ctfshow CRYPTO

原创

sec0nd_ 于 2022-03-16 00:11:00 发布 292 收藏 2

分类专栏： crypto 文章标签： 安全 ctf crypto

文章目录

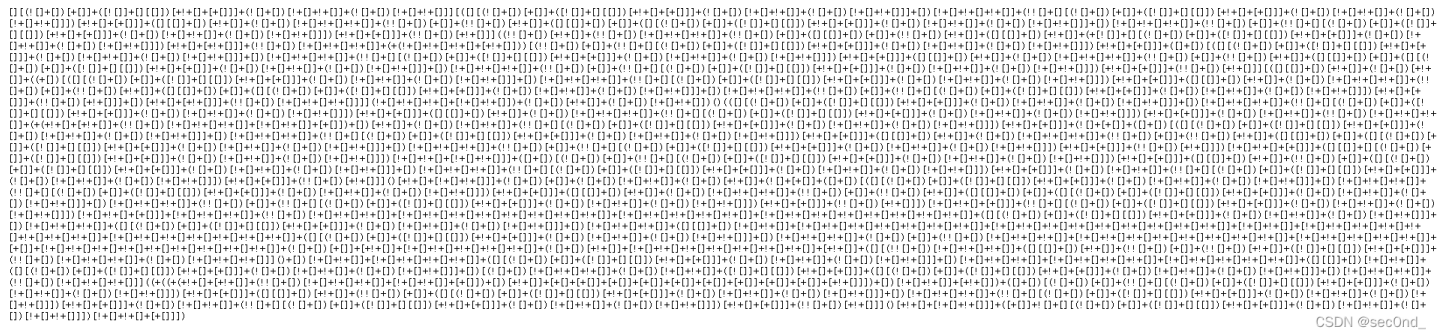
密码学签到

```
}wohs.ftc{galf
```

倒序输出就是flag了

crypto2

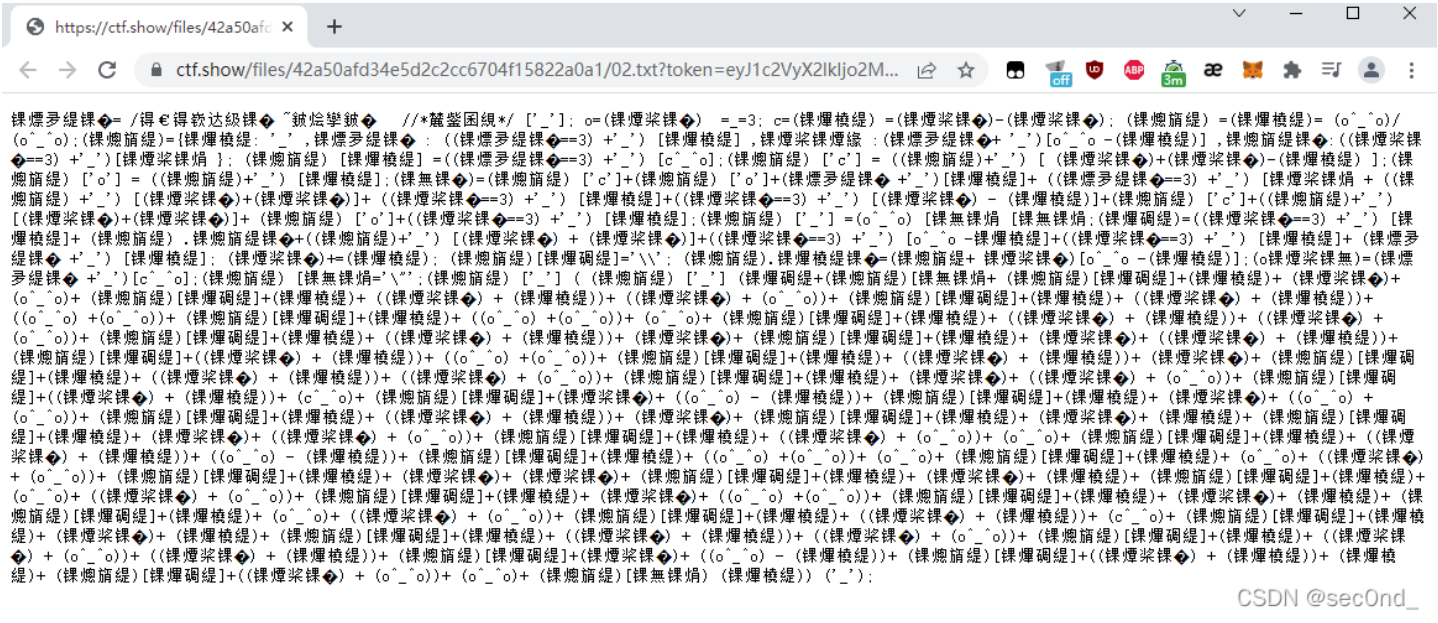
题目是一长串的符号



这个是jsfuck 、可以看这个地址

直接在控制台粘贴回车

crypto3



是乱码哎，但是感觉不太对，里面有一些颜文字，应该是颜文字加密，需要抓个包看看正确的相应报文

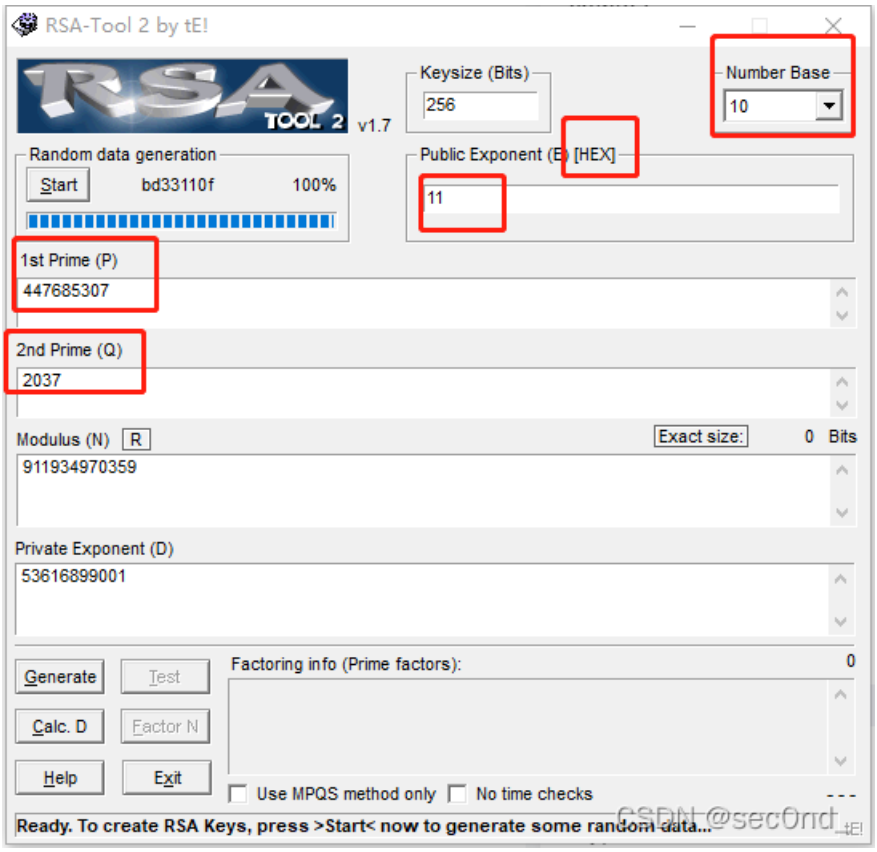
同样是控制台 粘贴 回车就出来了

crypto4

```
p=447685307 q=2037 e=17
提交flag{d}即可
```

这里用到的工具是RSA-Tool2

注意e是十六进制的

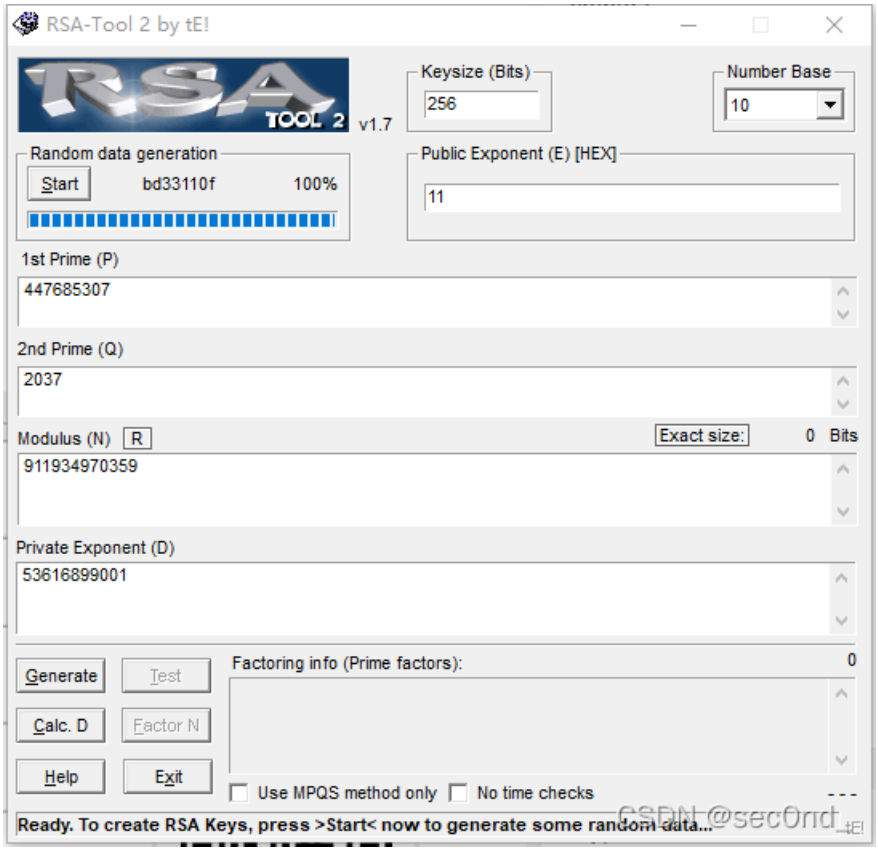


输入好之后，点击求D即可

crypto5

```
p=447685307 q=2037 e=17 c=704796792
提交flag{m}
```

先用工具计算其他值

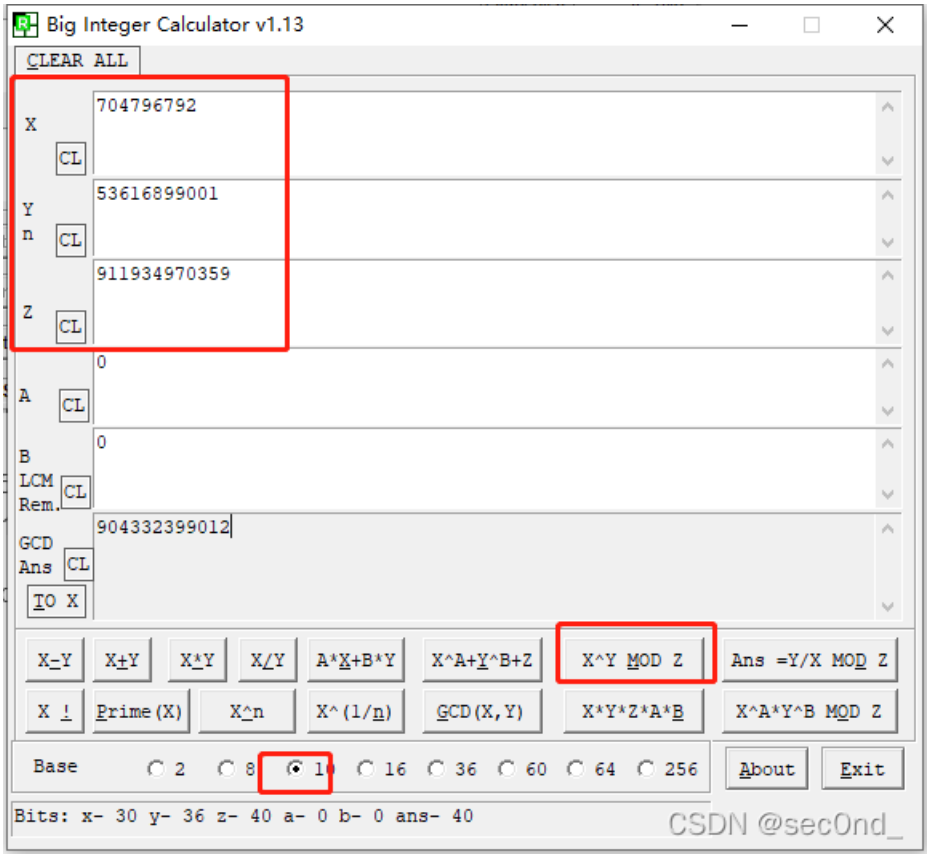


n=911934970359

d=53616899001

解密步骤：m= c^d mod n

使用工具big integer calculate



crypto6

密钥为 加密方式 名称，区分大小写

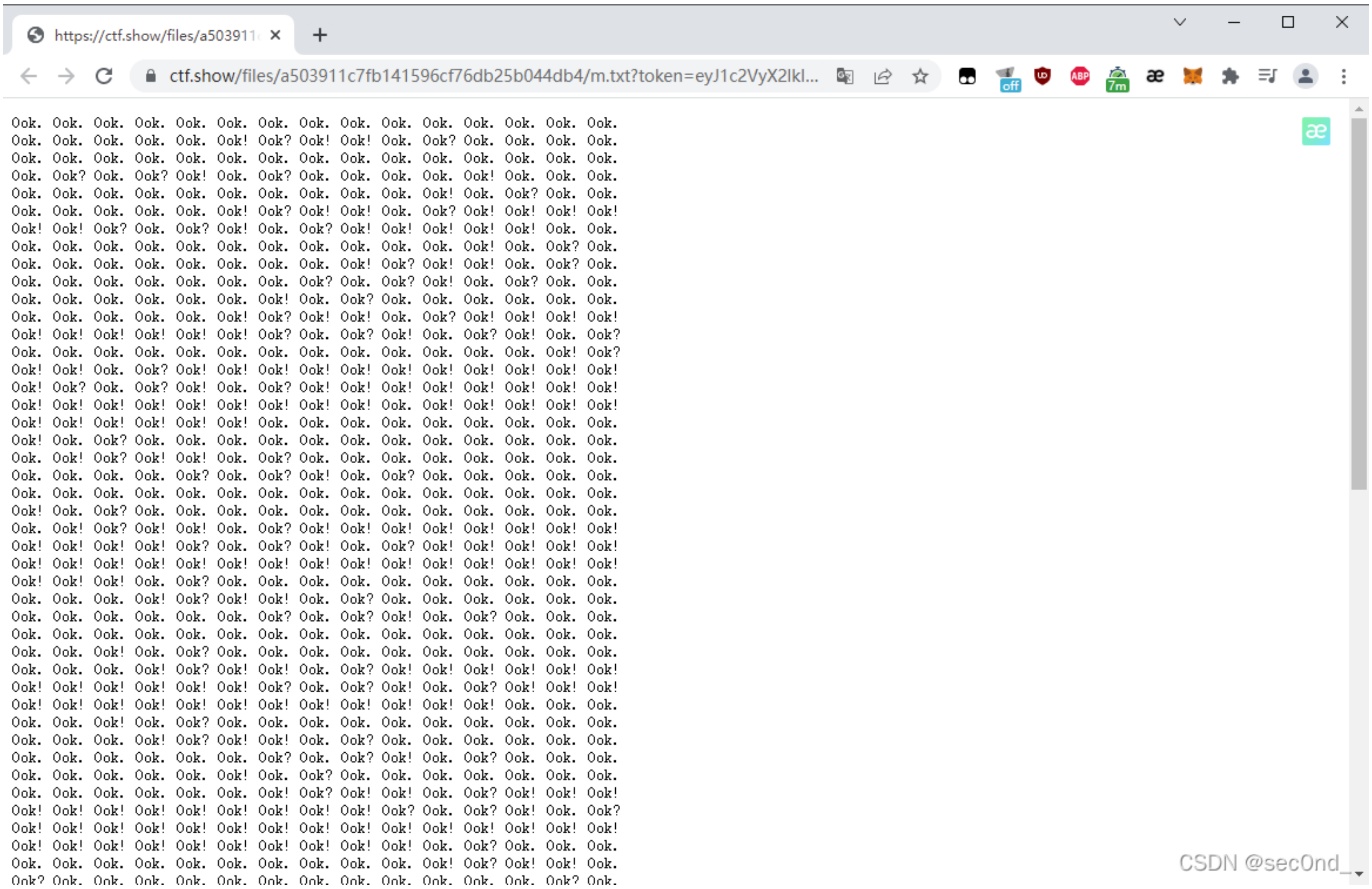
```
U2FsdGVkX19mGsGlfI3nciNVpWZZRqZO2PYjJ1ZQuRqoiknyHSWeQv8ol0uRZP94
MqeD2xz+
```

这个是Rabbit加密，密钥应该就是Rabbit

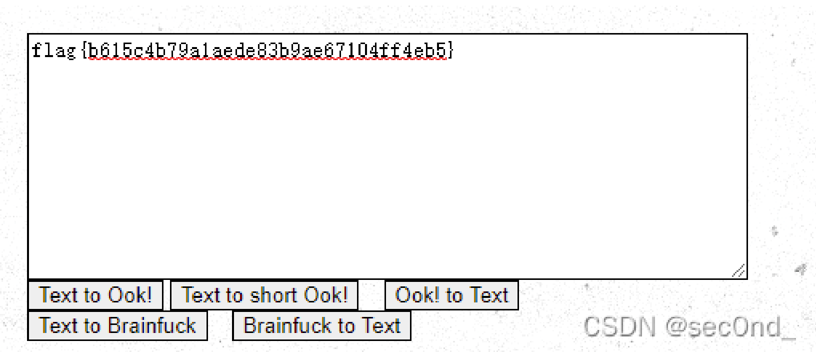
在线解密工具地址

U2FsdGVkX1开头的可能是rabbit，AES，DES

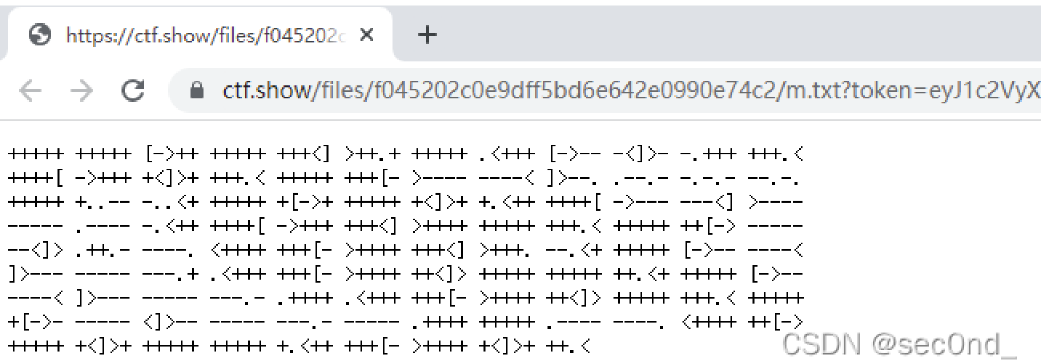
crypto7



这个就很常见了，Ook！在线解密

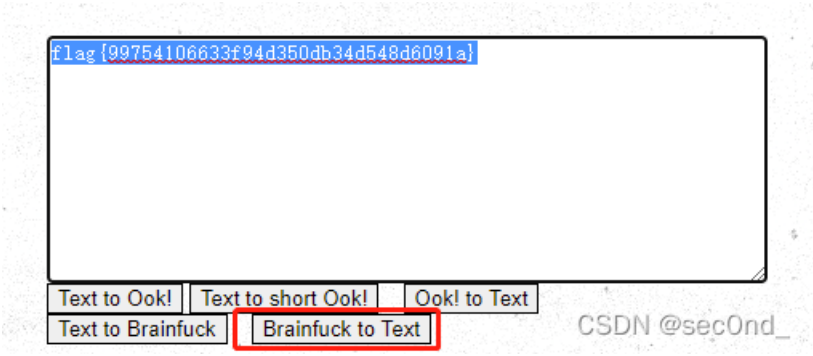


crypto8

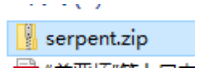


这个是跟上面Ook！是在一起的

brainfuck编码 在线解密

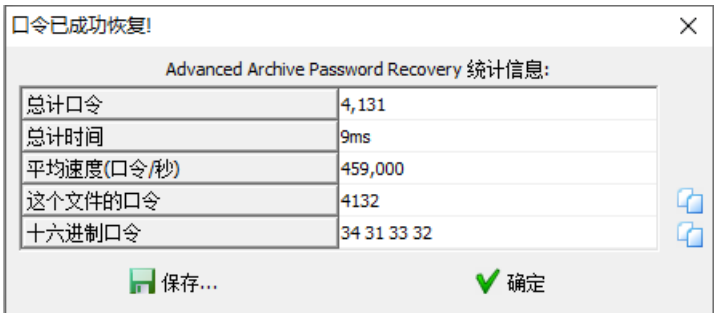


crypto9

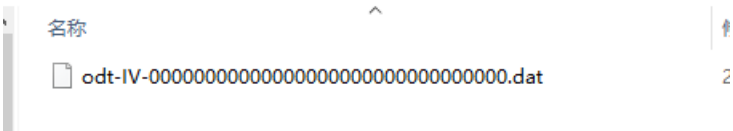


是一个加密后的压缩包，但是这个名字有点眼熟

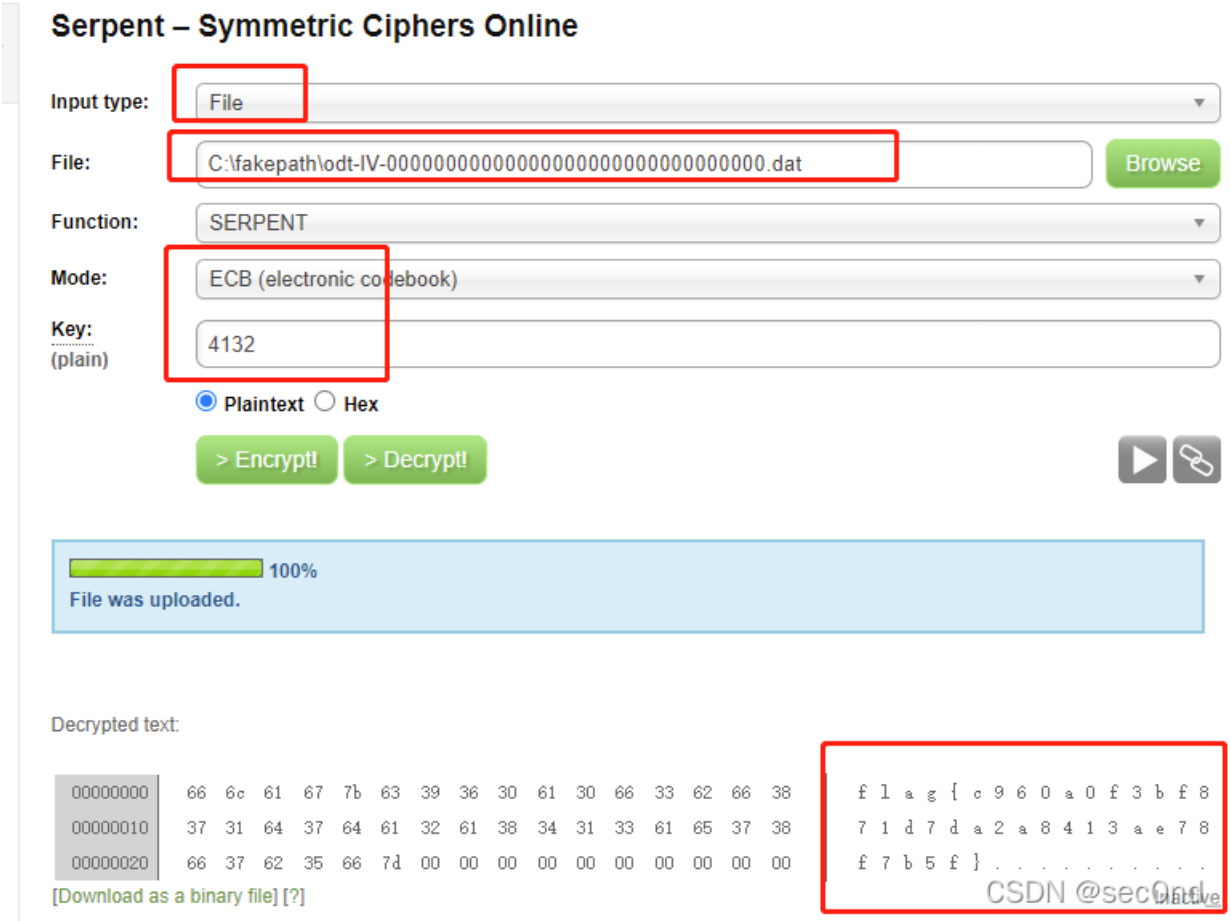
算了先暴力破解出这个压缩包的密码



得到密码是4132

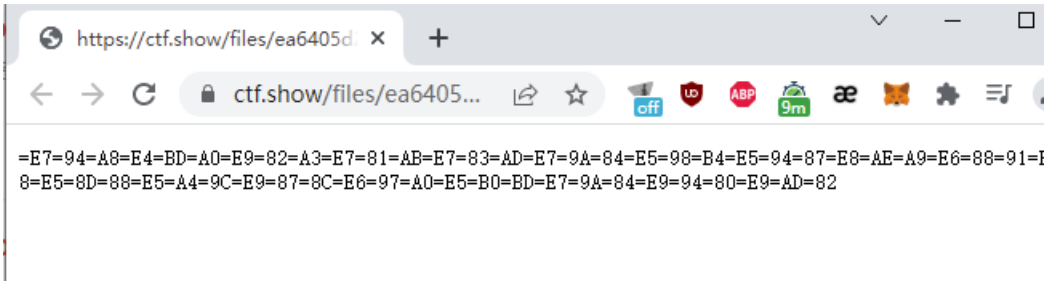


得到一个.dat文件，这时候想到了文件的名字，是一个对称密码，在线解密



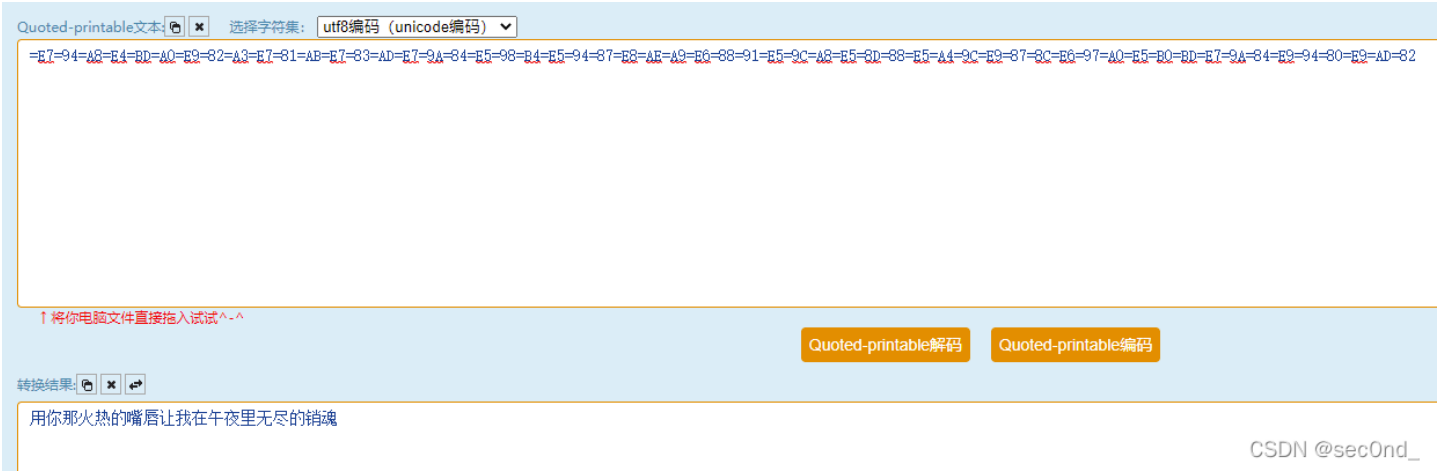
得到flag

crypto10

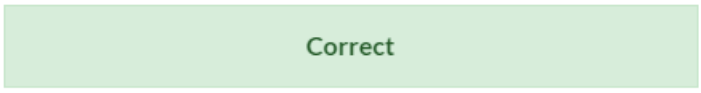


Quoted-printable 可译为“可打印字符引用编码”、“使用可打印字符的编码”，我们收邮件，查看信件原始信息，经常会看到这种类型的编码！

在线解密地址：http://web.chacuo.net/charsetquotedprintable



嗯？？？？？这flag？？？？？？



绝了

crypto11

密文 `a8db1d82db78ed452ba0882fb9554fc`

32位，盲猜md5 somd5在线解密

crypto0

```
gmbh{ifmmp_dug}
```

有括号哎，应该是凯撒密码吧，试一试

flag{hello_ctf}

crypto12

```
uozt{Zgyzhv_xlwv_uiln_xguhsld}
```

感觉是栅栏密码或者移位，试了多次发现出不来，翻了翻古代密码的其他，有个埃特巴什码

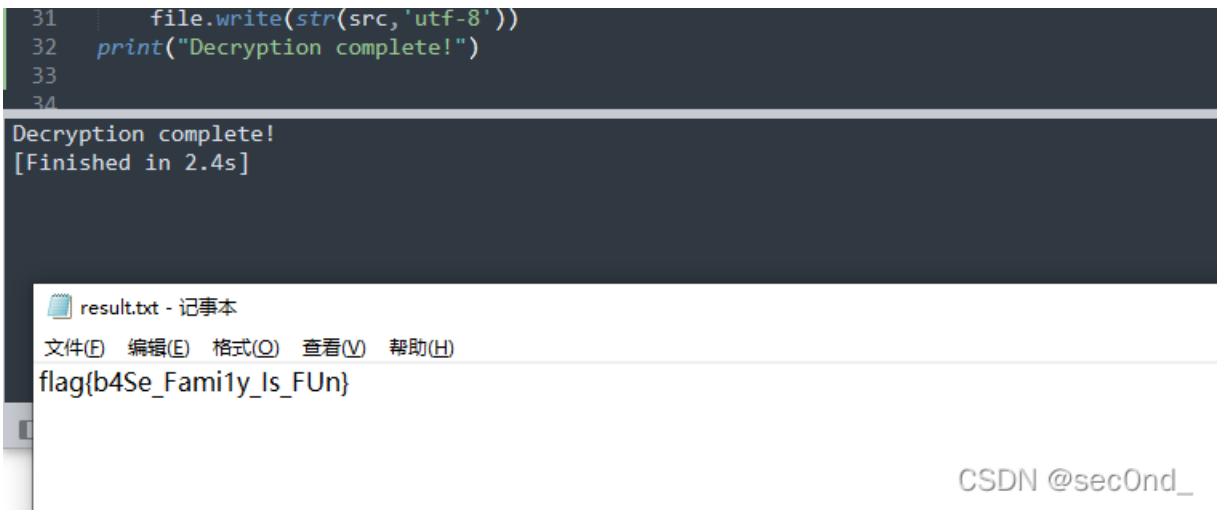
crypto13

看这个名字，感觉应该是base来回嵌套了，这文件这么大
啊来回解了四五次还很长，看来要用脚本了

```python
import base64

s=''
with open('base.txt', 'r', encoding='UTF-8') as f:
    s=''.join(f.readlines()).encode('utf-8')
src=s
while True:
    try:
        src=s
        s=base64.b16decode(s)
        str(s,'utf-8')
        continue
    except:
        pass
    try:
        src=s
        s=base64.b32decode(s)
        str(s,'utf-8')
        continue
    except:
        pass
    try:
        src=s
        s=base64.b64decode(s)
        str(s,'utf-8')
        continue
    except:
        pass
    break
with open('result.txt','w', encoding='utf-8') as file:
    file.write(str(src,'utf-8'))
print("Decryption complete!")
```

把脚本和base.txt放在同一目录下，运行后，result.txt中即为flag



crypto14

```
00110011 00110011 00100000 00110100 00110101 00100000 00110101 00110000 00100000 00110010 01100110 00100000 0011
0011 00110011 00100000 00110101 00110110 00100000 00110100 01100101 00100000 00110100 00110110 00100000 00110100
 00110110 00100000 00110110 01100100 00100000 00110100 01100101 00100000 00110100 00110101 00100000 00110100 001
10001 00100000 00110110 01100101 00100000 00110110 01100011 00100000 00110100 00111000 00100000 00110100 0011010
0 00100000 00110011 00110101 00100000 00110110 00110100 00100000 00110100 00110011 00100000 00110100 01100100 00
100000 00110110 01100100 00100000 00110101 00110110 00100000 00110100 00111000 00100000 00110100 00110100 001000
00 00110011 00110101 00100000 00110110 00110001 00100000 00110110 00110100 00100000 00110011 00111001 00100000 0
0110111 00110101 00100000 00110100 00110111 00100000 00110000 01100001
```

这个是二进制，先转为16进制得到3EP/3VNFFmNEAnlHD5dCMmVHD5ad9uG

这个有点像base64，但是解不出来……

尝试了其他的，还是没思路

现在来分析一下base64的编码表

字符	值	字符	值	字符	值	字符	值
A	0	Q	16	g	32	w	48
B	1	R	17	h	33	x	49
C	2	S	18	i	34	y	50
D	3	T	19	j	35	z	51
E	4	U	20	k	36	0	52
F	5	V	21	l	37	1	53
G	6	W	22	m	38	2	54
H	7	X	23	n	39	3	55
I	8	Y	24	o	40	4	56
J	9	Z	25	p	41	5	57
K	10	a	26	q	42	6	58
L	11	b	27	r	43	7	59
M	12	c	28	s	44	8	60
N	13	d	29	t	45	9	61
O	14	e	30	u	46	+	62
P	15	f	31	v	47	/	63

flag经过base64编码后是ZmxhZw==

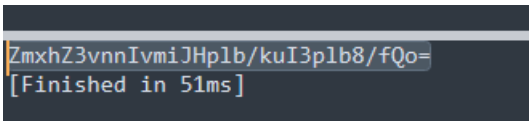
现在把前四位对比一下 Zmxh和3EP/，他们在base64的编码表中相差30，所以，这一串也都是相差30个之后的结果

找到了羽师傅的脚本

```
#author 羽
s= '3EP/3VNFFmNEAnlHD5dCMmVHD5ad9uG'
t = 'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/'
l=""
for i in s:
    l += t[(t.index(i)-30)%64]

if len(l)%4!=0:
    l=l+"="*(4-(len(l)%4))
print(l)
```

运行一下就得到了base64编码后的flag



再去解码就得到了flag