SBCTF Week1 Crypto Official Writeup

Week1当然要轻松+愉快了，出了些ez题

hard_pic_encode (200pts 33solves)

hard<ez<baby，所以这是签到题

xor函数都给写好了，复制粘贴一下3s就打完了

```python
from PIL import Image
import numpy as np
from random import getrandbits
from Crypto.Util.number import *

flagImg = Image.open('enc.png')
width = flagImg.width
height = flagImg.height

def xorImg(keyImg, sourceImg):
    img = Image.new('RGB', (width, height))
    for i in range(height):
        for j in range(width):
            p1, p2 = keyImg.getpixel((j, i)), sourceImg.getpixel((j, i))
            img.putpixel((j, i), tuple([(p1[k] ^ p2[k]) for k in range(3)]))
    return img

noise = Image.open('noise.png')
dec = xorImg(noise, flagImg)
dec.show()
```

SBCTF{ez_x0r_s1gn1n#0919}

SuperBag

很基本的背包密码，注意到 `array_1` 是个等比数列，符合超递增性质，所以我们想在 `array_1` 的空间下解这个背包问题，自然只需要将对应的密文 $c*w^{-1}\ mod\ p$，然后写个贪心算法就好了，后来看选手（小轩）的wp，因为这里 `array_1` 是个 `2` 为公比的等比数列，所以 $c*w^{-1}\ mod\ p$之后的结果 `(bin[:2])[::-1]` 实际就是flag的二进制，很好，贴选手exp

```
from Crypto.Util.number import *
import binascii
from sympy import mod_inverse

w =
84887833870795852548188125601100630597699019501917299910881538466689242164177650
091593031898
p =
12864636377246591635405653721103587864723923166591479127850379224514067042046813
3896396259473
ct =
10509542148387162770578562958650402550911746202322918796657059787475092866386414
39979384069318

c = ct%p
w_inv = mod_inverse(w, p)
x = (c * w_inv) % p
flag = long_to_bytes(int((bin(x)[2:])[::-1],2))
print(flag)
```

baby_pic_encode

本意是一个 `Pell方程` 求 `(x,y)` + `logistic` 加密，时间略仓促，果然在 `logistic` 的参数设置上出现了点问题，在参数选择优秀的情况下，是必须要准确求解 `(x,y)` 的。这里看选手（vivi）的

wp可以近似拟合 `(x,y)` 的数值然后decrypt，得到的图片跟原图虽然有差距但是不影响读flag，多加密几轮就好了，害

exp

```
from Crypto.Util.number import *
import cv2
import numpy as np
import matplotlib.pyplot as plt

'''
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

N =
739060350418548486232529224397852213699569319882618667059876802401//91227264531
1126825454808203115116214167300498169578336170171
solve_pell(N)
'''

x,y=
(119780432163041267719168714551168498439647406105769642732022219099595249690227
4598087506312088418718100155741819887119183590814770815821770641423452872319026
5258954049265874593430746473325952394221300373688723100184053565353581509311388
7919889344650153988103931799605180983935,

1330786082102509628866299094115430279618376851716326235808414260747692640205932
1733093521932622405302531075970287389387851550944194933962404774405698827248997
3951248841981789556700464122886498064174099830150635364015305631114544842915649
813828048355083811328172463684697552)

def decrypt(img,key):
    [w,h]=img.shape
    x1=key[0]
    x2=key[1]
    x3=key[2]
    u1=key[3]
    u2=key[4]
    u3=key[5]
    n=key[6]
    img_tmp=np.zeros((w,h))
    for k in range(n):
        for i in range(w):
            for j in range(h):
                x1=u1*x1*(1-x1)
                x2=u2*x2*(1-x2)
                x3=u3*x3*(1-x3)
                r1=int(x1*255)
                r2=int(x2*255)
                r3=int(x3*255)
                img_tmp[i][j]=(img[i][j]-((r1+r2)^r3))%256
        x1=key[0]
        x2=key[1]
        x3=key[3]
    return img_tmp

key=[]
key.append(round(x/y*(0.00030),16))
key.append(round(x/y*(0.00050),16))
key.append(round(x/y*(0.00070),16))
key.append(round(y*3000/x,16))
key.append(round(y*3200/x,16))
key.append(round(y*3600/x,16))
key.append(1)

img_encrypt=cv2.imread("encrypt.png")
img_gray = cv2.cvtColor(img_encrypt,cv2.COLOR_RGB2GRAY)
img_decrypt=decrypt(img_gray,key)
cv2.imwrite('decrypt.png',img_decrypt)
```

broken_pem

`ez_pic_encode` 没出完，临时把这个出了好久的题拿来了，结果开赛前的周末NSS round16出了一个完全一模一样的题，，，很好

选手（*Libr）的wp写的比我好，贴他的wp👇

读了pycryptodome的源码，DER格式的内容大概是 `[type][length_type][length][content]` ，type的内容可以参考这里。

因为RSA PEM内容肯定是整数序列，内容如下：

```
RSAPrivateKey ::= SEQUENCE {
    version           Version,
    modulus           INTEGER,  -- n
```

```
          publicExponent    INTEGER,  -- e
          privateExponent   INTEGER,  -- d
          prime1            INTEGER,  -- p
          prime2            INTEGER,  -- q
          exponent1         INTEGER,  -- d mod (p-1)
          exponent2         INTEGER,  -- d mod (q-1)
          coefficient       INTEGER,  -- (inverse of q) mod p
          otherPrimeInfos   OtherPrimeInfos OPTIONAL
      }
```

后面序列有四个数，所以可以得到q，题目也给了e，根据rsa的原理，有

$$
\begin{gathered}
ed \equiv 1 \pmod{\phi(N)} \\
\implies ed \equiv 1 \pmod{\phi(p)\phi(q)} \\
\implies ed \equiv 1 \pmod{\phi(q)} \\
\implies ed \equiv 1 \pmod{q-1} \\
\implies d \equiv e^{-1} \pmod{q-1} \\
\text{又} m \equiv c^d \pmod{N} \implies m \equiv c^d \pmod{p \cdot q} \\
\implies m \equiv c^d \pmod{q} \\
\implies m \equiv c^{e^{-1} \pmod{q-1}} \pmod{q}
\end{gathered}
$$

可知，在q已知的情况下能够求出m

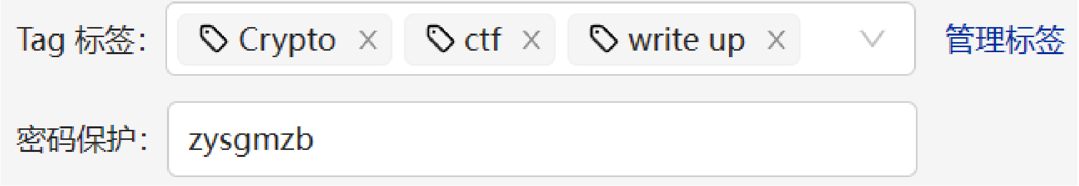
编写exp.py如下

```
from binascii import a2b_base64, hexlify
from Crypto.Util.number import long_to_bytes as l2b

pem = (
    a2b_base64(
        """
1ixI9xAcwhdVVjzfp55wYLPya5DWWP9zmpMMxYV0Zb74j/r/+ajucrs15/+rG2Rf
BHBMSTFwn4mbL60OfhReOuj3T7cNBYYYHgFGC5kANsa/HVKQegWebJNNAoGBANRg
g8lUzD5t2iE1wrOtzepOCCGNmTeoJckArrsOWBRbJ7U95FJy9pz7beEmH8Upfgjt
ErHXRALLzeKhrKf18nsHg2YsvK5zSD149g+iPhL1JPi/x2BndcYMgBuicMR7eZ59
jDVs72sELL+5tsunUsvu51VHaNi+JwRLHMOe2WgZAoGAKbCaUZR1Dit2zkiIkeg7
WQCdadFnVGoyFOGNlDYLSB4lBE5tqnXfUzQiqTzMnYmynj1VhBaOF3uw4gKWxzkB
aGvDhglVo2LsMrcEMQcv8uqRYZ/50Y4yDcyas1RhsDJ8PrVJOeom7xf5P/GXClIO
mtmiFnna+NzuCdextFZnE+ECgYEAzflBN11XrWCLQtRKHkt9vzWo6ynSpMkexGA2
FtMll7CExWHedBxtk/jCK6/29hh01SFglTyrCG8zIg8dTdTaNHon9UuEP0ktkfkj
5Cu9OlOpZNtS+eu9rLPo92RHLDh4zr8C4bniRg9JezUZ1VBVm9X7ZJkaVcOuQZq7
rfn87tkCgYEAgnKtFAEEOq6UqSgzbYSTPsgpHlQy8ZAzJBZhupevBxXFQyjl6UCD
KSeDSvjgpHngIVEdrpm8xGmHpaYGhdvUBX3RmFv5wg/Lhb5Y/aMu3Tpv2hhysmv1
thD5ts5oRIwKrl0ZlrQPybnYLHMixky5R9JJohRv8Dmgp15afJ4PEHc=
-----END RSA PRIVATE KEY-----
""".replace(
            "-----END RSA PRIVATE KEY-----", ""
        ).replace(
            "\n", ""
        )
    )
    .hex()
    .split("0281") #02表示下面的type是整数 81表示接下来一位是长度
)[1:] #第一块是无用块。
pem = [int(i[2:], 16) for i in pem] #去掉长度标识转成整数。
q = pem[0]
e = 0x10001
d = pow(e, -1, q - 1)
c = int(

 """64cf9253ce6f8bb37ad43cbb473a0577d036144d5dc9ce0ae2fa5a485950096b0b78b06f06b
cc60b6f92eddc34ff1ea1e1573b82912c4aea70c645bf11c9bf36a291ff9793390051e412ab209e
b199cf0ea0c100e4c7af7a650848c14ec44b7d78a13da503a30eb8ef37e432bcd587bc7cebfc4d8
9aaaf4b8f3f84c5947a623375008a8d211e97057923c115e320ccaf9cb9f839a0c03c8d337b061c
a58c8ccf9d3fdbb121fce009b313ee7381a124b80ff9f1ed0217cca2cf58306e9a99baa7aafcfab
90164ab45fd37f240a584c5631a5325249b371551c8daaab8882cd01b439b383d7c557534a99e7a
f5e64afdf6d22d0fb6f67944996aa874150b9deffb""",
    16,
)
m = pow(c, d, q)
print(l2b(m).decode())
```

希望选手们都这样写wp，方便我 `Ctrlc+v`

a bit limit

参考之前博客写过的一篇内容，RSA大冒险2这个题，不过比赛期间把这篇博客锁了，但是密码是zysgmzb😆



Tag 标签: Crypto × ctf × write up × 管理标签

密码保护: zysgmzb

后来发现还是下手轻了， `smallroot` +几位爆破一下就行，不过是week1也就无所谓了，或许week4会上一个终极revenge版本

strange_pic_encode

上周跟r3kapig打国际赛看到的挺好玩的题，利用 `Peano` 曲线加密了一张图片，写个decode就好

```
def Peano(k, x, y):
    if k == 0:
        return 1
    lens = 3 ** k
    cnt = (3 ** (k * 2)) // 9

    if x < lens // 3:
        if y < lens // 3:
            return Peano(k - 1, x, y)
        elif y < lens * 2 // 3:
            return cnt + Peano(k - 1, lens // 3 - 1 - x, y - lens // 3)
        else:
            return cnt * 2 + Peano(k - 1, x, y - lens // 3 * 2)
    elif x < lens * 2 // 3:
        if y < lens // 3:
            return cnt * 5 + Peano(k - 1, x - lens // 3, lens // 3 - 1 - y)
        elif y < lens * 2 // 3:
            return cnt * 4 + Peano(k - 1, lens * 2 // 3 - 1 - x, lens * 2 // 3
- 1 - y)
        else:
            return cnt * 3 + Peano(k - 1, x - lens // 3, lens - 1 - y)
    else:
        if y < lens // 3:
            return cnt * 6 + Peano(k - 1, x - lens * 2 // 3, y)
        elif y < lens * 2 // 3:
            return cnt * 7 + Peano(k - 1, lens - 1 - x, y - lens // 3)
        else:
            return cnt * 8 + Peano(k - 1, x - lens * 2 // 3, y - lens * 2 // 3)

import itertools, numpy as np
from PIL import Image

c = np.array(Image.open('encrypto.png'))[::-1, :, :]
d = {Peano(6, x, y): (y, x) for x, y in itertools.product(range(729),
repeat=2)}
e = np.array([c[d[i+1]] for i in range(729*729)]).reshape(729, 729, 3)
Image.fromarray(e).show()
```

SBCTF{emmm_1s_author'5_w1f3} SBCTF{emmm_1s_author'5_w1f3}

SBCTF{emmm_1s_author'5_w1f3} SBCTF{emmm_1s_author'5_w1f3}

SBCTF

By Spirit Team & BXS Team

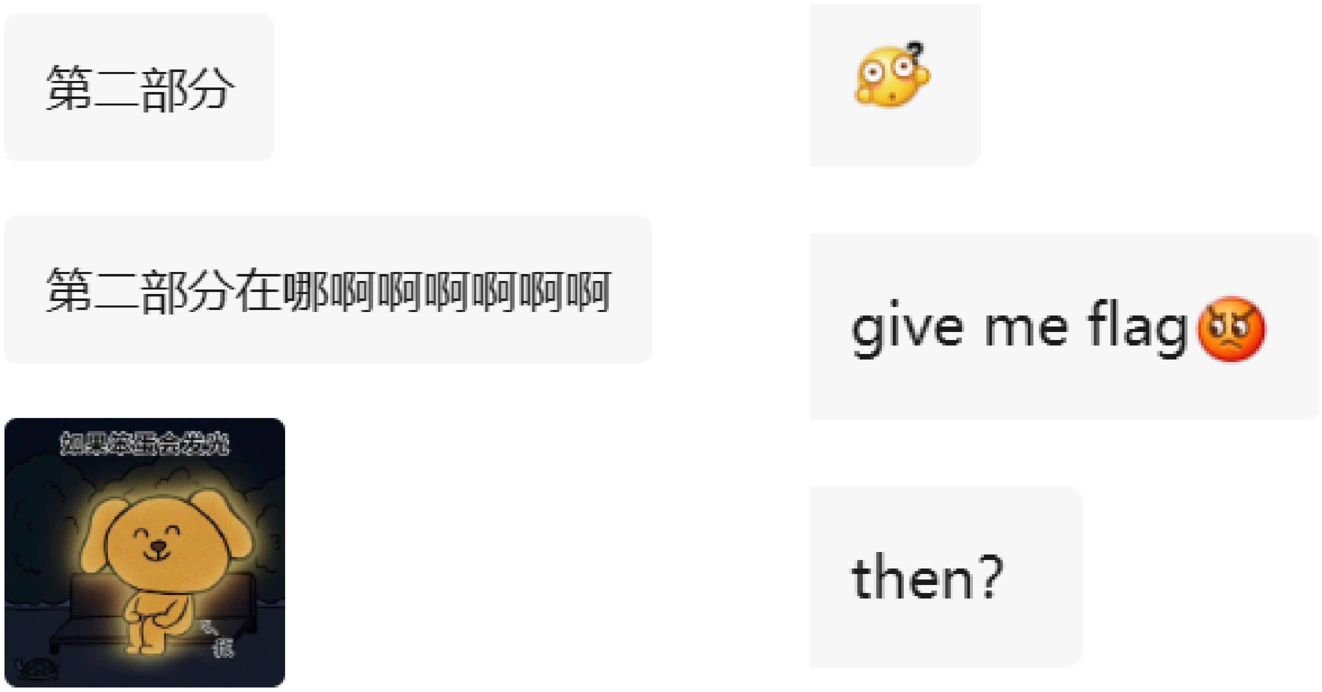
SBCTF{emmm_1s_author'5_w1f3} SBCTF{emmm_1s_author'5_w1f3}

SBCTF{emmm_1s_author'5_w1f3}

Signin——彩蛋

这题真的很难蚌，想法挺自然的，在公众号二维码图片后面明文隐写了BXS的b站账号和要发的内容（signin），结果😰👇

1.询问型



第二部分

😵

第二部分在哪啊啊啊啊啊啊

give me flag😡

then?

2.rce型

```
flag
```

```
cat flag
```

```
1
```

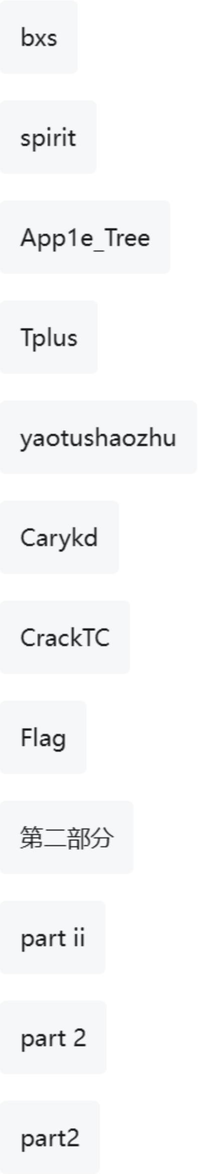
```
Can you give me flag
```

😵

```
/flag
```

```
Spirit
```

3.暴力枚举型



bxs

spirit

App1e_Tree

Tplus

yaotushaozhu

Carykd

CrackTC

Flag

第二部分

part ii

part 2

part2

4.破防型

flag

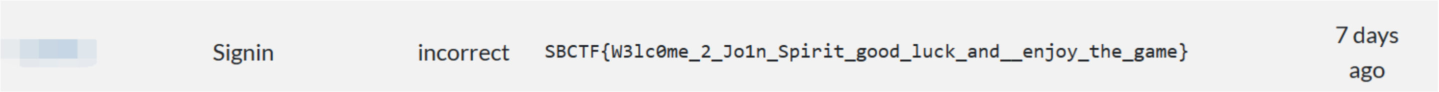
flag

🔪🔪

5.App1e_Tree的钓鱼（并且成功）行为

SBCTF{W3lc0me_2_Jo1n_Spirit_good_luck_and__enjoy_the_game}

星期一 13:08



[redacted]	Signin	incorrect	SBCTF{W3lc0me_2_Jo1n_Spirit_good_luck_and__enjoy_the_game}	7 days ago

太幽默了

😭😭😭😭😭

嘿嘿

👿👿

SBCTF{why_you_can_not_signin}

😢😢

虾仁猪心啊

😭😭😭

以上素材均来自选手，匿名发布，如有侵删