

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](#)

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/weixin_52444045/article/details/123513984

版权



[crypto](#) 专栏收录该内容

5 篇文章 0 订阅

订阅专栏

文章目录

密码学签到

[crypto2](#)

[crypto3](#)

[crypto4](#)

[crypto5](#)

[crypto6](#)

[crypto7](#)

[crypto8](#)

[crypto9](#)

[crypto10](#)

[crypto11](#)

[crypto0](#)

[crypto12](#)

[crypto13](#)

[crypto14](#)

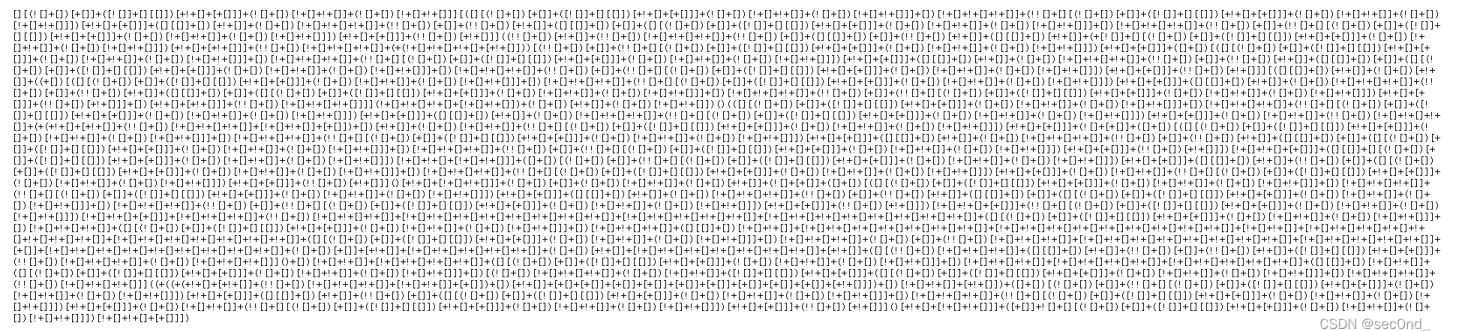
密码学签到

```
}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

Screenshot of a browser's developer console. The 'Console' tab is active, showing a long, repetitive string of characters. The text below the console is: flag[3e858ccd79287cfe8509f15a71b4c45d]. There are also navigation icons and the CSDN logo visible.

crypto3

Screenshot of a browser displaying a CTF challenge page. The address bar shows 'https://ctf.show/files/42a50afdc...'. The page content shows a heavily obfuscated code block with many escaped characters and symbols, intended for decryption. The CSDN logo is visible in the bottom right corner.

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

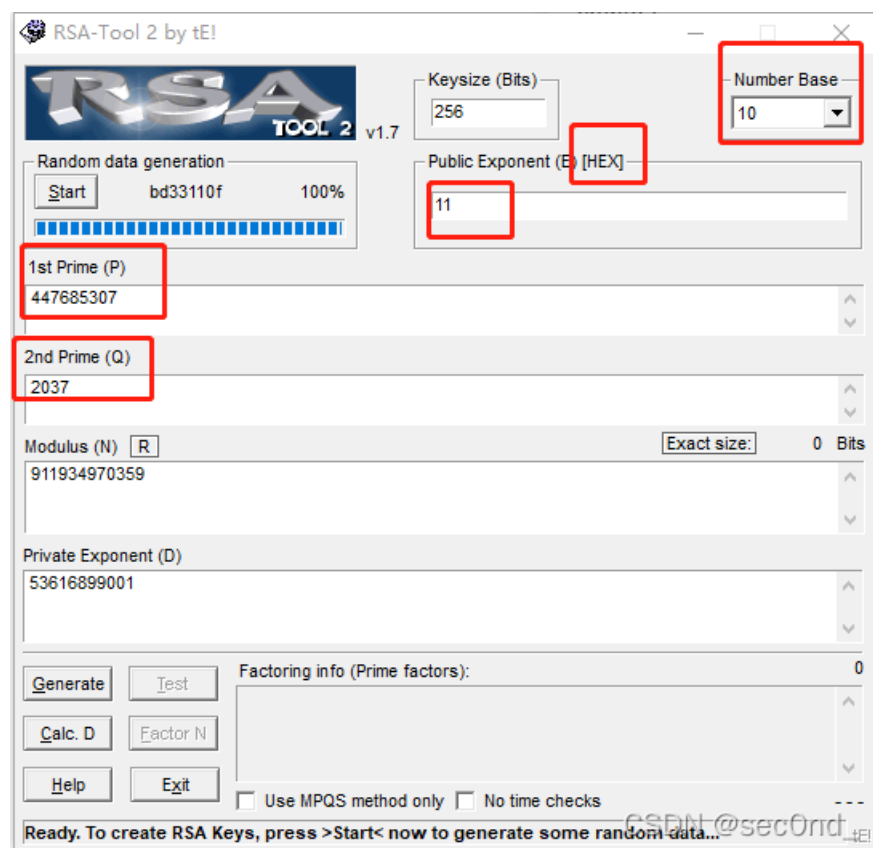
crypto4

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



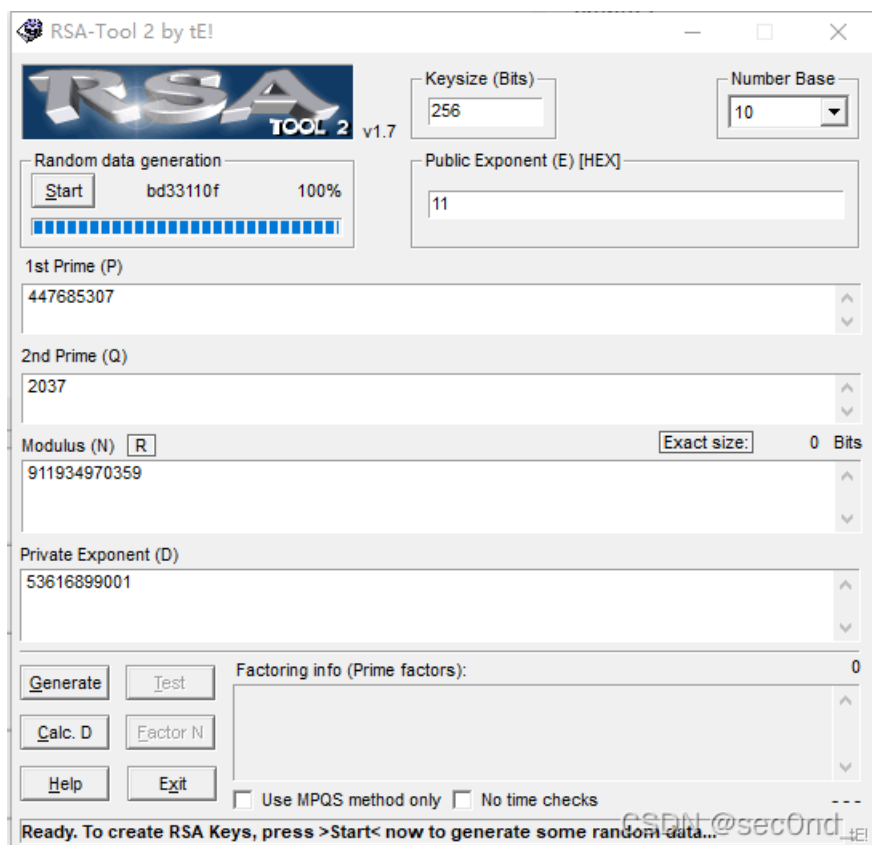
输入好之后，点击求D即可

crypto5

p=447685307 q=2037 e=17 c=704796792

提交flag{m}

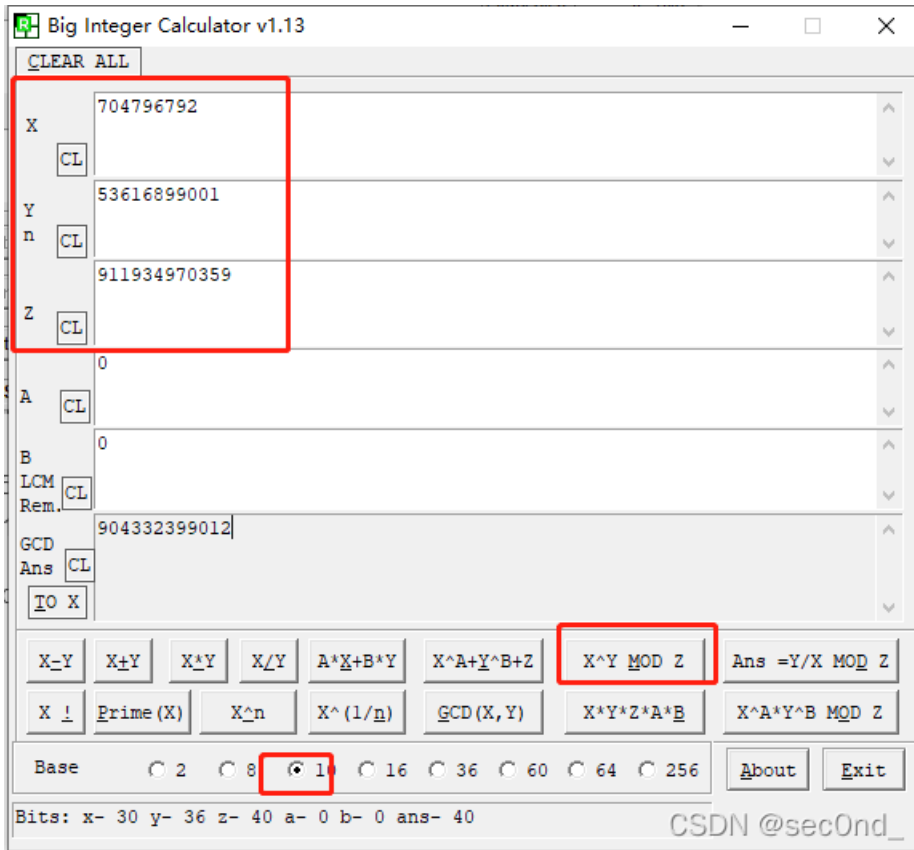
先用工具计算其他值



$n=911934970359$

$d=53616899001$

解密步骤: $m = c^d \pmod n$
使用工具big integer calculate



crypto6

密钥为 加密方式 名称, 区分大小写

U2FsdGVkX19mGsG1fI3nciNVpWZZRqZ02PYjJ1ZQuRqoiknyHSWeQv8o10uRZP94
MqeD2xz+

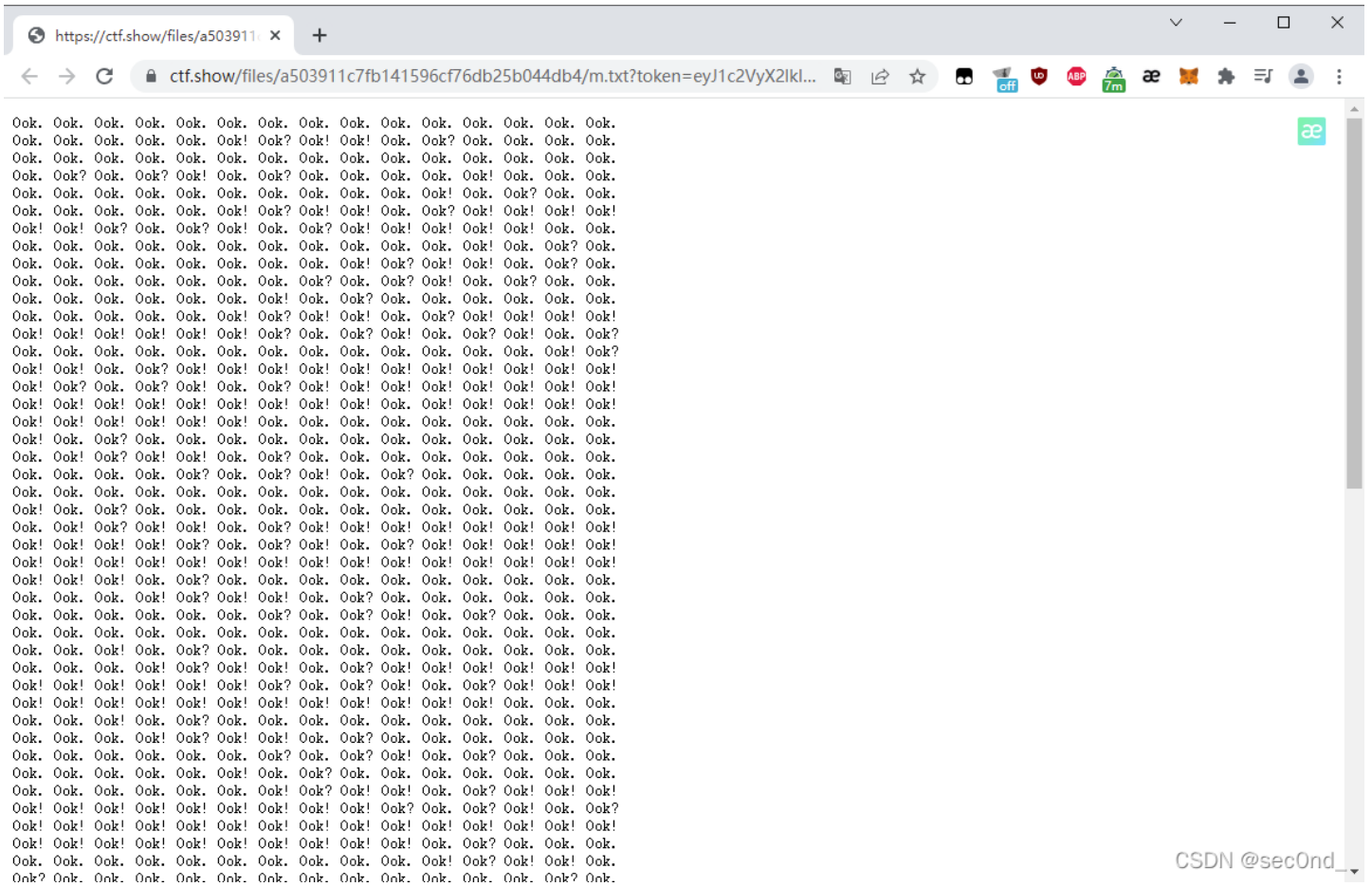
这个是Rabbit加密, 密钥应该就是Rabbit

[在线解密工具地址](#)

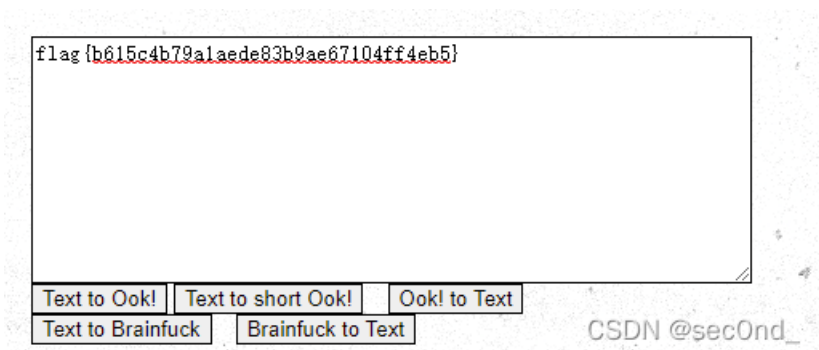


U2FsdGVkX1开头的可能是rabbit, AES, DES

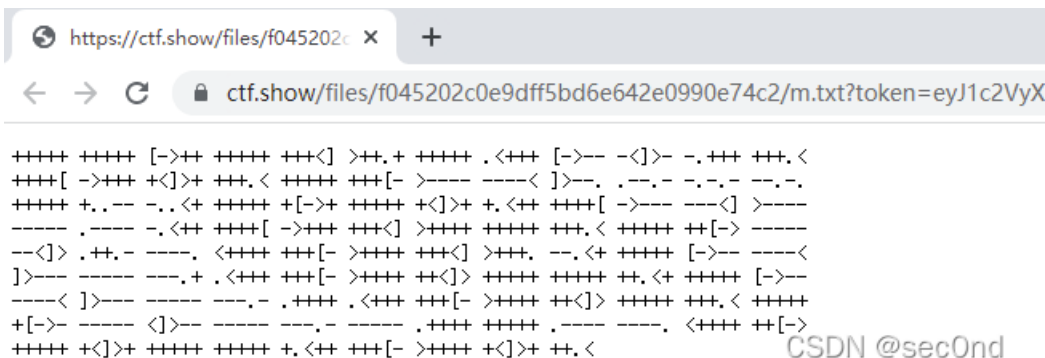
crypto7



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

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

Serpent – Symmetric Ciphers Online

Input type: File

File: C:\fakepath\odt-IV-00000000000000000000000000000000.dat

Function: SERPENT

Mode: ECB (electronic codebook)

Key (plain): 4132

Plaintext Hex

> Encrypt! > Decrypt!

100% File was uploaded.

Decrypted text:

00000000	66 6e 61 67 7b 63 39 36 30 61 30 66 33 62 66 38	f l a g { c 9 6 0 a 0 f 3 b f 8
00000010	37 31 64 37 64 61 32 61 38 34 31 33 61 65 37 38	7 1 d 7 d a 2 a 8 4 1 3 a e 7 8
00000020	66 37 62 35 66 7d 00 00 00 00 00 00 00 00 00 00	f 7 b 5 f }

[Download as a binary file] [?]

CSDN @sec0nd

得到flag

crypto10

Browser address bar: https://ctf.show/files/ea6405d...

Decrypted content: =E7=94=A8=E4=BD=A0=E9=82=A3=E7=81=AB=E7=83=AD=E7=9A=84=E5=98=B4=E5=94=87=E8=AE=A9=E6=88=91=I
8=E5=8D=88=E5=A4=9C=E9=87=8C=E6=97=A0=E5=B0=BD=E7=9A=84=E9=94=80=E9=AD=82

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！

```
--_000_0C84B99647CCED41BD7BA204FDF3A9719DF17E53EX2010C500wanco_  
Content-Type: text/html; charset="gb2312"  
Content-Transfer-Encoding: quoted-printable  
p class=3DMsoNormal<span style=3D' font-family:=CB=CE=CC=E5'>C4=BF=C7=B0=  
=CE=DE=CF=DF=D3=D0=D2=BB=B6=F6=CE=DE=CF=DF=B2=CA=C6=B1=BF=CD=BB=A7=B6=CB=BD=  
=D3=C8=EB=D6=A7=B8=B6=B1=A6=C7=AE=B0=FC=CF=EE=C4=BF=A3=AC=B8=C3=CF=EE=C4=BF=  
=D0=E8=C7=F3=D6=F7=D5=BE</span><span lang=3DEN-US>service</span><span style=  
=3D' font-family:=CB=CE=CC=E5'>BD=D3=BF=DA=CC=E1=B9=A9=D6=A7=B3=D6=A3=AC=BE=  
=D7=CC=85=DD=8A=C7=F3=FC=FB=8A=8D=8C=FE=A1=A3</span><span lang=3DEN-1S><span>
```

CSDN @secOnd_

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

Quoted-printable文本: [x] 选择字符集: utf8编码 (unicode编码) [v]

=E7=94=A8=E4=BD=A0=E9=82=A3=E7=81=AB=E7=B3=AD=E7=9A=84=E5=98=B4=E5=94=87=E8=AF=A9=E6=88=91=E5=9C=A8=E5=8D=88=E5=A4=9C=E9=87=8C=E6=97=A0=E5=B0=E7=9A=84=E9=94=80=E9=AD=82

↑ 将你电脑文件直接拖入试试 ^-^

Quoted-printable解码 Quoted-printable编码

转换结果: [x] [v] [e]

用你那火热的嘴唇让我在深夜里无尽的销魂

CSDN @secOnd_

嗯????? 这flag?????

Correct

绝了

crypto11

密文 `a8db1d82db78ed452ba0882fb9554fc`

输入让你无语的MD5

a8db1d82db78ed452ba0882fb9554fc

解密

md5

ctf

CSDN @sec0nd_

crypto0

gmbh{ifmmp_dug}

有括号哎, 应该是凯撒密码吧, 试一试

Caesar Cipher

gmbh{ifmmp_dug}

加密

密钥

解密

djye{fcjfm_ard}

ekzf{gdkkn_bse}

flag{hello_ctf}

CSDN @sec0nd_

flag{hello_ctf}

crypto12

uozt{Zgyzhv_xlww_uiln_xguhsld}

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

The screenshot shows a web application titled "Atbash". It features two text input fields. The top field contains the ciphertext "uozt{Zgyzhv_xlww_uiln_xguhsld}". Below the fields are two buttons: "原文" (Original) with an upward arrow and "埃特巴什码" (Atbash Code) with a downward arrow. The bottom field contains the plaintext "flag{Atbase_code_from_ctfshow}". At the bottom right of the interface, there is a watermark "CSDN @see0nd_".

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

```

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

```

31     file.write(str(src,'utf-8'))
32     print("Decryption complete!")
33
34
Decryption complete!
[Finished in 2.4s]

```

result.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
flag{b4Se_Fami1y_Is_FUn}

CSDN @secOnd_

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/3VNFFmNEAnIHD5dCMmVHD5ad9uG

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

```
ZmxhZ3vnnIvmiJHp1b/kuI3plb8/fQo=
[Finished in 51ms]
```

再去解码就得到了flag