
1110000000111100001000111011110110110001001011001101001011100010100011011011100
0100010001111001010100101110111011110111001101100101111101101010101100001101100011
0110111110011010101111001011001100010110100101011100111010011001110000111101110000
0011011100000011111000001000001011111000101101110011100110100000110111101100110000
0011010111111111010110011010111010101001000010011110110011111011010101111011101001
0100101101111111010011101000110101111101111000110011111111001011000010010010010110
0101011100101010011010101010111101110100111011100001001011110101101011111110001111
1111100100000000011100111001000010111111101001110110001010011010011100100100011000
0110000011010001110100100000101101111101011000000101000001110001011001010010001000
0110000001000100100100101110100111111111011100100100100101111111001110000111110110
0011110011111100101001001100010',
'0010000000001010111100001100011101111101111000100100111010101011100101100110010111
1010110001110101000000110000011000000001100000011010111111101110010011011101101000
0010001111100011100100010100111001011001000100011001010101111001110100001111110110
0101100001111000110101111100011011100001100011001110010010110011111000001001001011
1100101110111000101101111111111011010100010111011000010010101110111010000011010000001
0001010100001011110100100001100000000011101001010101011110111010111110110010001010
0010001100110010101011011000101001000101011011101101111110101110011100110111111111
1101001110111101001001111001111111101001100111111110110001000111100010111000101110
0001101101111110111010111010011100001110000101011011110001100101101001101011100011
1010110011010001110110101110100011101100010011011000110011010101011001001101111100
0011111010011110111000010001000011110001011100001000001000111111011010000100011101
1010010011011001011011101001111110101111000001110101010011010101111000011010111011
1101101011011000000100001100011',
'1110110110010001011100111111011111011100111110101001100111110010000100011100110101
1101010001011111010111010111101011110010110001001100100101110100010101100011011100
0001000010100100010011101011000101000011111011011100001100110001000110100001000113
1111110000010111100010010100000000100100100111011100001001110011100010010110101111
1101011110110110100111011101011111011001100100001000101010001001011011010101110000
0010111110010011001111000100100111110010111100111101101101011100100111101000110011
1000110000110000011000001111101010010111100000010101111101000011111000001011111000
1000001001011101011010010101010100111110010101110001100100101100010101010100110110
0001011000001000111001111001110011100011010101011101001101000000110000101100001111
0110100000001111100010111111010111100110000011011000100100110111010011001111101100
1
0110001100010100111010111100100001011001011111011101100101011010000001010010110000
0000111000111000010000000100111110001101001100000000110111011111010011111100010111
0110000001000100101001100000 1',
'0001101010101010100001001001100010000101010100001010001000100011101100110001001
0000100111000011010001010111101011011100110101101110111000001100100010010010100001
1101110100011100100101001110001000101011011101110010011111011100101010010111010100
0010011111010111001001011010000100001001000110111100111010001000101110110011101111
0101110110010010101101010100010100100010111001101111111101100111111111000000000111
0000001001100011000100011010101000101100001010100011000010100111010101011101101001
1011101100101001110001010100110011000011010110001000010011010111010000110100101101
1111001110011001100101011010010101011111011011110000011101000111110111000000000001
1101101110100001100101001011100111011100010011101111010010100010001101110110001111
1100010111011011011111100111100000001110001100001000010100101100110111010100001010
0100100010011001000010100111110010100000101101101001111000110100000110111101010011
0100110001010000011100001111010101010001101100111000101111011101011101101010110111
0000011000000101001010111101 1']
```

```
for rrr in range(4):
    key =  l[rrr][0:32]
    R = ''
    tem = key
    i=0
    for i in range(32):
        output = '?' + key[:31]
        ans = int(tem[-1-i]) ^ int(output[-1]) ^ int(output[-4]) ^
int(output[-8]) ^ int(output[-11]) ^ int(output[-15]) ^ int(output[-20]) ^
int(output[-25]) ^ int(output[-28])
        R += str(ans)
        key = str(ans) + key[:31]
    R = str(hex(int(R[::-1],2))[2:])
    print(R)
```

得到 fbbbee82 3f434f91 93379078 80e4191a，再按格式连起来就好

hgame{fbbbee82-3f43-4f91-9337-907880e4191a}

考点是LFSR，正好mask最高位是1，所以能这样写，一位一位倒退，不然我估计要用深搜广搜啥的。

原题好像是2018 CISCN 线上赛 oldstreamgame

ezRSA

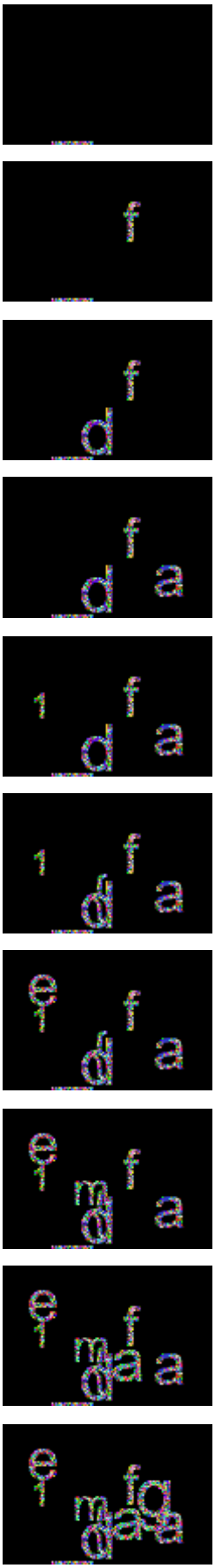
```
from Crypto.Util.number import *

leak1=14912717007361127196818257675129033155901844180572531042609541283758922767075754074392986585365039983910283843150720074472493965946320015801246967697998769641905090084279822566586181233111363289243874272420291641606026658159016906386768829928898573410412763223217565735269789838344132347745065817972772890866
leak2=11612299271467091538130991696749043648902000117288064416717991546702179489292797727208059664178556911913425903752238833519804315220615025910348557455881642474020473621555193348258394195999462535658120105453452939578174433863102142370317114645666343295584359854812259330878224522079201871650853849740257670946
c=10529481867532520034258056773864074017027019578041866245400647840230251661652999709715919620810933437191661180003295923273655675729588558899592524235622728816065501918076120812236580344991140980991532347991252705288633014913479970610056845543523591324177567061948922552275235486615514913932125436543991642607028689762693617305246716492783116813070355512606971626645594961850567586340389705821314842096465631886812281289843132258131809773797777049358789182212570606252509790830994263132020094153646296793522975632191912463919898988349282284972919932761952603379733234575351624039162440021940592552768579639977713099971
#print(isPrime(leak1),isPrime(leak2))#True True
p=leak1
q=leak2
n=p*q
phi=(p-1)*(q-1)
e=0x10001
d=inverse(e,phi)
m=pow(c,d,n)
print(long_to_bytes(m))
```

leak1，leak2正好是p，q

b'hgame{F3rmat_l1tt1e_the0rem_is_th3_bas1s}'

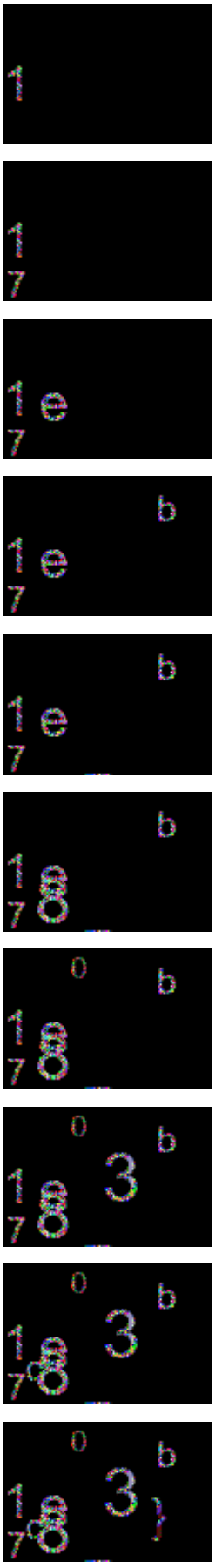
strange_image

抽张图片每个异或，分成两组



hgame{1adf_

第一个字母不会出现，但知道flag格式，问题不大



17eb_803c}

hgame{1adf_17eb_803c}

Misc

签到

嗯，签到

SignIn

ps拉了一下