

Cryptohack刷题记录(四) Mathematics部分 Brainteasers

Part 1 writeup

原创

[Mr_AgNO3](#) 已于 2022-02-05 21:16:22 修改 186 收藏

文章标签: [python](#) [密码学](#) [数学](#)

于 2022-01-25 01:29:33 首次发布

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/u010883831/article/details/122678117>

版权

文章目录

Mathematics

Brainteasers Part 1

1. Successive Powers
2. Adrien's Signs
3. Modular Binomials
4. Broken RSA

Mathematics

Brainteasers Part 1

1. Successive Powers

给了一个数组

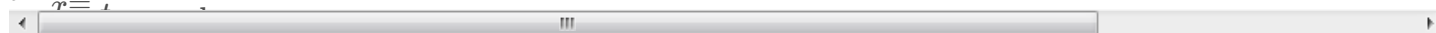
是 x 的连续阶乘对 p 取模的结果

p 为三位素数

记 $t=[588, 665, 216, 113, 642, 4, 836, 114, 851, 492, 819, 237]$

则有

$t *$



```
t = [588, 665, 216, 113, 642, 4, 836, 114, 851, 492, 819, 237]
pmin = max(t)+1

for p in trange(pmin,0x3f3f3f3):
    try:
        x = [(t[i]*invert(t[i-1],p))%p for i in range(1,ll)]
        if len(set(x)) == 1:
            print(p,x)
    except:
        pass
```

```
C:\Users\Mr_AgNO3>python "D:\Desktop\Successive Powers.py"
0% | 0/66318495 [00:00<?, ?it/s]9
19 [mpz(209), mpz(209), mpz(209), mpz(209), mpz(209), mpz(209), mpz(209), mpz(209), mpz(209), mpz(209), mpz(209)]
```

结果p=919, x=209

flag为 `crypto{p,x}`

2. Adrien's Signs

加密函数

```
from random import randint

a = 288260533169915
p = 1007621497415251

FLAG = b'crypto{????????????????}'

def encrypt_flag(flag):
    ciphertext = []
    plaintext = ''.join([bin(i)[2:].zfill(8) for i in flag])
    for b in plaintext:
        e = randint(1, p)
        n = pow(a, e, p)
        if b == '1':
            ciphertext.append(n)
        else:
            n = -n % p
            ciphertext.append(n)
    return ciphertext
```

CSDN @Mr_AgNO3

随机量为n, 为指数部分

使用离散对数 `sympy.discrete_log`

```
exec('c'+open("D:\\Downloads\\output_80fc6398d2fd9f272186d0af510323f9.txt").read())
from sympy import discrete_log as dl
from tqdm import *
ff = ""
for i in trange(len(c)):
    cc = c[i]
    try:
        r = dl(p,cc,a)
        ff += '1'
    except:
        r = dl(p,-cc%p,a)
        ff += '0'
print(ff)
from number import *
print(l2b(int(ff,2)))
```



```

# Sage
n = 2777285740987525752941599091121421197584430718443024145189940783875050302432336789554098160658670998598000343
5082116995888017731426634845808624796292507989171497629109450825818587383112280639037484593490692935998202437639
626747133650990603330945135315052099542730044735671932355350619429917509327258086792499646670907234803979167153
2087686780371930131344000507505648120385901049083659971752366419711205320674523590861048490771521043641301554667
1034478367679465233737115549451849810421017181842615880836253875862101545582922437858358265964489786463923280312
860843031914516061327752183283528015684588796400861331354873
e = 16
ct = 113031747618944311467356975694891347472349751441621721624016745672730348313919369163972340683461154591346024
4396360406367937928591930222571905019359017924019142961207213162977994837982103961041509978435107344321891135632
8815458050694493726951231241096695626477586428880220528001269746547018741237131741255022371957489462380305100634
6004992044357632013711887694460549257481519871756566773427790434350470481305991230815810363627122086927480346202
4559044876240654380406993587312316158275679951722666683531658889630692665932105427650771441487668473812142112417
7324568084533020088172040422767194971217814466953837590498718

from number import *

R.<x> = Zmod(n)[]
f = x^2-c
r8 = [i[0] for i in f.roots()]
r4,r2,r = [],[],[]

for rr8 in r8:
    f = x^2 - rr8
    r4 += [i[0] for i in f.roots()]

for rr4 in r4:
    f = x^2 - rr4
    r2 += [i[0] for i in f.roots()]

for rr2 in r2:
    f = x^2 - rr2
    r += [i[0] for i in f.roots()]

for m in r:
    print(l2b(m))

```

得到

```

b"Hey, if you are reading this maybe I didn't mess up my code too much. Phew. I
really should play more CryptoHack before rushing to code stuff from scratch aga
in. Here's the flag: crypto{m0dul4r_squ4r3_r00t}"
b"\xb7p\x09\xcc\x8d\x02\xeb\x80\x09\xfb\xed\xfa\x04\t\x09\x01\x07\x16*\xf5\xb

```

flag

是 `crypto{m0dul4r_squ4r3_r00t}`