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

文章目录

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
```


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)))
```

```
01100011011100100111100101110000011101000110111101111011011100000011010001110100011101000110010101110010011011100111001101011111001100010110111001011111011100100110010100110101011001000111010100110011011100111001101111101
b'crypto{p4tterns_1n_re5idu3s}'
```

---

3. Modular Binomials

Rearrange the following equations to get `p,q`

```
N = p*q
c1 = (2*p + 3*q)**e1 mod N
c2 = (5*p + 7*q)**e2 mod N
```

给了N,e1,e2,c1,c2，要求得p,q

Factordb能分解

149055622578427140579327241295750028254053935026508697671159426064086003433803278662589821 Factorize!

Result:

status (?)	digits	number
FF	617 (show)	1490556225...73<617> = 1122740001...73<309> · 1327605878...01<309>

---

4. Broken RSA

给了n,e,c
注意到e为16，并且有

$$m \equiv c \mod n$$

> If you think you're doing the right thing but getting garbage, be sure to check all possible solutions.

注意，这个方程会有两个解，两个都要考虑并进行下一步计算。

第一步能得到 m 的值，记为r8，以此类推,有

$$(m) \equiv r8 \mod n$$

```
# Sage
n = 27772857409875257529415990911214211975844307184430241451899407838750503024323367895540981606586709985980003435082116995888017731426634845808624796292507989171497629109450825818587383112280639037484593490692935998202437639626747133650990603333094513531505209954273004473567193235535061942991750932725808679249964667090723480397916715320876867803719301313440005075056481203859010490836599717523664197112053206745235908610484907715210436413015546671034478367679465233737115549451849810421017181842615880836253875862101545582922437858358265964489786463923280312860843031914516061327752183283528015684588796400861331354873
e = 16
ct = 11303174761894431146735697569489134747234975144162172162401674567273034831391936916397234068346115459134602443963604063679379285919302225719050193590179240191429612072131629779948379821039610415099784351073443218911356328815458050694493726951231241096695626477586428880220528001269746547018741237131741255022371957489462380305100634600499204435763201371188769446054925748151987175656677342779043435047048130599123081581036362712208692748034620245590448762406543804069935873123161582756799517226666835316588896306926659321054276507714414876684738121421124177324568084533020088172040422767194971217814466953837590498718

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}"
```

flag

是 crypto{m0dul4r_squ4r3_r00t}