Week_1 做题记录

- **Crypto**
  - Dancing Line
  - Easy RSA
  - English Novel
  - Matryoshka
- **Misc**
  - 这个压缩包有点麻烦

Crypto

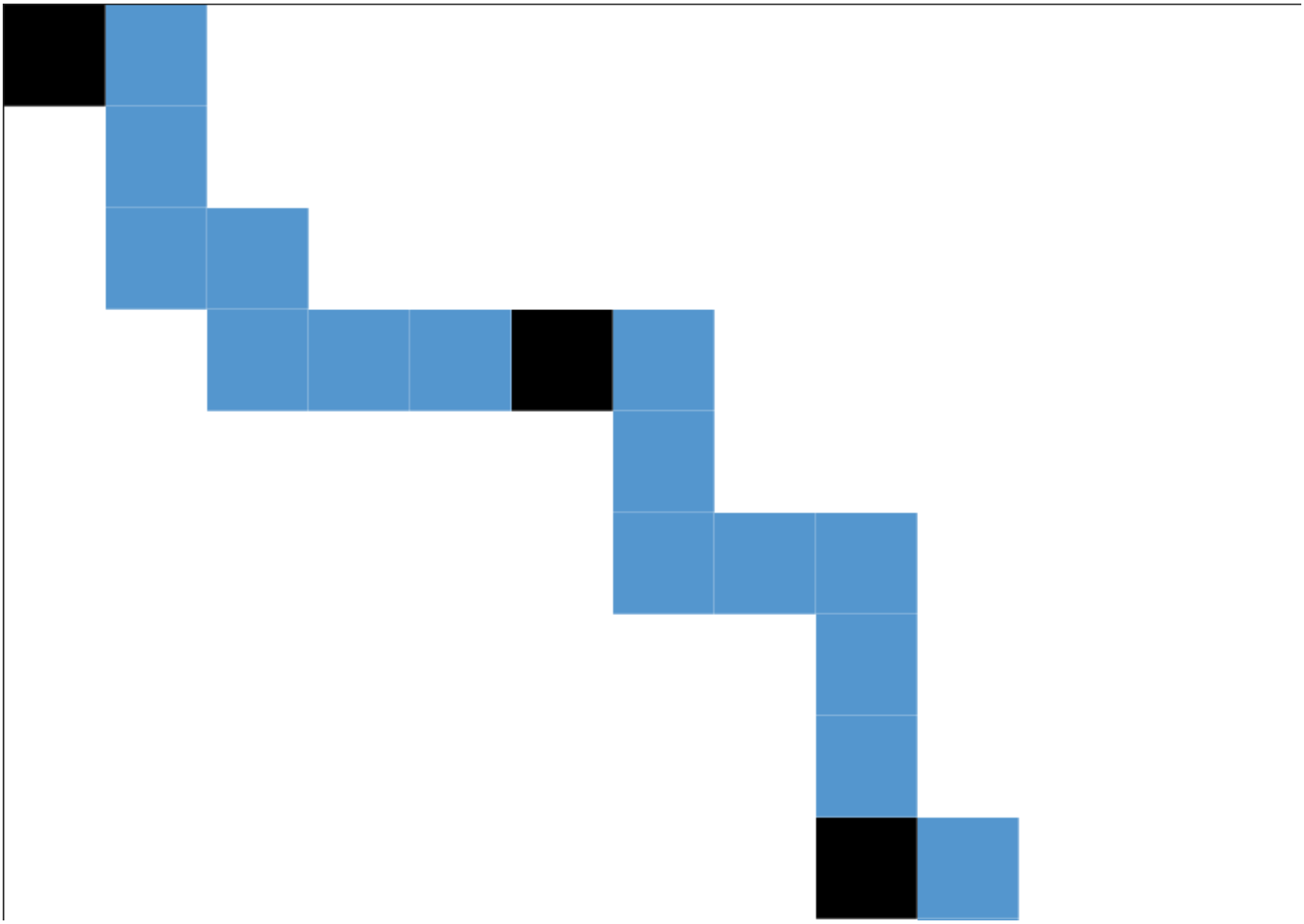
Danceing Line

- **思路**

一开始的时候我是以图片分析为主，查看了图片的图层以及像素块的位置等，发现并没有什么规律或者隐藏信息；

查看图片源文件以及hex排除了图片文件本身具有的信息隐藏了flag（被压缩包搞出的后遗症）；

在PS里分析图片发现每2个黑块间有7个蓝块，蓝块仅存在向下和向右延伸的情况；



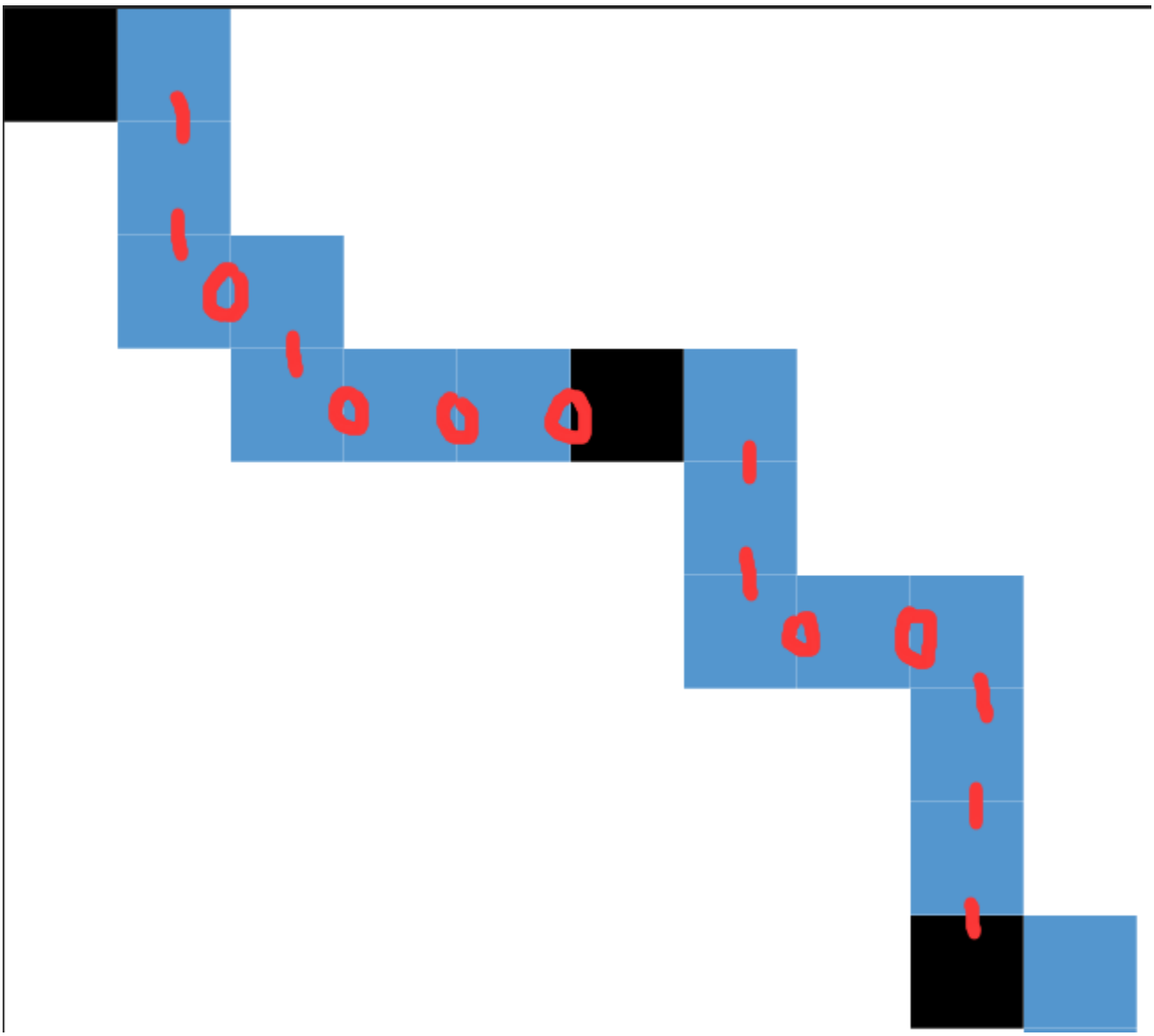
据此分析，蓝块可能代表一个由两种符号和分隔符组成的密文，猜测二进制和摩斯代码，但是摩斯代码加密后的密文长度会改变，排除；

那就验证一下二进制，7位二进制，那后面可能是ASCII码转换，逆向推理hgame查询它转化后的二进制码得到如图；

hgame

h	ASCII	16进制 68	10进制 104	2进制 1101000	g	ASCII	16进制 67	10进制 103	2进制 1100111
a	ASCII	16进制 61	10进制 97	2进制 1100001	m	ASCII	16进制 6D	10进制 109	2进制 1101101
e	ASCII	16进制 65	10进制 101	2进制 1100101					

结合图片以及得到的二进制可得到一下规律；



根据间隙，横为1，竖为0，然后我把二进制手打在了txt上

- **代码**

```
"""
user:鸢柒
purpose:dancing line
time:2022.01.27__19:50
"""
#读取二进制文件
with open('F:\\Face_to_the_prison\\Crypto\\HGAME\\Dancing Line\\line.txt','r') as f:
    bin_line=list(f.read().split('\n'))
#打印二进制
print(bin_line)
oct_line=[]
#转化为十进制
for i in bin_line:
    oct_line.append(int(i,2))
print(oct_line)
flag=''
#ASCII转化为flag
for i in oct_line:
    flag=flag+str(chr(i))
print(flag)
```

此题未能及时提交，故不能验证flag的正确性（手打的二进制可能存在错误）

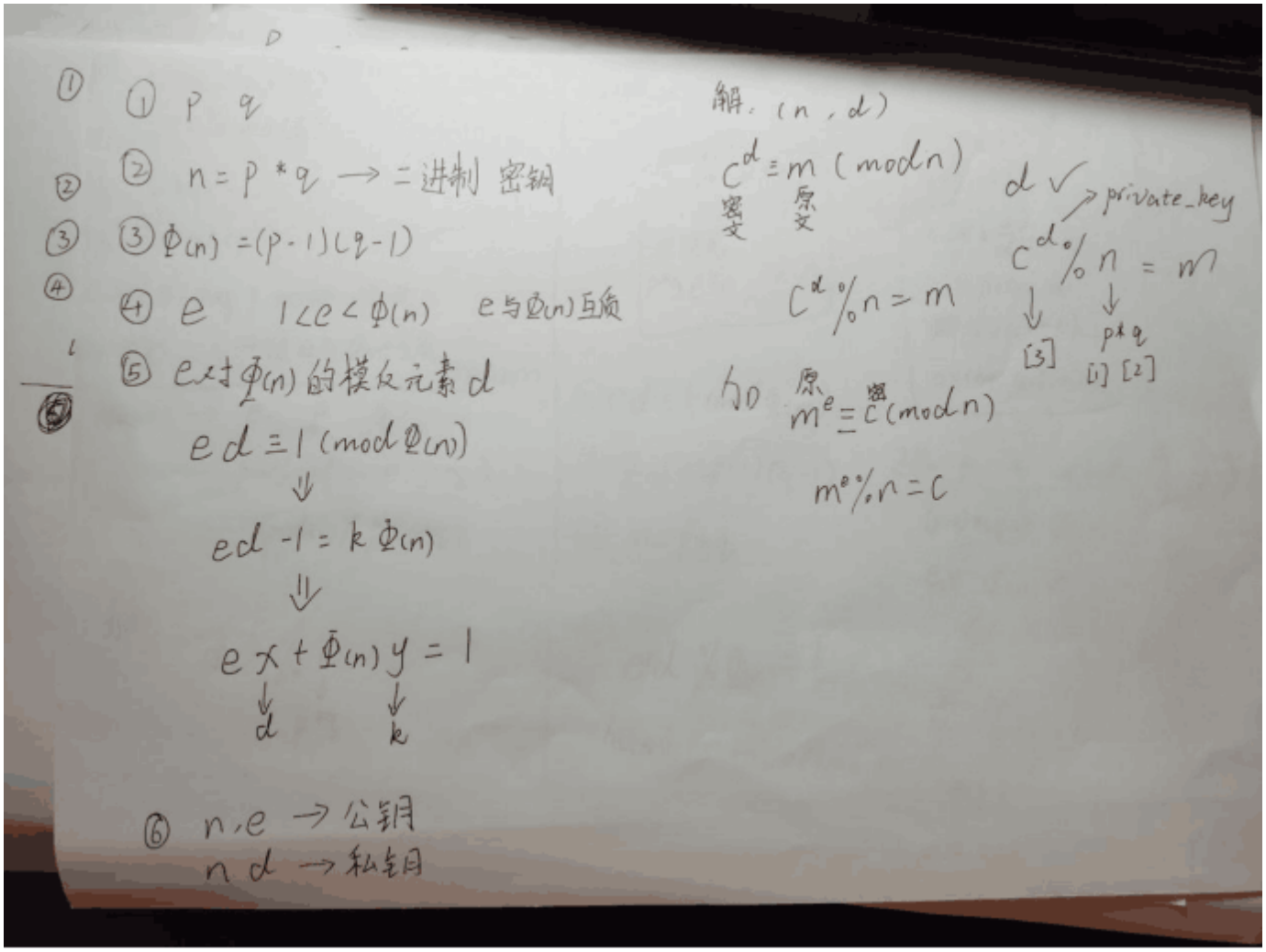
```
['1101000', '1100111', '1100001', '110110
[104, 103, 97, 109, 101, 123, 68, 97, 110
hgame{Danc1ng_L1ne_15_fun,_15n't_1t?}
```

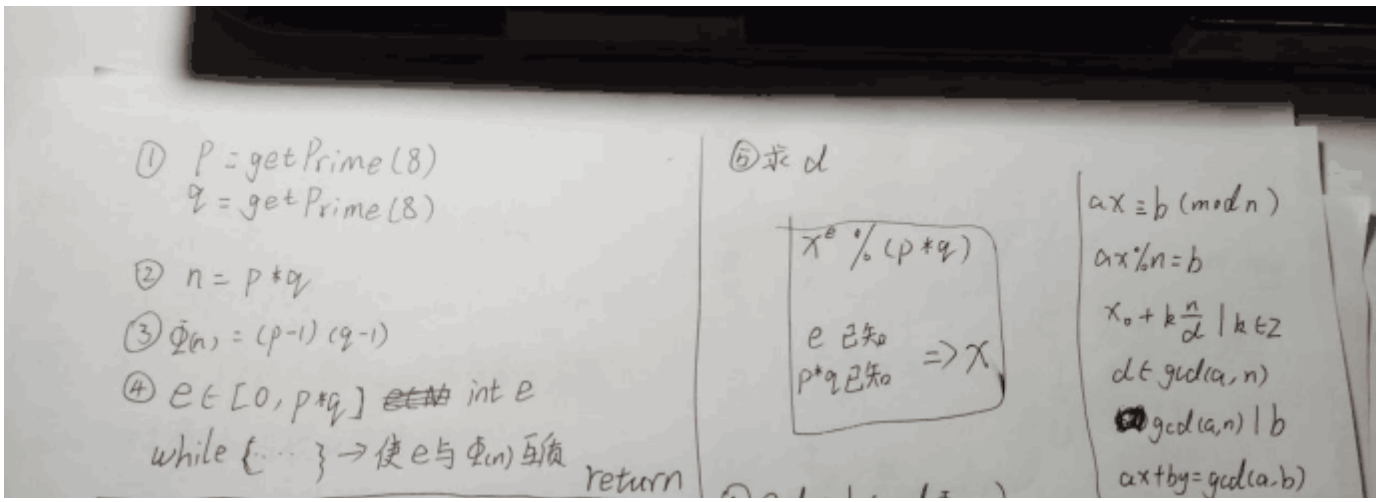
Easy RSA

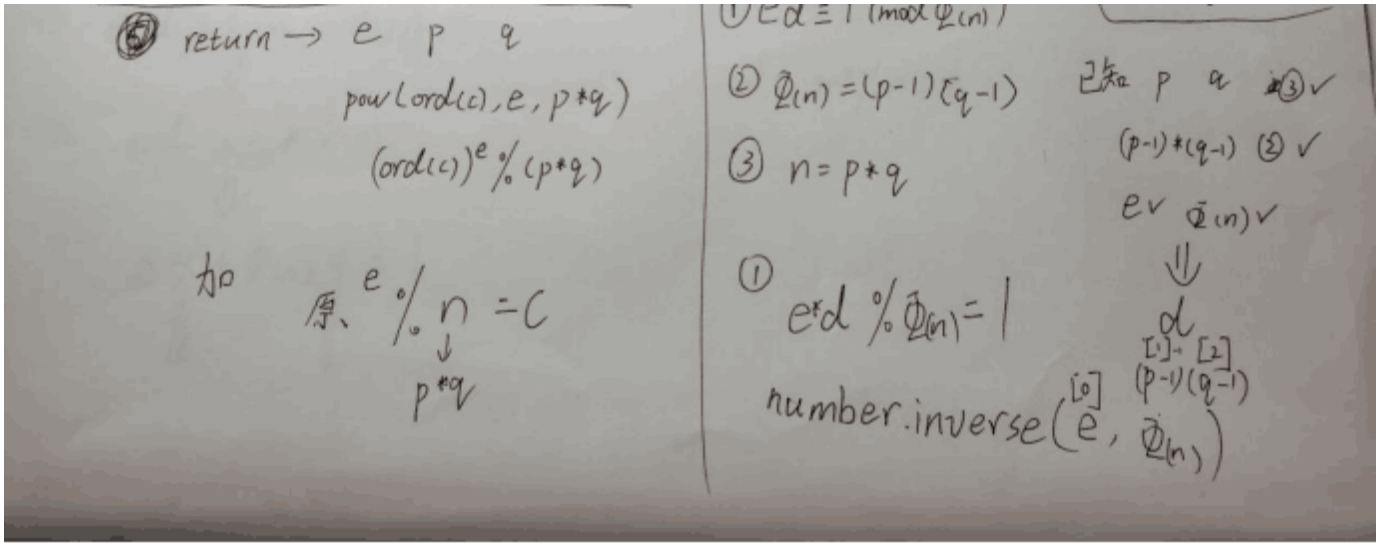
- **思路**

根据题目是一个rsa加密，打开加密的python文件，分析内容，与rsa标准加密做对比；

> 手稿








```
def encrypt(c):
    #两个质数
    p = getPrime(8)
    q = getPrime(8)
    #n=p*q，欧拉函数（n），取一个数e与(n)互质
    e = randint(0, p * q)
    while gcd(e, (p - 1) * (q - 1)) != 1:
        e = int(next_prime(e))
    #返回 e，p，q，flag的ASCII码的e次方mod（p*q）
    return e, p, q, pow(ord(c), e, p * q)
```

rsa加密、解密的原理及方法建议参考：(进入后无法浏览可刷新界面)

> 阮一峰_RSA算法原理（一）
>
> 阮一峰_RSA算法原理（二）

- 求出ASCII码：密文（pow(ord(c), e, p * q)设为c,原文为m
  `m=pow(c, private_key) % (p * q)` 缺少私钥
  设私钥为d
- 解密条件：求出d（私钥）
  1. e*d mod (oula(n))=1（已知e，需要oula(n)）
  2. oula(n)=(p-1)*(q-1)（已知p，q）
  3. n=p*q（已知p，q）

乘法逆元可以采用python里的第三方库

```
from Crypto.Util.number import inverse
```

- **代码**

```
"""
user:鸢柒
purpose:rsa解密
time:2022.1.20
"""
from Crypto.Util.number import inverse
if __name__ == '__main__':
    private_key = []
    m = []
    zu = [(12433, 149, 197, 104), (8147, 131, 167, 6633), (10687, 211, 197, 35594), (19681, 131
          (33577, 251, 211, 38798), (30241, 157, 251, 35973), (293, 211, 157, 31548), (26459, 1
          (27479, 149, 223, 32728), (9029, 223, 137, 20696), (4649, 149, 151, 13418), (11783, 2
          (13537, 179, 137, 11702), (3835, 167, 139, 20051), (30983, 149, 227, 23928), (17581,
          (35381, 223, 179, 37774), (2357, 151, 223, 1849), (22649, 211, 229, 7348), (1151, 179
          (8431, 251, 163, 30226), (38501, 193, 211, 30559), (14549, 211, 151, 21143), (24781,
          (8051, 179, 131, 7994), (863, 181, 131, 11493), (1117, 239, 157, 12579), (7561, 149,
          (19813, 239, 229, 53463), (4943, 131, 157, 14606), (29077, 191, 181, 33446), (18583,
          (30643, 173, 191, 27293), (11617, 223, 251, 13448), (19051, 191, 151, 21676), (18367,
          (18861, 149, 191, 5139), (9581, 211, 193, 25595)]
    #计算乘法逆元得到私钥d
    for i in zu:
        private_key.append(inverse(i[0], (i[1] - 1) * (i[2] - 1)))
    #根据d以及解密公式得到原文的ASCII码，转化输出
    for j in range(len(zu)):
        m.append(pow(zu[j][3], private_key[j]) % (zu[j][1] * zu[j][2]))
        print(chr(m[j]), end='')
```

输出结果为：hgame{L00ks_l1ke_y0u've_mastered_RS4!}

English Novel

- **思路**
  先来看看文件夹里给了些啥：
  - encrypt
  - original
  - encrypt python文件
  - flag.enc(一开始没看见有这东西就去解密钥了)

简单看一下encrypt和original里面的东西，都分成了410份，根据题目一部分是加密的，一部分是康师傅找到的原文，经过python分析发现两部分文本的总字数是相等的且等于原文长度，和原文中“一部分”不符，经过询问出题人得知一部分指的是有很多本小说，这是其中一本。

有打乱的原文和密文，自然是要找到原文与之对应的密文的两者的编号；

再来看看加密的python文件：

```
def encrypt(data, key):
    assert len(data) <= len(key)
    result = ""
    for i in range(len(data)):
        if data[i].isupper():
            result += chr((ord(data[i]) - ord('A') + key[i]) % 26 + ord('A'))
        elif data[i].islower():
            result += chr((ord(data[i]) - ord('a') + key[i]) % 26 + ord('a'))
        else:
            result += data[i]
    return result
```

根据加密文件可以看出，这是对英文字母的一个ASCII码根据密钥key进行的位移，那么我们还需要求解的就是key，要求解key必须知道密文对应的原文是哪一部分；

根据加密文件，密文与原文的标点符号是对应的，位置没有改变，根据非字母位置或者根据字母位置可以得出密文与原文的匹配数对；

```
def fuhao(data):
    fu=[]
    for i in range(len(data)):
        if data[i].islower():
            fu.append(i)
    return fu
```

```
    for i in range(len(enc_fu)):
        for j in range(i,len(ori_fu)):
            if enc_fu[i]==ori_fu[j]:
                yua.append([i,j])
```

得到匹配数对后，根据数对可以用原文的ASCII码与密文的ASCII码做差得到key，根据key解密验证

```
def key_to(enc,ori):
    key=[]
    for i in range(len(enc)):
        key.append(ord(enc[i])-ord(ori[i]))
    return key
```

验证结果见代码区块，根据验证可知我们得到了正确的密钥，然后嘞，这密钥干嘛?

没错，文件里还有个被遗忘了的flag.enc文件，用十六进制编辑器打开看看是什么东西：（010 editor)

klsyf{W0_j0v_ca0z_'Ks0ao-bln1qstxp_juqfqy'?}

居然是flag的密文，有了密钥，密文，该干啥不用说了吧?

```
flag='klsyf{W0_j0v_ca0z_\'Ks0ao-bln1qstxp_juqfqy\'?}'
    flag_1=[]
    for i in range(len(key)):
        print(decrypt(flag,key[i]))
        flag_1.append(decrypt(flag,key[i]))
```

工作结束，会得到200多个flag，每个flag都有几个字符是错的，我们可以根据是否可翻译来筛选出错误较少的几个进行拼凑得出正确的flag。（解的话还有一个密钥是符号对应位置）

- **代码**

```
"""
user:鸢柒
purpose:novel
time:2022.01.23__16:19
"""
#解密
def decrypt(data, key):
    assert len(data) <= len(key)
    result = ""
    for i in range(len(data)):
        if data[i].isupper():
            result += chr((ord(data[i]) - ord('A') - key[i]) % 26 + ord('A'))
        elif data[i].islower():
            result += chr((ord(data[i]) - ord('a') - key[i]) % 26 + ord('a'))
        else:
            result += data[i]
    return result
#查找储存对应字母位置
def fuhao(data):
    fu=[]
    for i in range(len(data)):
        if data[i].islower():
            fu.append(i)
    return fu
#得出密钥
def key_to(enc,ori):
    key=[]
    for i in range(len(enc)):
        key.append(ord(enc[i])-ord(ori[i]))
    return key
if __name__ == '__main__':
    enc = []
    ori = []
    yua = []
#存储密文和原文
    for i in range(0, 410):
        with open('F:\Face_to_the_prison\Crypto\HGAME\English Novel\encrypt\part' + str(i) + '.
            enc.append(encr.read())
        with open('F:\Face_to_the_prison\Crypto\HGAME\English Novel\original\part' + str(i) + '
            ori.append(orig.read())
    enc_fu=[]
    ori_fu=[]
#匹配密文和原文
    for i in enc:
        enc_fu.append(fuhao(i))
    for j in ori:
        ori_fu.append(fuhao(j))
    for i in range(len(enc_fu)):
        for j in range(i,len(ori_fu)):
```

```
            if enc_fu[i]==ori_fu[j]:
                yua.append([i,j])
    key=[]
#计算key，并测试key的正确性
    for i in yua:
        key.append(key_to(enc[i[0]],ori[i[1]]))
    print(key[2])
    print(decrypt(enc[1],key[1]))
#flag解密
    flag='klsyf{W0_j0v_ca0z_\'Ks0ao-bln1qstxp_juqfqy\'?}'
    flag_1=[]
    for i in range(len(key)):
        #print(decrypt(flag,key[i]))
        flag_1.append(decrypt(flag,key[i]))
#根据条件筛选
        if decrypt(flag,key[i])[:26]=='hgame{D0_y0u_kn0w_\'Kn0wn-p':
            print(decrypt(flag,key[i]),'dui',yua[i])
    print(decrypt(flag,key[2])[:26])
```

运行结果：

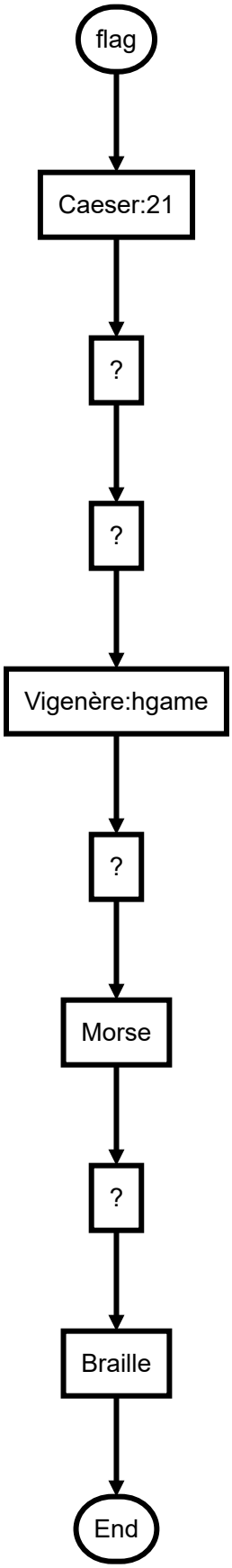
```
[3, 0, 18, 12, 1, -2, 0, -16, 9, -15, 1, 0, 8, -8, -13, -7, 3,
 a long life, I have had much time for thought as I lay alone
"Now, comrades, what is the nature of this life of ours? Let u
hgame{D0_y0u_kn0w_'Kn0wn-pla1ntext_jutack'?} dui [200, 346]
hgame{D0_y0u_kn0w_'Kn0wn-pla1nsext_attack'?} dui [203, 269]
hgame{D0_y0u_kn0w_'Kn0wn-pln1ntext_atqack'?} dui [236, 275]
hlame{W0_y0v_kn0w_'Kn0an-p
```

拼凑一下即可

符号位密钥不再演示

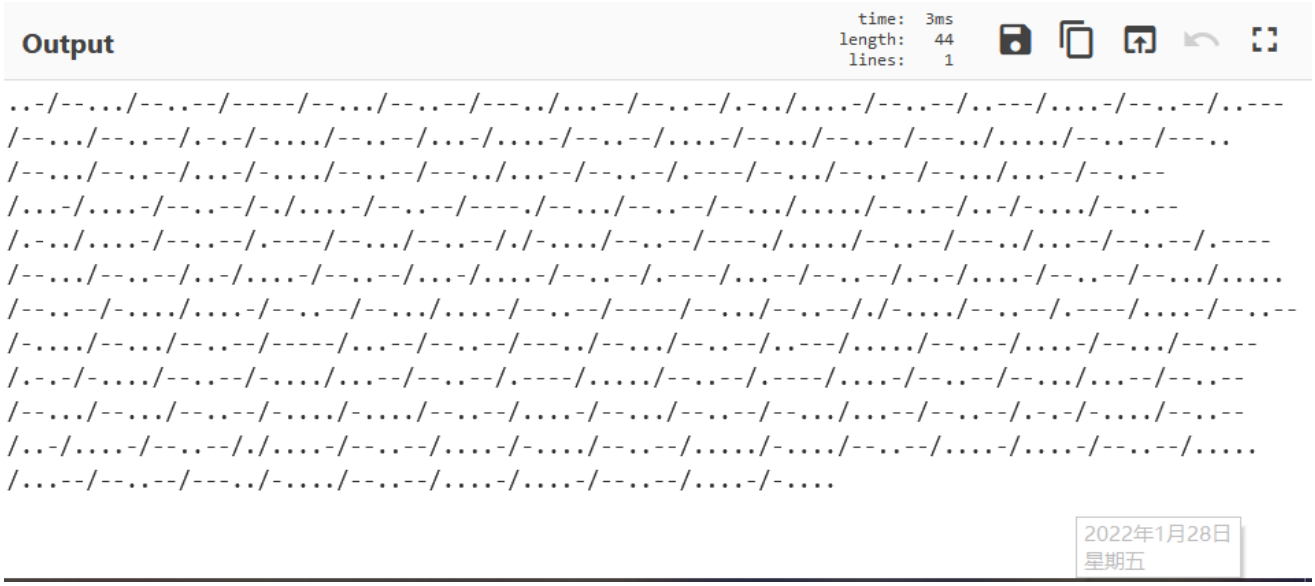
Danceing Line

- **思路**
  根据公告里的提示：



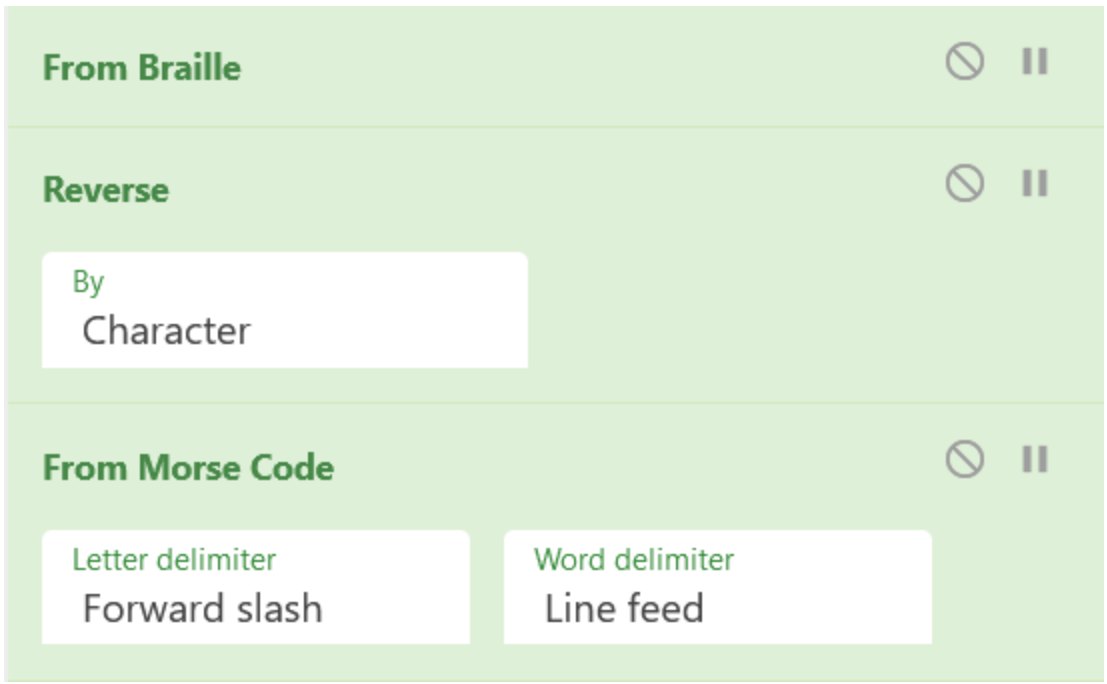
并给了一个特别特别好用的工具：CyberChef

- **步骤**



2022年1月28日
星期五

"."与"-" 肯定是摩斯，但是流程图中还有一步在中间，不改变摩斯的可解密必然是置换，没有给密钥，先尝试下逆序，python[::-1]或者工具里的reverse



46,66,42,75,66,45,46,6E,6D,4C,73,36,44,33,73,69,59,74,4C,36,58,32,70,34,69,4E,30,63,64,53,6C,79,6B,6D,39,72,51,4E,39,6F,4D,53,31,6A,6B,73,39,72,4B,32,52,36,6B,4C,38,68,6F,72,30,3D

可以比较明显的，看出这和平时看到16进制编辑器里的hex很像

hex 解码后得出

FfBufEFnmLs6D3siYtL6X2p4iN0cdSlykm9rQN9oMS1jks9rK2R6kL8hor0=

虽然还有一步维吉尼亚加密但根据等号等特征已经可以看出下下一步是Base64了

维吉尼亚解密，base64解码后得到：

c0bmvghyz_{0Raz_gxxm0thkzo_0ob0m_vokcczt_!r}

这倒数第二步肯定是置换，至于是怎么换还不清楚，但凯撒加密的密钥是对所有明文统一的，置换和凯撒顺序调换不会有改变，所以先凯撒大帝来一波：（单字母密钥的维吉尼亚加密=凯撒加密）

h0gralmde_{0Wfe_lccr0ympet_0tg0r_atphhey_!w}

到这里结果就很明显了，奇数位字符和偶数位字符拼凑得到flag

```
>>> a='h0gralmde_{0Wfe_lccr0ympet_0tg0r_atphhey_!w}'
>>> b=0
>>> c=''
>>> for i in a:
        if b%2==0:
                c=c+i


>>> c
'h0gralmde_{0Wfe_lccr0ympet_0tg0r_atphhey_!w}'
>>> for i in a:
        if b%2==0:
                c=c+i
        b=b+1


>>> c
'h0gralmde_{0Wfe_lccr0ympet_0tg0r_atphhey_!w}hgame{Welc0me_t0_the_w'
>>> c='hgame{Welc0me_t0_the_w'
>>> d=''
>>> for i in a:
        if b%2!=0:
                d=d+i
        b=b+1


>>> d
'0rld_0f_crypt0graphy!}'
>>>
```

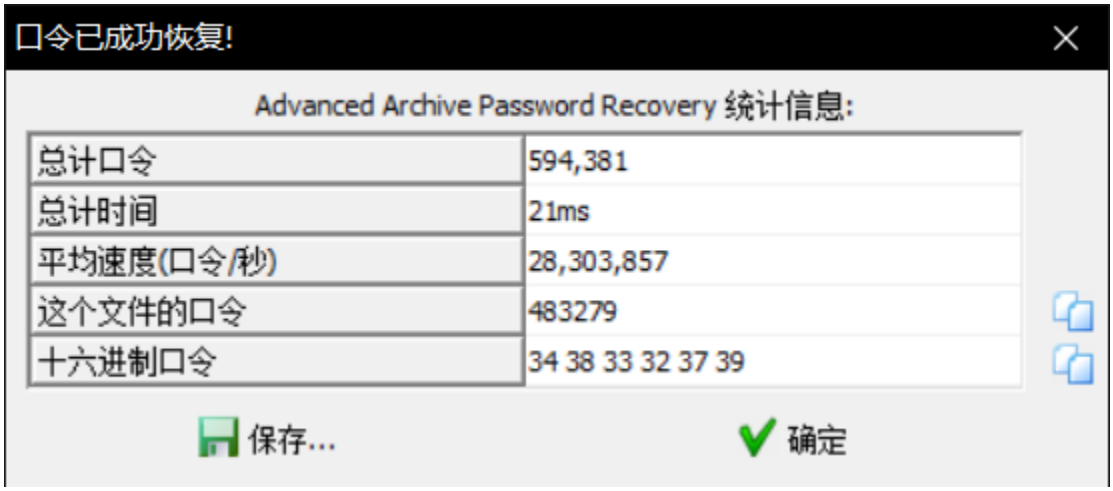
hgame{Welc0ome_t0_the_w0rld_0f_crypt0graphy!}

Misc

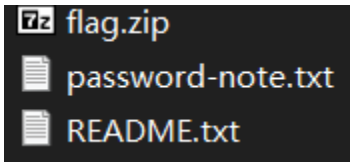
这个压缩包有点麻烦

- **过程**:

  1. 二话不说，先暴力来一波：

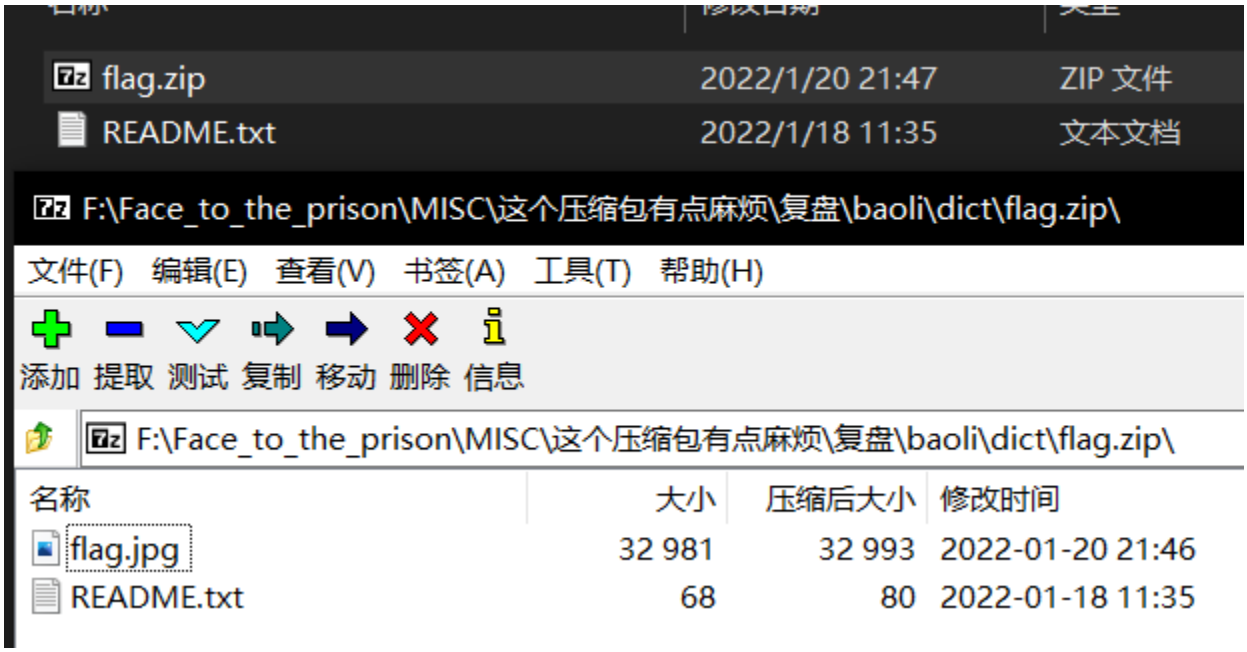


得到



观察password.txt，应该是要字典破解，白给的字典不加白不加

得到



有个文件名一样的东西可以尝试压缩一下看看crc效验码

从压缩文件里看压缩方式是ZipCrypto Store(这种压缩方式7z是没有的，得去找找2345快压和360压缩，全家桶警告，官网下，注意点鼠标)

CRC	算法
966AC0E8	Store

[redacted]

CRC	算法
4BDC1793	ZipCrypto Store
966AC0E8	ZipCrypto Store

开始明文解压，解压出密钥就可以停止了，这里因为有保存就不复盘了

用密钥会重新解出一个压缩包保存,解压得到一个.jpg文件，写着where is flag

查看文件基础信息没有什么奇怪的东西

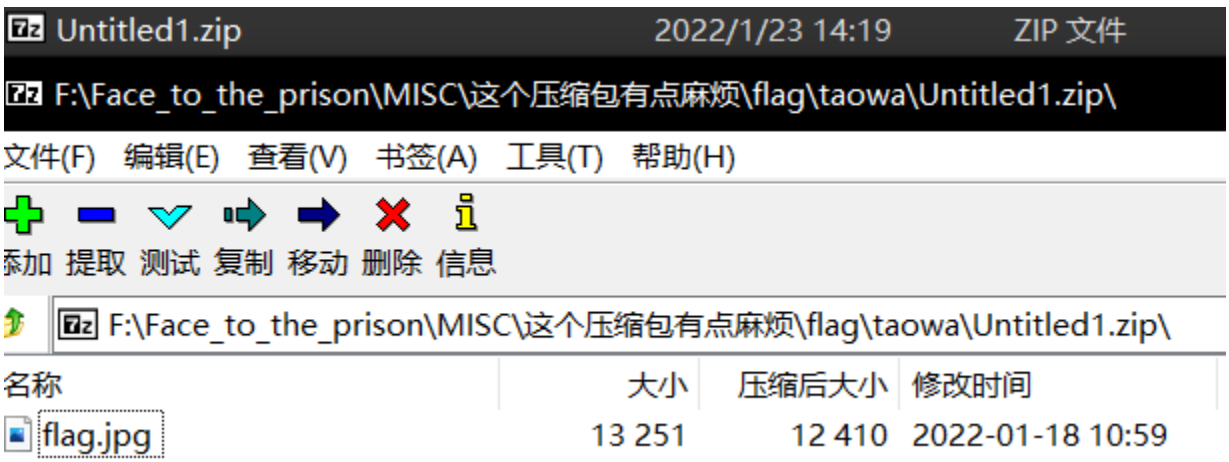
联想到隐写二进制图片与压缩包连接因为结束特征码不会显示后面的内容，用压缩文件压缩起来点击jpg文件发现里面还有个加密的jpg

名称	大小	压缩后大小
flag.jpg	13 251	12 410

CRC	算法
F320A65C	ZipCrypto Deflate

用16进制编辑器打开找到结束位置查看是否存在压缩包的文件头

```
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A ( .Š( .Š( .Š( .Š
```



```
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 02 8A 28 A0 02 8A 28 A0 02 8A 28 A0 02 8A  ( .Š( .Š( .Š( .Š
28 A0 0F FF D9 50 4B 03 04 14 00 09 00 08 00 7D  ( .ÿÙPK........}
57 32 54 5C A6 20 F3 7A 30 00 00 C3 33 00 00 08  W2T\¦ óz0..Ã3...
00 00 00 66 6C 61 67 2E 6A 70 67 CC 5A 67 54 53  ...flag.jpgÌZgTS
51 12 8E 02 82 54 41 41 04 24 2A BD 89 4A 2F 92  Q.Ž.‚TAA.$*½‰J/’
45 04 44 C4 A8 54 41 88 8A 94 80 10 69 D2 42 A2  E.DÄ¨TAˆŠ”€.iÒB¢
F4 CE 02 02 AB 08 59 A9 22 22 D2 7B 22 BD 09 48  ôÎ..«.Y©""Ò{"½.H
17 10 12 82 80 08 91 84 12 1E 24 84 7D EE EE D9  ...‚€.‘„..$„}îîÙ
3D 67 7F ED EE D9 1F 7B 93 9C 97 73 5E 9B B9 33  =g.íîÙ.{“œ—s^›¹3
73 E7 9B EF CE C1 D7 03 12 E4 D8 75 13 33 13 C8  sç›ïÎÁ×..äØu.3.È
A1 43 87 20 9F C0 0F E4 60 06 62 04 39 7C E8 D0  ¡C‡ ŸÀ.ä`.b.9|èÐ
EF EF DF 07 1B 07 DB EF 71 84 9D 9D 8D 83 F3 08  ïïß...Ûïq„...ƒó.
27 E7 EF 1F D7 51 9E A3 5C 5C DC 5C 9C 9C DC 7C  'çï.×Qž£\\Ü\œœÜ|
DC DC 3C BC E0 E0 3C CA 2F C0 C7 CB FF FB FF EF  ÜÜ<¼àà<Ê/ÀÇËÿûÿï
87 FC BE FD F7 5D E0 97 97 8B 93 EB F7 15 FF D9  ‡ü¾ý÷]à——‹“ë÷.ÿÙ
38 F8 04 11 E4 82 E8 1D 56 65 3B 74 0E 72 58 F0  8ø..ä‚è.Ve;t.rXð
10 9B E0 A1 83 0E 08 14 02 39 C4 01 4A FB 5B E0  .›à¡ƒ....9Ä.Jû[à
BF 8F 43 87 D9 D8 39 8E 80 22 71 F3 80 17 D4 1C  ¿.C‡ÙØ9Ž€"qó€.Ô.
03 C5 67 63 3B 0C 0A CB C1 CE 0E 9E 0D 03 CF 43  .Ågc;..ËÁÎ.ž..ÏC
D8 05 39 84 CE 5E 34 3C 72 FC F6 03 CE 73 3E 27  Ø.9„Î^4<rüö.Îs>'
2E 3D 4F 7D C3 25 75 A5 A2 55 F8 CE 30 55 5A ED  .=O}Ã%u¥¢UøÎ0UZí
A1 6F F8 51 6E 91 93 A2 A7 C4 64 64 E5 E4 15 14  ¡oøQn‘“¢§Äddåä..
D5 35 34 B5 B4 75 74 8D AE 1A 9B 98 5E 33 BB 6E  Õ54µ´ut.®.›˜^3»n
69 65 6D 63 6B 77 D7 DE F9 91 8B AB 9B 3B D2 C3  iemckw×Þù‘‹«›;ÒÃ
CF 3F E0 69 60 50 70 48 44 64 54 74 4C 6C 5C 7C  Ï?ài`PpHDdTtLl\|
5A FA 8B 8C CC AC 3F BD 7C 95 97 5F 50 58 54 FC  Zú‹ŒÌ¬?½|•—_PXTü
```

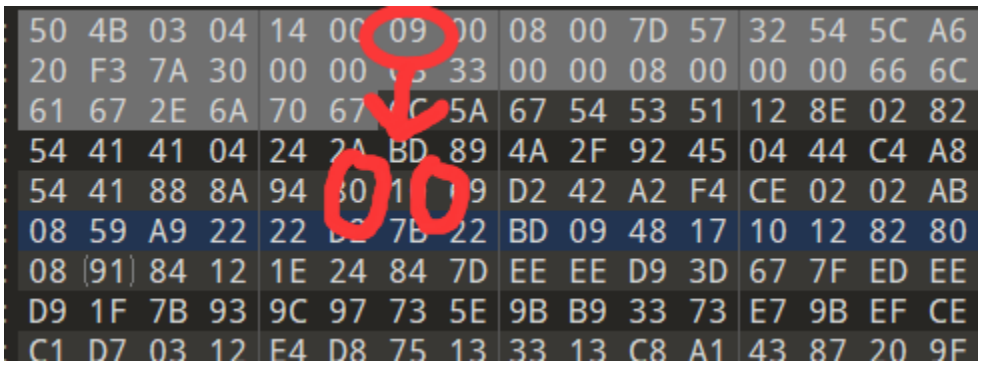
果然有一个zip文件的文件头，复制文件头及以下部分，在新页面粘贴导出为zip得到隐藏的zip

Untitled1.zip 2022/1/23 14:19 ZIP 文件

F:\Face_to_the_prison\MISC\这个压缩包有点麻烦\flag\taowa\Untitled1.zip\

F:\Face_to_the_prison\MISC\这个压缩包有点麻烦\flag\taowa\Untitled1.zip\

名称	大小	压缩后大小	修改时间
flag.jpg	13 251	12 410	2022-01-18 10:59

> 这里就比较难受了
>
> 这里我尝试hgame???...来掩码破解时间过长，暴力时间过长或找不到，用前面的密钥解析失败，重新压缩一个jpg的deflate的zip部分明文破解失败，在去网上查找的时候找到了一种伪加密的zip进行尝试：
>
> 
>
> 把对应的加密的标志位改为00后解压成功得到jpg
>
> hgame{W0w!_y0U_Kn0w_z1p_3ncrYpt!}

总结

虽然我很菜，学东西慢，还喜欢打游戏，不过在hgame的这段时间里我不断的在网上学习，脑子里的知识切切实实的在一天比一天多，且影响深刻，吃饭、上厕所都在想题这种情况居然也发生在我身上了，专心做题的时候感觉不到时间，一不小心就干到夜里3，4点，第二天虚的.....在不断学习以及思考和出题人交流的过程里确实让我学到了很多经验，每次把题解出来的时候，除了爽就是身心舒畅，简直不要太嗨，感谢hgame，感谢vidar，感谢出题人。