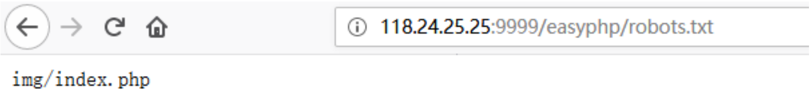
Week2-Theffth

Web:

1.easy_php

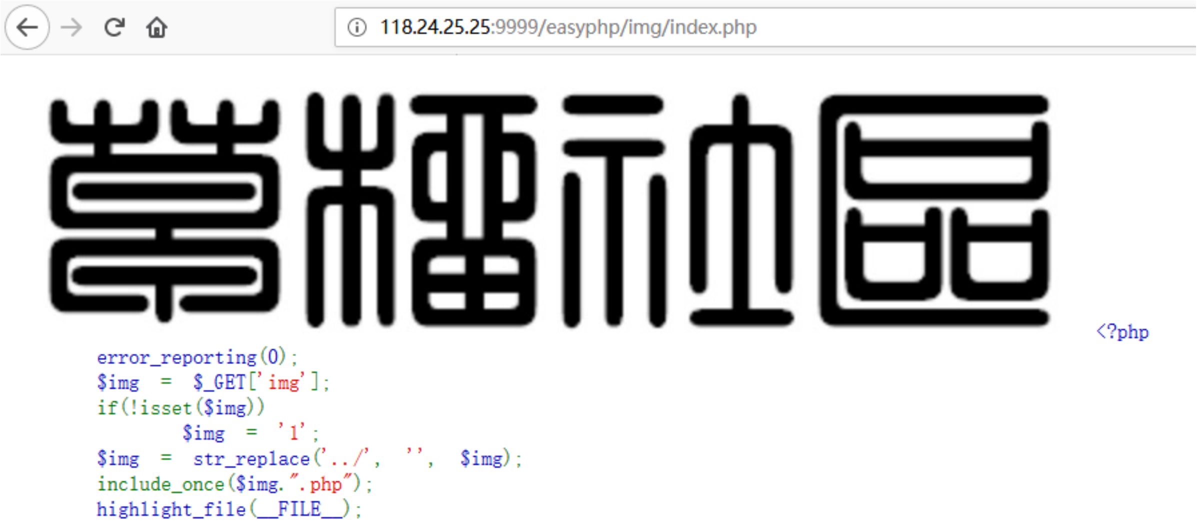
点击题目链接，看到标题为:Where is my robots，想到robots协定，于是：



118.24.25.25:9999/easyphp/robots.txt

```
img/index.php
```

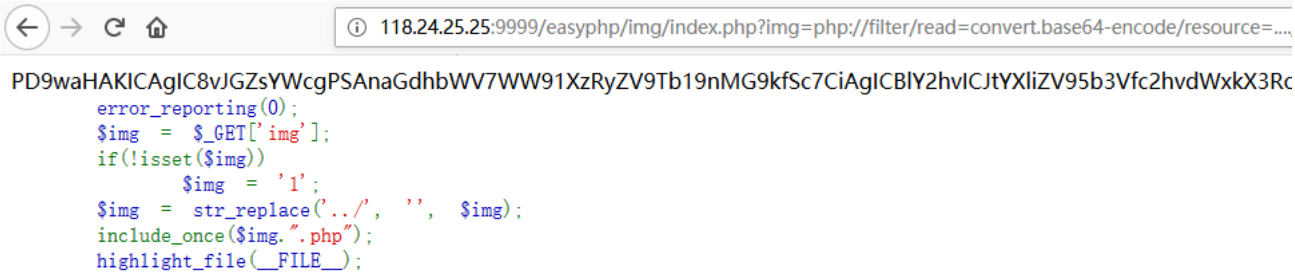
发现一个新的网页，打开，



118.24.25.25:9999/easyphp/img/index.php

```
<?php
    error_reporting(0);
    $img = $_GET['img'];
    if(!isset($img))
        $img = '1';
    $img = str_replace('../', '', $img);
    include_once($img.".php");
    highlight_file(__FILE__);
```

分析php，发现关键include_once（），想到文件包含漏洞，根据去年的wp知：include_once函数在加载../flag.php会解析flag.php文件导致不能显示flag.php的内容，然而利用PHP伪协议，又因为代码第五行会把../部分去掉，于是利用双写绕过：

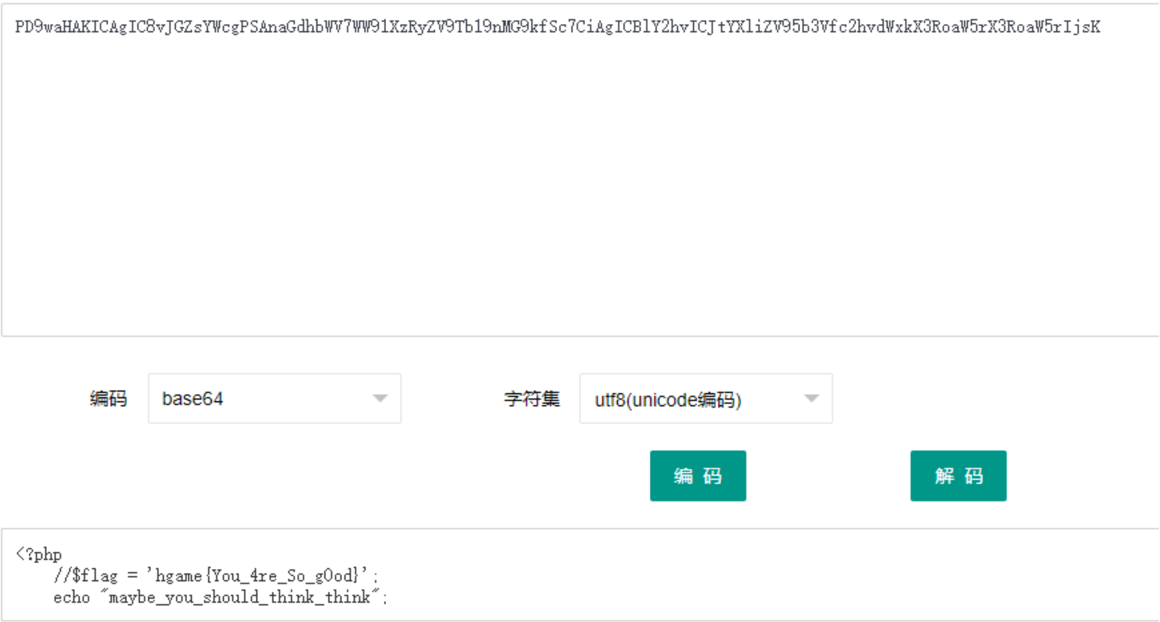


118.24.25.25:9999/easyphp/img/index.php?img=php://filter/read=convert.base64-encode/resource=....

PD9waHAKICAgIC8vJGZsYWcgPSAnaGdhbWV7WW91XzRyZV9Tb19nMG9kfSc7CiAgICBlY2hvICJtYXliZV95b3Vfc2hvdWxkX3R

```
    error_reporting(0);
    $img = $_GET['img'];
    if(!isset($img))
        $img = '1';
    $img = str_replace('../', '', $img);
    include_once($img.".php");
    highlight_file(__FILE__);
```

把源文件的base64编码解码，得到flag:hgame{You_4re_So_gOod}



PD9waHAKICAgIC8vJGZsYWcgPSAnaGdhbWV7WW91XzRyZV9Tb19nMG9kfSc7CiAgICBlY2hvICJtYXliZV95b3Vfc2hvdWxkX3RoaW5rX3RoaW5rIjsK

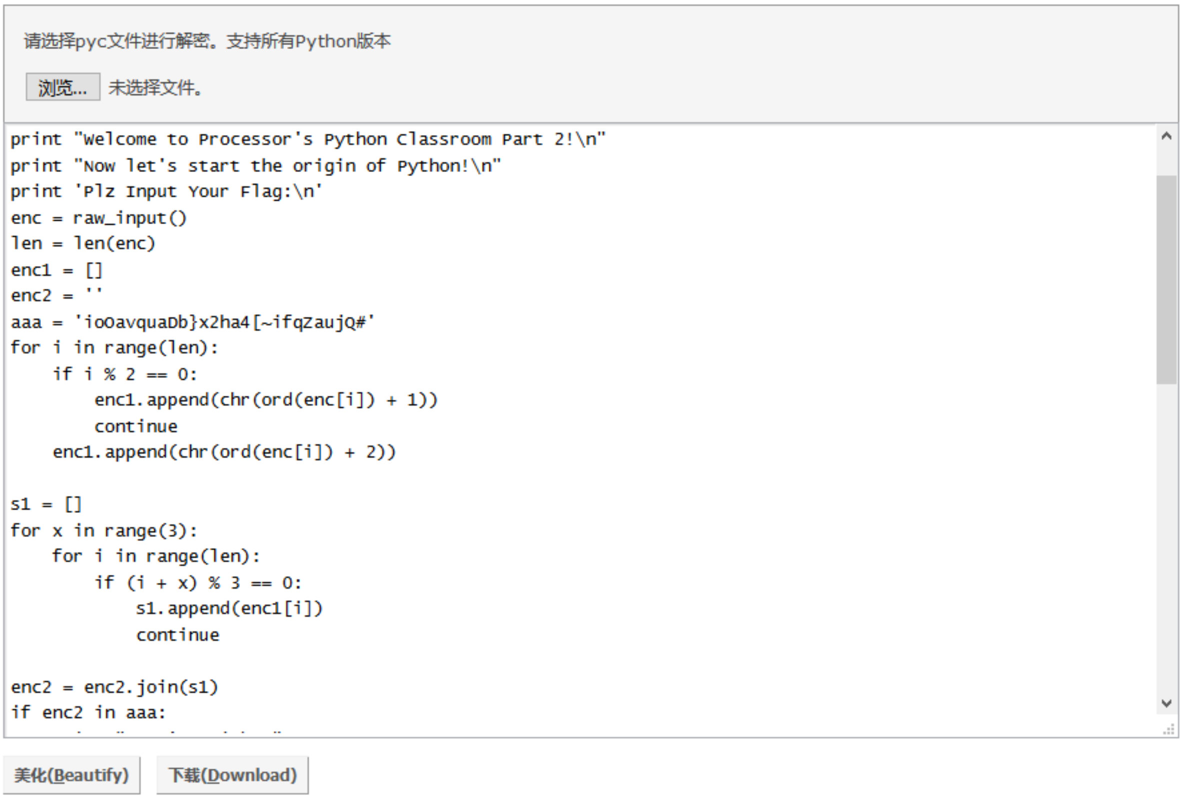
编码 base64 字符集 utf8(unicode编码)

```
<?php
    //$flag = 'hgame{You_4re_So_g0od}';
    echo "maybe_you_should_think_think";
```

Re:

1.Pro的Python教室（二）

看到题目是.pyc文件，于是python在线反编译，



请选择pyc文件进行解密。支持所有Python版本

```
print "Welcome to Processor's Python Classroom Part 2!\n"
print "Now let's start the origin of Python!\n"
print 'Plz Input Your Flag:\n'
enc = raw_input()
len = len(enc)
enc1 = []
enc2 = ''
aaa = 'ioOavquaDb}x2ha4[~ifqZaujQ#'
for i in range(len):
    if i % 2 == 0:
        enc1.append(chr(ord(enc[i]) + 1))
        continue
    enc1.append(chr(ord(enc[i]) + 2))

s1 = []
for x in range(3):
    for i in range(len):
        if (i + x) % 3 == 0:
            s1.append(enc1[i])
            continue

enc2 = enc2.join(s1)
if enc2 in aaa:
```

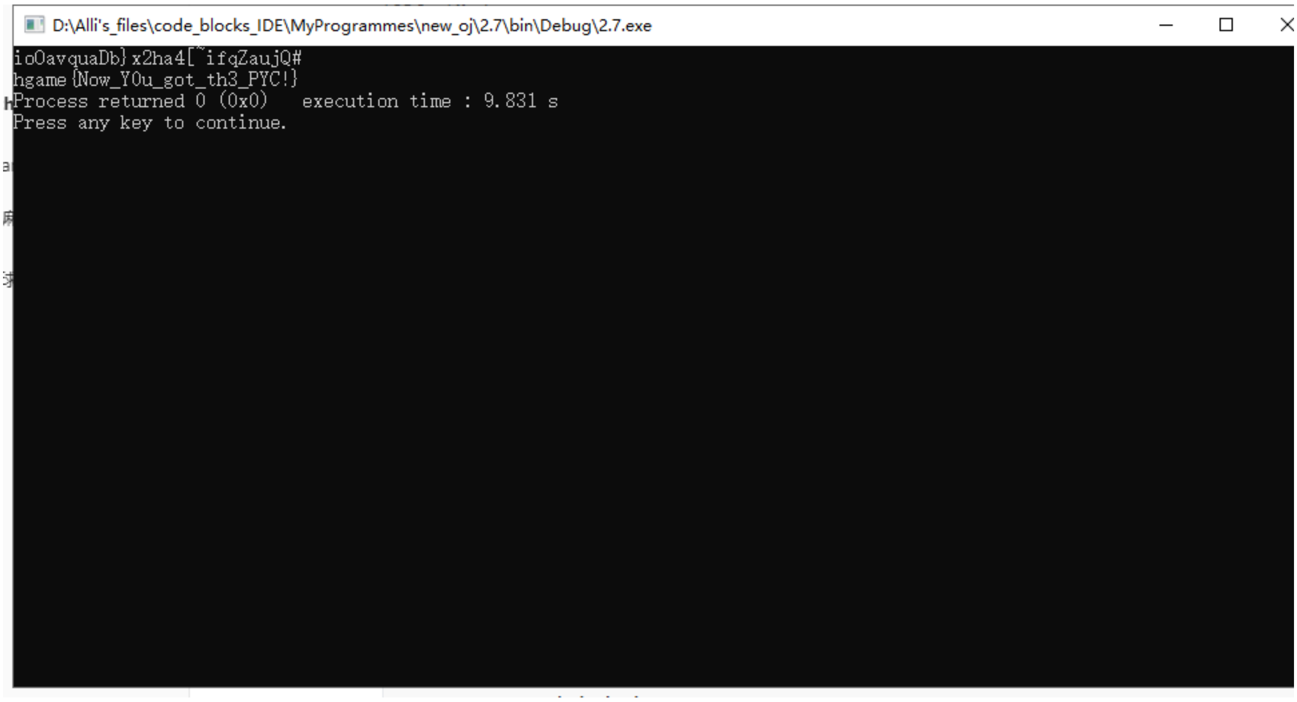
得到一段Python:

```
x

print "Welcome to Processor's Python Classroom Part 2!\n"

print "Now let's start the origin of Python!\n"

print 'Plz Input Your Flag:\n'

enc = raw_input()

len = len(enc)

enc1 = []

enc2 = ''

aaa = 'ioOavquaDb}x2ha4[~ifqZaujQ#'

for i in range(len):
    if i % 2 == 0:
        enc1.append(chr(ord(enc[i]) + 1))
        continue
    enc1.append(chr(ord(enc[i]) + 2))


s1 = []
for x in range(3):
    for i in range(len):
        if (i + x) % 3 == 0:
            s1.append(enc1[i])
            continue


enc2 = enc2.join(s1)
if enc2 in aaa:
    print "You 're Right!"
else:
    print "You're Wrong!"
    exit(0)
```

分析一下，写了以下C：

```
xxxxxxxxxx
#include <stdio.h>
#include <stdlib.h>


int main()
{
    char  a[256],b[256],c[256];
    scanf("%s",a);
    int i,j,m=0;
    for(i=0;i<9;i++)
    {
        b[3*i]=a[i];
    }
    for(i=1;i<=2;i++)
    {
        for(j=1;j<=9;j++)
        {
            b[3*j-i]=a[i*9+j-1];
        }
    }
    for(i=0;b[i]!='\0';i++)
    {
        if(i%2==0)
        {
            c[i]=b[i]-1;
        }
        else
        {
            c[i]=b[i]-2;
        }
    }
    printf("%s",c);

    return 0;

}
```

于是得到flag:hgame{Now_Y0u_got_th3_PYC！}



D:\Alli's_files\code_blocks_IDE\MyProgrammes\new_oj\2.7\bin\Debug\2.7.exe

```
ioOavquaDb}x2ha4[~ifqZaujQ#
hgame{Now_Y0u_got_th3_PYC!}
Process returned 0 (0x0)   execution time : 9.831 s
Press any key to continue.
```

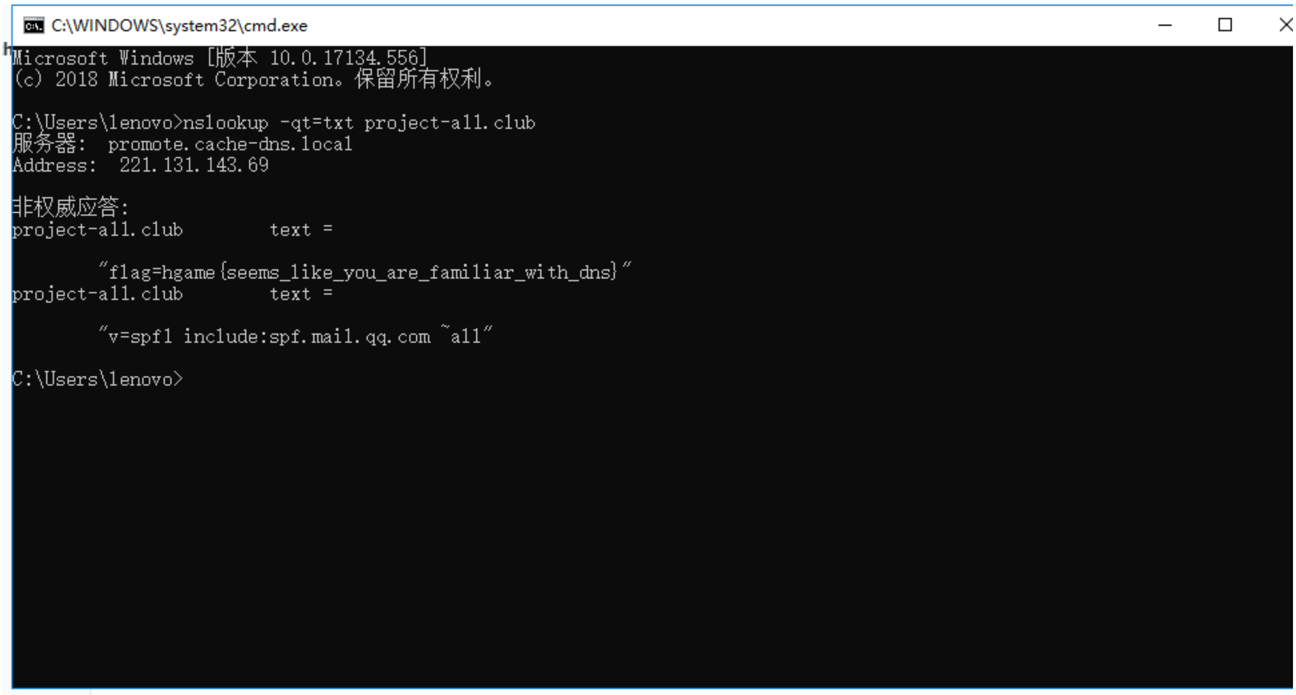
Misc:

1.Are You Familiar with DNS Records?

打开题目，发现网页打不开，根据标题DNS记录，搜索得知txt记录可用于隐藏一些信息，常用于反垃圾邮件，用wireshark查看流量包，找不到txt记录，于是搜索txt记录查看方式，根据

> **如何用本地计算机查询DNS记录?**
>
> 打开命令提示符窗口(开始--运行---输入CMD--回车)
>
> nslookup的语法为 nslookup –qt=类型 目标域名(注意qt必须小写)

试着查询了一下：



C:\WINDOWS\system32\cmd.exe

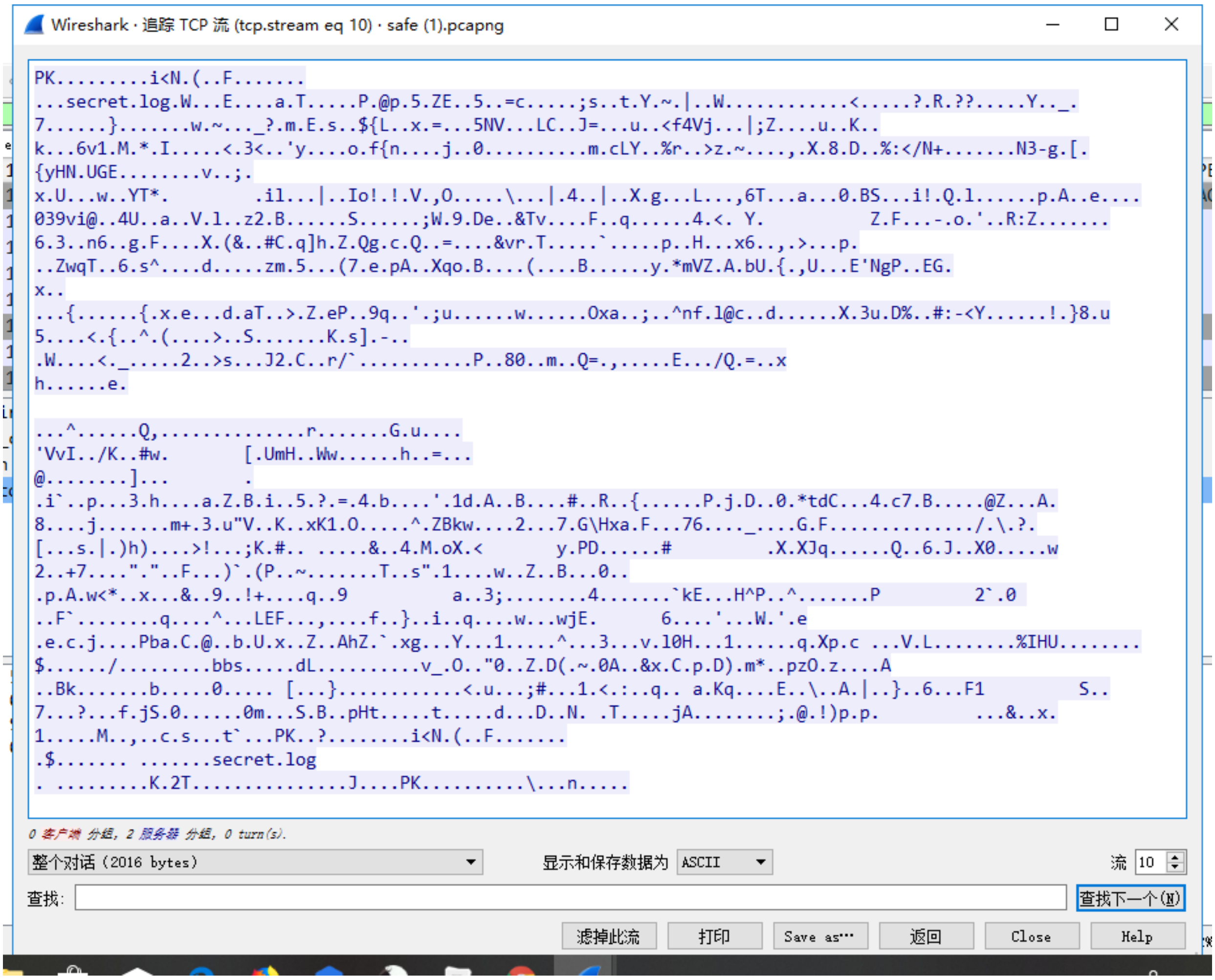
```
Microsoft Windows [版本 10.0.17134.556]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\lenovo>nslookup -qt=txt project-all.club
服务器:  promote.cache-dns.local
Address:  221.131.143.69

非权威应答:
project-all.club        text =

        "flag=hgame{seems_like_you_are_familiar_with_dns}"
project-all.club        text =

        "v=spf1 include:spf.mail.qq.com ~all"

C:\Users\lenovo>
```

找到flag:hgame{seems_like_you_are_familiar_with_dns}

参考资料：https://xz.aliyun.com/t/1942，http://tool.chinaz.com/nslookup/

2.找得到我嘛？小火汁

用wireshark放流量包，没有啥子思路，后来有了hint：https，想到https=http+SSL(TLS)，再结合名字safe想到wireshark对https解密，于是目标就是找到解密的私钥，根据对tcp流的追踪，到第八层的时候发现secret的压缩包，继续追踪发现：


Wireshark · 追踪 TCP 流 (tcp.stream eq 10) · safe (1).pcapng
查找下一个(N)
滤掉此流
打印
Save as…
返回
Close
Help

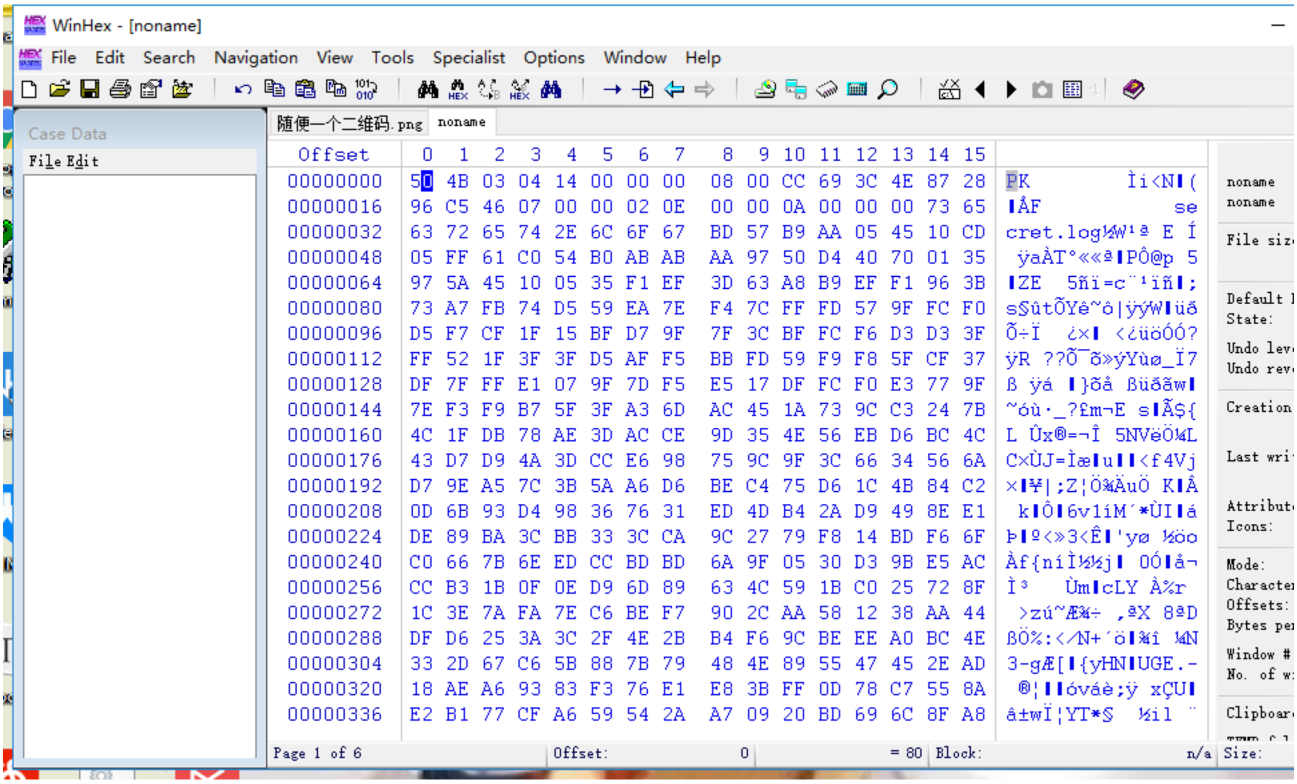
```
PK.........i<N.(..F.......
...secret.log.W...E....a.T.....P.@p.5.ZE..5..=c.....;s..t.Y.~.|..W............<.....?.R.??.....Y.._.
7......}.......w.~..._?.m.E.s..${L..x.=...5NV...LC..J=...u..<f4Vj...|;Z....u..K..
k...6v1.M.*.I.....<.3<..'y....o.f{n....j..0..........m.cLY..%r..>z.~....,.X.8.D..%:</N+.......N3-g.[.
{yHN.UGE........v..;.
x.U...w..YT*.        .il...|..Io!.!.V.,O.....\...|.4..|..X.g...L...,6T...a...0.BS...i!.Q.l......p.A..e....
039vi@..4U..a..V.l..z2.B......S......;W.9.De..&Tv....F..q......4.<. Y.          Z.F...-.o.'..R:Z.......
6.3..n6..g.F....X.(&..#C.q]h.Z.Qg.c.Q..=....&vr.T.....`.....p..H...x6..,.>...p.
..ZwqT..6.s^....d.....zm.5...(7.e.pA..Xqo.B....(....B......y.*mVZ.A.bU.{.,U...E'NgP..EG.
x..
...{......{.x.e...d.aT..>.Z.eP..9q..'.;u......w......Oxa..;..^nf.l@c..d......X.3u.D%..#:-<Y......!.}8.u
5....<.{..^.(....>..S.......K.s].-..
.W....<._.....2..>s...J2.C..r/`..........P..80..m..Q=.,.....E.../Q.=..x
h......e.

...^......Q,..............r.......G.u....
'VvI../K..#w.       [.UmH..Ww......h..=...
@........]...       .
.i`..p...3.h....a.Z.B.i..5.?.=.4.b....'.1d.A..B....#..R..{......P.j.D..0.*tdC...4.c7.B.....@Z...A.
8....j.......m+.3.u"V..K..xK1.O.....^.ZBkw....2...7.G\Hxa.F...76...._....G.F............../.\.?.
[...s.|.)h)....>!...;K.#.. .....&..4.M.oX.<       y.PD......#          .X.XJq......Q..6.J..X0.....w
2..+7....".."..F...)`.(P..~.......T..s".1....w..Z..B...0..
.p.A.w<*..x...&..9..!+....q..9          a..3;........4.......`kE...H^P..^.......P        2`.0
..F`........q....^...LEF...,....f..}..i..q....w...wjE.      6....'...W.'.e
.e.c.j....Pba.C.@..b.U.x..Z..AhZ.`.xg...Y...1.....^...3...v.10H...1......q.Xp.c ...V.L........%IHU........
$....../.........bbs.....dL..........v_.O.."0..Z.D(.~.0A..&x.C.p.D).m*..pzO.z....A
..Bk.......b.....0..... [...}...........<.u...;#...1.<.:..q.. a.Kq....E..\..A.|..}..6...F1         S..
7...?...f.jS.0......0m...S.B..pHt.....t.....d...D..N. .T.....jA........;.@.!)p.p.         ...&..x.
1.....M..,..c.s...t`...PK..?........i<N.(..F.......
.$....... .......secret.log
. .........K.2T...............J....PK..........\...n.....
```

0 客户端 分组, 2 服务器 分组, 0 turn(s).

整个对话 (2016 bytes)　　显示和保存数据为 ASCII　　流 10

查找：

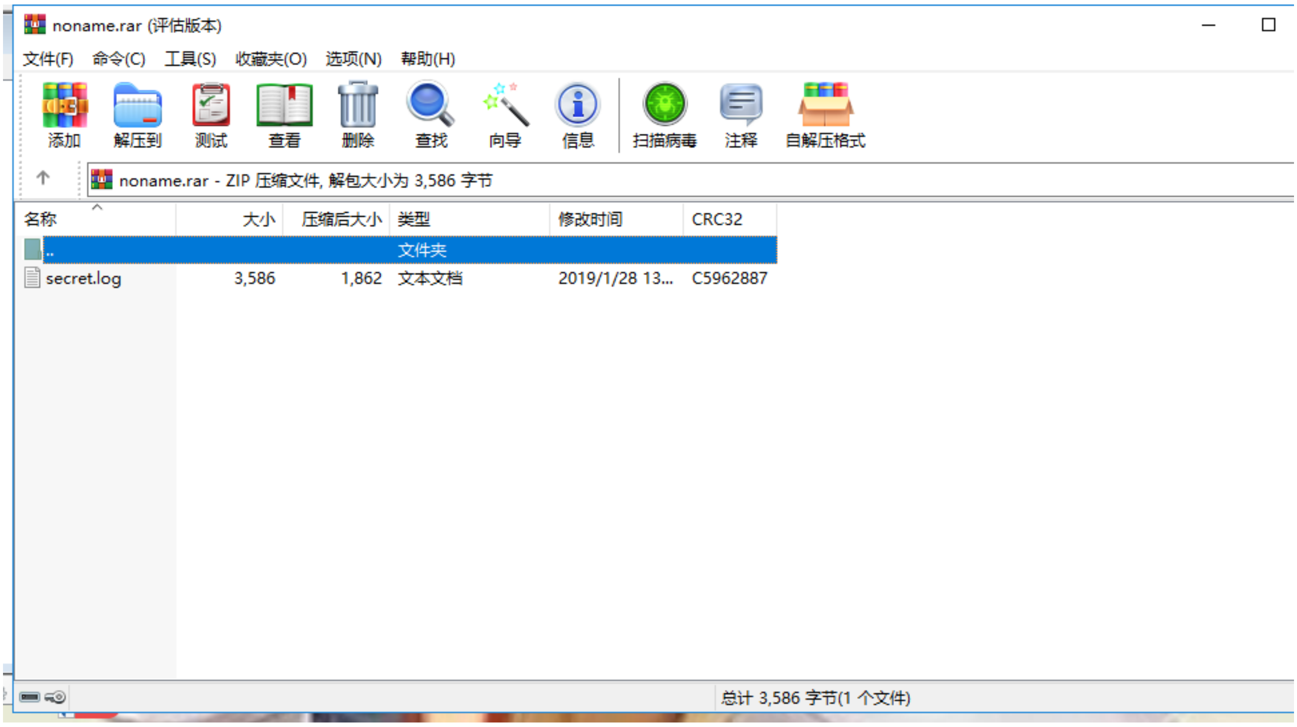
找到关键字：secret.log，又是以PK开头，是压缩包形式，以原数据形式导出复制到winhex里：



```
 Offset     0  1  2  3  4  5  6  7   8  9 10 11 12 13 14 15

00000000   50 4B 03 04 14 00 00 00  08 00 CC 69 3C 4E 87 28   PK          Ìi<N▌(
00000016   96 C5 46 07 00 00 02 0E  00 00 0A 00 00 00 73 65   ▌ÅF               se
00000032   63 72 65 74 2E 6C 6F 67  BD 57 B9 AA 05 45 10 CD   cret.log½W¹ª E Í
00000048   05 FF 61 C0 54 B0 AB AB  AA 97 50 D4 40 70 01 35    ÿaÀT°««ª▌PÔ@p 5
00000064   97 5A 45 10 05 35 F1 EF  3D 63 A8 B9 EF F1 96 3B   ▌ZE  5ñï=c¨¹ïñ▌;
00000080   73 A7 FB 74 D5 59 EA 7E  F4 7C FF FD 57 9F FC F0   s§ûtÕYê~ô|ÿýW▌üð
00000096   D5 F7 CF 1F 15 BF D7 9F  7F 3C BF FC F6 D3 D3 3F   Õ÷Ï  ¿×▌ <¿üöÓÓ?
00000112   FF 52 1F 3F 3F D5 AF F5  BB FD 59 F9 F8 5F CF 37   ÿR ??Õ¯õ»ýYùø_Ï7
00000128   DF 7F FF E1 07 9F 7D F5  E5 17 DF FC F0 E3 77 9F   ß ÿá ▌}õå ßüðãw▌
00000144   7E F3 F9 B7 5F 3F A3 6D  AC 45 1A 73 9C C3 24 7B   ~óù·_?£m¬E s▌Ã${
00000160   4C 1F DB 78 AE 3D AC CE  9D 35 4E 56 EB D6 BC 4C   L Ûx®=¬Î 5NVëÖ¼L
00000176   43 D7 D9 4A 3D CC E6 98  75 9C 9F 3C 66 34 56 6A   C×ÙJ=Ìæ▌u▌▌<f4Vj
00000192   D7 9E A5 7C 3B 5A A6 D6  BE C4 75 D6 1C 4B 84 C2   ×▌¥|;Z¦Ö¾ÄuÖ K▌Â
00000208   0D 6B 93 D4 98 36 76 31  ED 4D B4 2A D9 49 8E E1    k▌Ô▌6v1íM´*ÙI▌á
00000224   DE 89 BA 3C BB 33 3C CA  9C 27 79 F8 14 BD F6 6F   Þ▌º<»3<Ê▌'yø ½öo
00000240   C0 66 7B 6E ED CC BD BD  6A 9F 05 30 D3 9B E5 AC   Àf{níÌ½½j▌ 0Ó▌å¬
00000256   CC B3 1B 0F 0E D9 6D 89  63 4C 59 1B C0 25 72 8F   Ì³   Ùm▌cLY À%r
00000272   1C 3E 7A FA 7E C6 BE F7  90 2C AA 58 12 38 AA 44    >zú~Æ¾÷ ,ªX 8ªD
00000288   DF D6 25 3A 3C 2F 4E 2B  B4 F6 9C BE EE A0 BC 4E   ßÖ%:</N+´ö▌¾î ¼N
00000304   33 2D 67 C6 5B 88 7B 79  48 4E 89 55 47 45 2E AD   3-gÆ[▌{yHN▌UGE.-
00000320   18 AE A6 93 83 F3 76 E1  E8 3B FF 0D 78 C7 55 8A    ®¦▌▌óváè;ÿ xÇU▌
00000336   E2 B1 77 CF A6 59 54 2A  A7 09 20 BD 69 6C 8F A8   â±wÏ¦YT*§  ½il ¨
```

Page 1 of 6　　Offset: 0　　= 80　　Block: n/a

再导出查看，找到私钥：

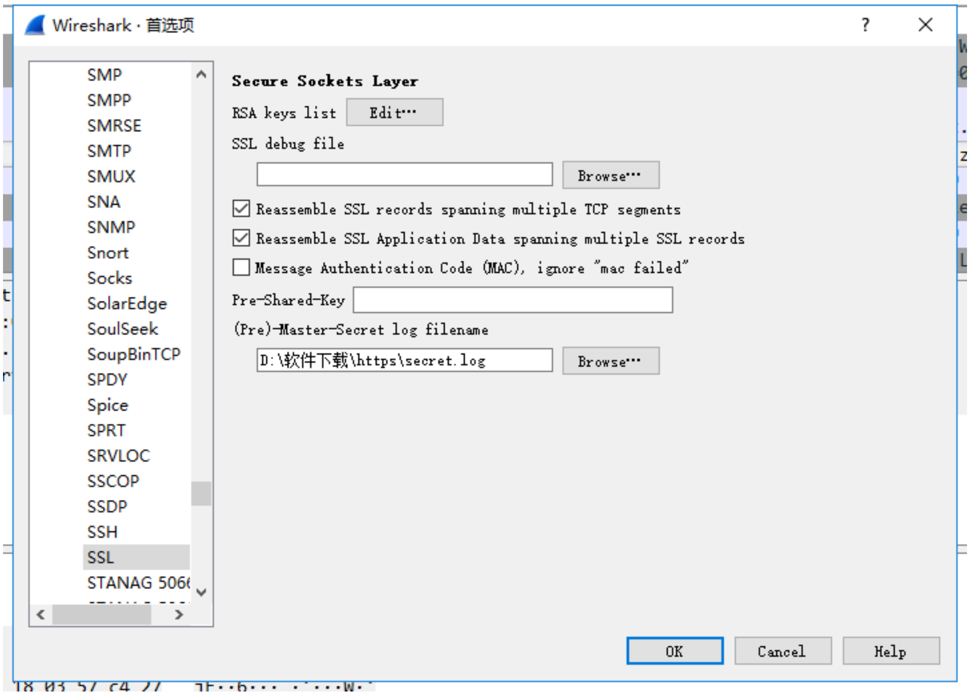


noname.rar - ZIP 压缩文件, 解包大小为 3,586 字节

名称	大小	压缩后大小	类型	修改时间	CRC32
..			文件夹		
secret.log	3,586	1,862	文本文档	2019/1/28 13...	C5962887

总计 3,586 字节(1 个文件)

于是先设置环境变量，保存私钥，再跑到wireshark里设置一下：



Secure Sockets Layer

RSA keys list Edit…

SSL debug file

Browse…

☑ Reassemble SSL records spanning multiple TCP segments

☑ Reassemble SSL Application Data spanning multiple SSL records

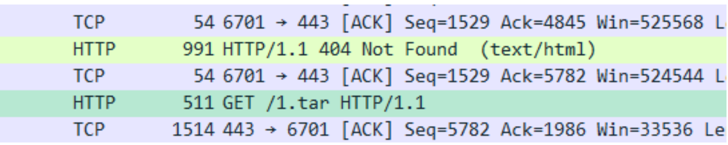
☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared-Key

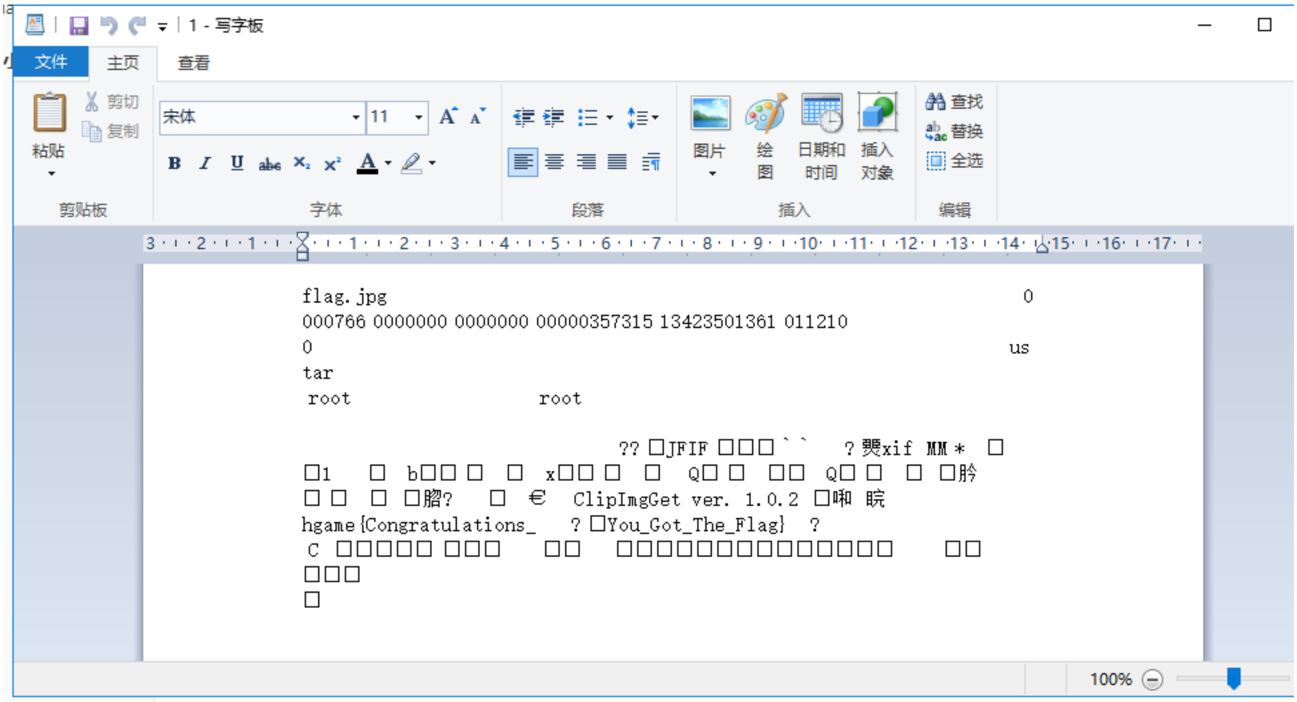
(Pre)-Master-Secret log filename

D:\软件下载\https\secret.log Browse…

解密完后，结合descrypted ssl数据找到了新出现的1.tar文件：


```
TCP        54 6701 → 443 [ACK] Seq=1529 Ack=4845 Win=525568 L
HTTP      991 HTTP/1.1 404 Not Found  (text/html)
TCP        54 6701 → 443 [ACK] Seq=1529 Ack=5782 Win=524544 L
HTTP      511 GET /1.tar HTTP/1.1
TCP      1514 443 → 6701 [ACK] Seq=5782 Ack=1986 Win=33536 Le
```

试着导出文件查看：



```
flag.jpg                                                              0
000766 0000000 0000000 00000357315 13423501361 011210
0                                                                    us
tar
 root                  root

                                ?? □JFIF □□□ ` `    ? 爂xif  MM *   □
□1    □  b□□ □   □  x□□ □   □   Q□ □   □□   Q□ □   □  □眕
□ □   □  □膪?    □   €  ClipImgGet ver. 1.0.2 □啣 眖
hgame{Congratulations_    ? □You_Got_The_Flag}   ?
 C  □□□□□ □□□    □□    □□□□□□□□□□□□□□    □□
□□□
□
```

在一堆乱码中看到hgame字样，尝试提交发现过了，可能是打开方式的问题...emmm，于是flag:hgame{Congratulations_You_Got_The_Flag}

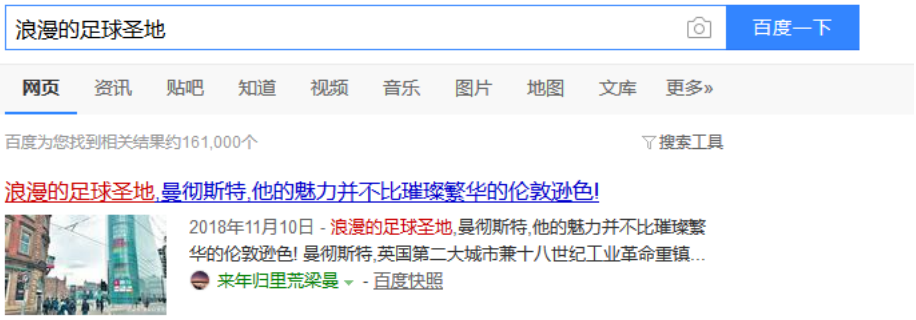
参考资料：https://zhuanlan.zhihu.com/p/36669377，http://blog.51cto.com/yttitan/1737904，

https://xz.aliyun.com/t/1966

Crypto:

1.浪漫的足球圣地：

看题目描述，emmm，没看懂，自动百度：

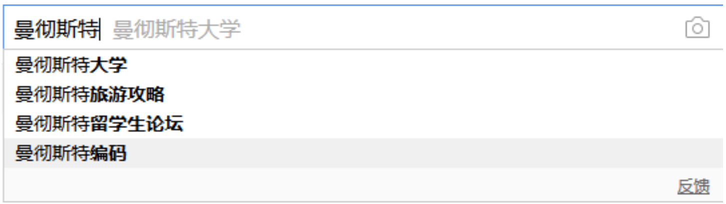


浪漫的足球圣地,曼彻斯特,他的魅力并不比璀璨繁华的伦敦逊色!

2018年11月10日 - 浪漫的足球圣地,曼彻斯特,他的魅力并不比璀璨繁华的伦敦逊色! 曼彻斯特,英国第二大城市兼十八世纪工业革命重镇...

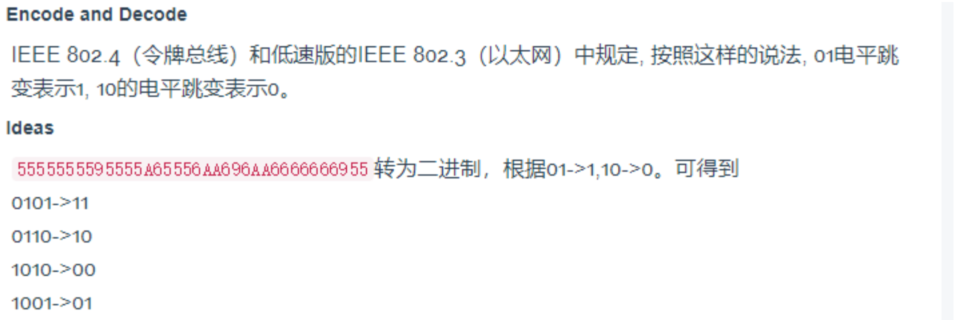
来年归里荒梁曩 - 百度快照

再次百度：



于是理解了出题人的意图...

根据：



Encode and Decode

IEEE 802.4（令牌总线）和低速版的IEEE 802.3（以太网）中规定, 按照这样的说法, 01电平跳变表示1, 10的电平跳变表示0。

Ideas

`5555555595555A65556AA696AA6666666955`转为二进制，根据01->1,10->0。可得到

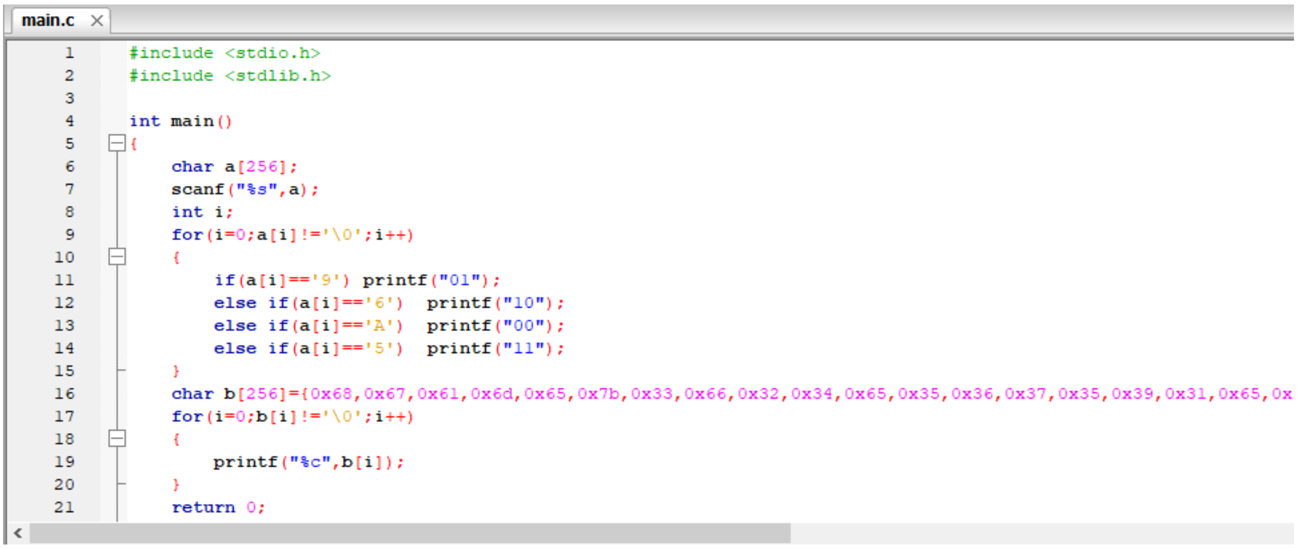
0101->11

0110->10

1010->00

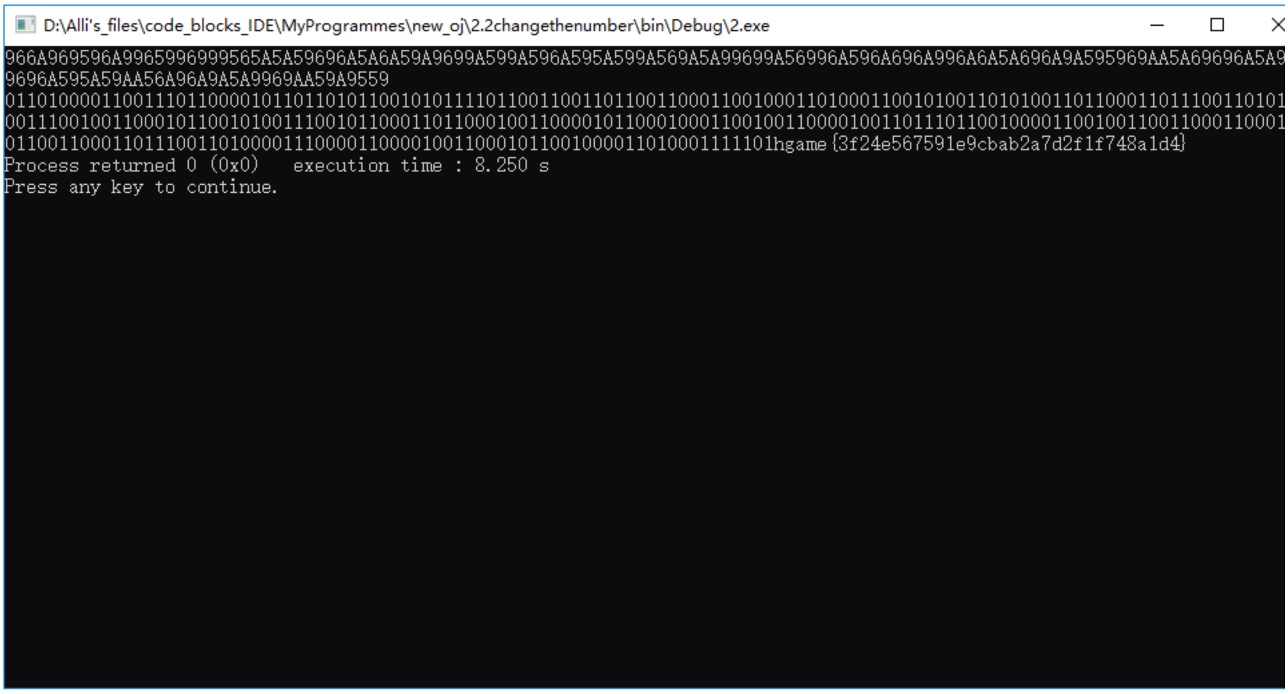
1001->01

编写了：



```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    char a[256];
    scanf("%s",a);
    int i;
    for(i=0;a[i]!='\0';i++)
    {
        if(a[i]=='9') printf("01");
        else if(a[i]=='6')  printf("10");
        else if(a[i]=='A')  printf("00");
        else if(a[i]=='5')  printf("11");
    }
    char b[256]={0x68,0x67,0x61,0x6d,0x65,0x7b,0x33,0x66,0x32,0x34,0x65,0x35,0x36,0x37,0x35,0x39,0x31,0x65,0x
    for(i=0;b[i]!='\0';i++)
    {
        printf("%c",b[i]);
    }
    return 0;
```

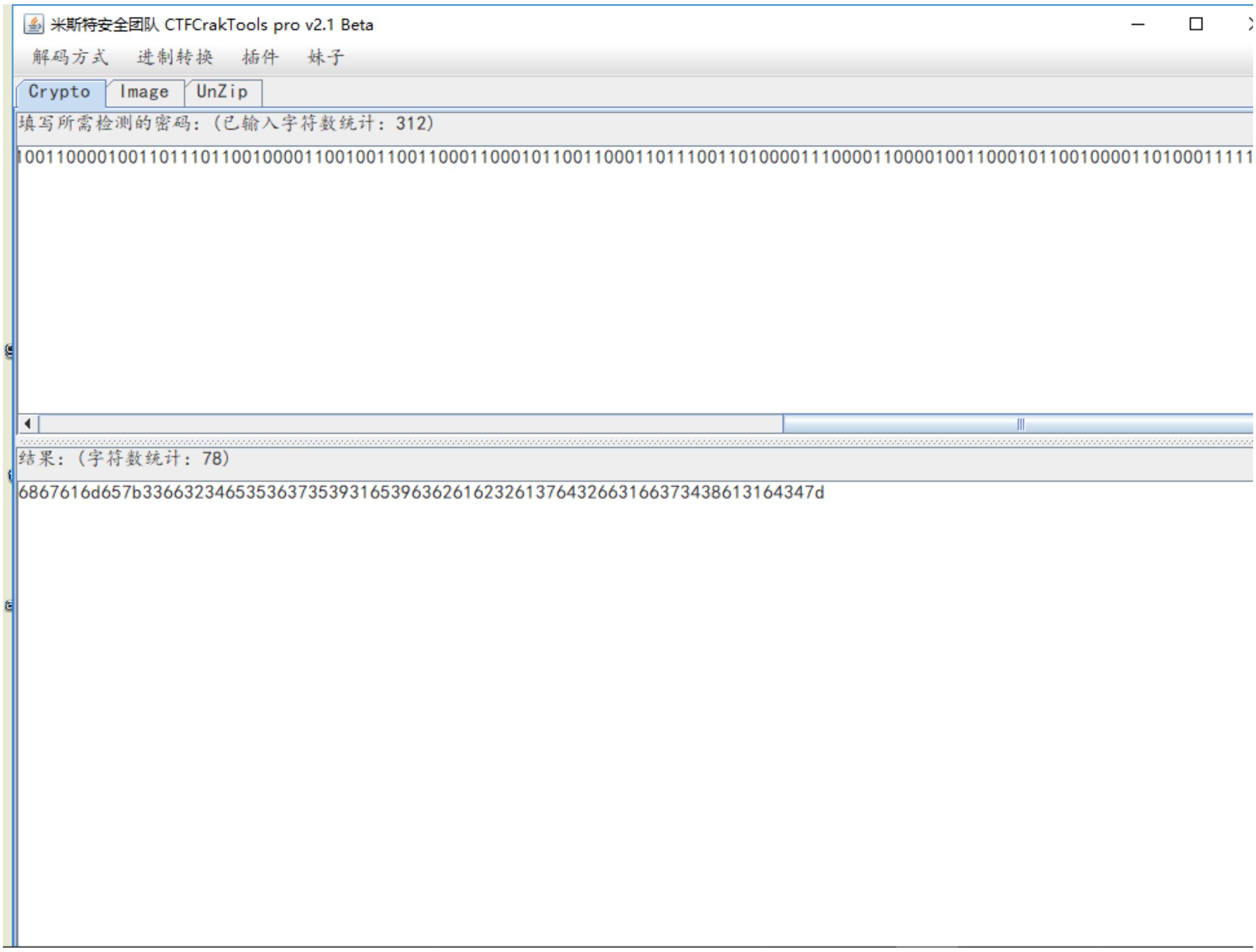
得到：



D:\Alli's_files\code_blocks_IDE\MyProgrammes\new_oj\2.2changethenumber\bin\Debug\2.exe

```
966A969596A9965996999565A5A59696A5A6A59A9699A599A596A595A599A569A5A99699A56996A596A696A996A6A5A696A9A595969AA5A69696A5A9
9696A595A59AA56A96A9A5A9969AA59A9559
011010000110011101100001011011010110010101111011001100110110011000110010001101000110010100110101001101100011011100110101
001110010011000101100101001110010110001101100010011000010110001000110010011000010011011101100100001100100110011000110001
011001100011011100110100001110000110000100110001011001000011010001111101hgame{3f24e567591e9cbab2a7d2f1f748a1d4}
Process returned 0 (0x0)   execution time : 8.250 s
Press any key to continue.
```

二进制转十六进制：



填写所需检测的密码：(已输入字符数统计：312)

```
1001100001001101110110010000110010011001100011000101100110001101110011010000111000011000010011000101100100001101000111111
```

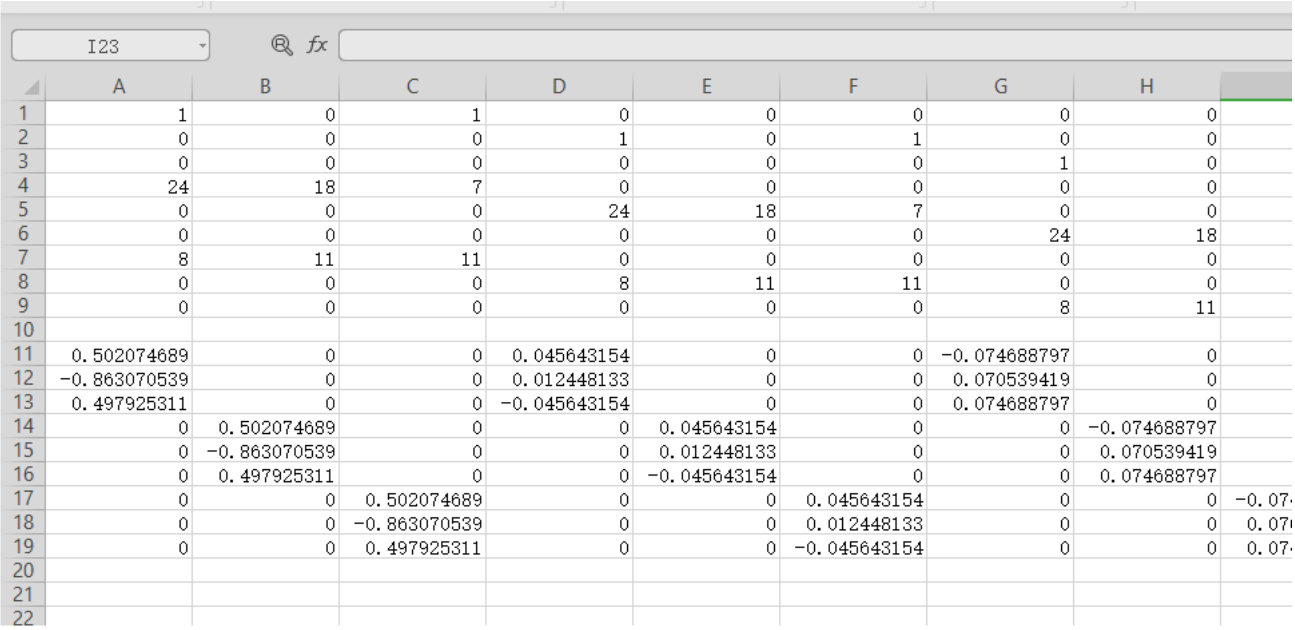
结果：(字符数统计：78)

```
6867616d657b3366323465353637353931653963626162326137643266316637343861316434 7d
```

再由ASCII码得到上上图的flag:hgame{3f24e567591e9cbab2a7d2f1f748a1d4}

2.Hill

一开始先理解了hill密码的加密原理，即：D*明文=密文（mod26)；再看九个字母对应密钥里的九个未知量想到解九元一次方程组，结果尝试了多次：



	A	B	C	D	E	F	G	H	
1	1	0	1	0	0	0	0	0	
2	0	0	0	1	0	1	0	0	
3	0	0	0	0	0	0	1	0	
4	24	18	7	0	0	0	0	0	
5	0	0	0	24	18	7	0	0	
6	0	0	0	0	0	0	24	18	
7	8	11	11	0	0	0	0	0	
8	0	0	0	8	11	11	0	0	
9	0	0	0	0	0	0	8	11	
10									
11	0.502074689	0	0	0.045643154	0	0	-0.074688797	0	
12	-0.863070539	0	0	0.012448133	0	0	0.070539419	0	
13	0.497925311	0	0	-0.045643154	0	0	0.074688797	0	
14	0	0.502074689	0	0	0.045643154	0	0	-0.074688797	
15	0	-0.863070539	0	0	0.012448133	0	0	0.070539419	
16	0	0.497925311	0	0	-0.045643154	0	0	0.074688797	
17	0	0	0.502074689	0	0	0.045643154	0	0	-0.07
18	0	0	-0.863070539	0	0	0.012448133	0	0	0.07
19	0	0	0.497925311	0	0	-0.045643154	0	0	0.07
20									
21									
22									

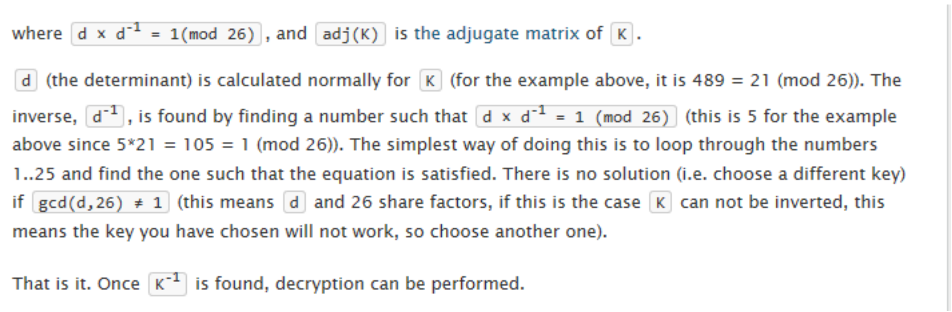
都得不到整数的密钥...，然后发现忽略了最后的密文是需要mod26这一点，又尝试手动+26，自然是失败的......

后来根据去年的wp找到hill密码的解密原理，即：D*密文=明文；D=明文 *密文^-1

于是先尝试前九个字母对应BABYSHILL，在解的过程中发现密文^-1是这样的：

$$\begin{pmatrix} 19 & 7 & 19 \\ 2 & 23 & 2 \\ 18 & 25 & 23 \end{pmatrix}^{(-1)} = \begin{pmatrix} \frac{479}{2115} & \frac{314}{2115} & \frac{-1}{5} \\ \frac{-2}{423} & \frac{19}{423} & 0 \\ \frac{-364}{2115} & \frac{-349}{2115} & \frac{1}{5} \end{pmatrix}$$

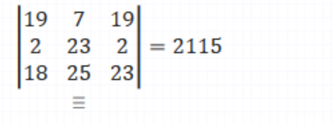
显然不是正确的数据，于是在原理的网页又找到了这一段：



> where $d \times d^{-1} = 1 (\mathrm{mod}\ 26)$, and adj(K) is the adjugate matrix of K.
>
> d (the determinant) is calculated normally for K (for the example above, it is 489 = 21 (mod 26)). The inverse, d^{-1}, is found by finding a number such that $d \times d^{-1} = 1\ (\mathrm{mod}\ 26)$ (this is 5 for the example above since 5*21 = 105 = 1 (mod 26)). The simplest way of doing this is to loop through the numbers 1..25 and find the one such that the equation is satisfied. There is no solution (i.e. choose a different key) if $\gcd(d,26) \neq 1$ (this means d and 26 share factors, if this is the case K can not be inverted, this means the key you have chosen will not work, so choose another one).
>
> That is it. Once K^{-1} is found, decryption can be performed.

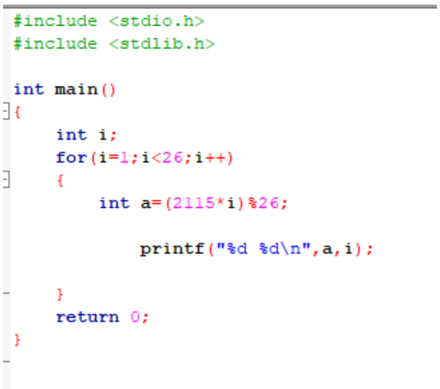
即：找到行列式->行列式的模逆元->这个数乘以伴随矩阵即可；

于是根据：

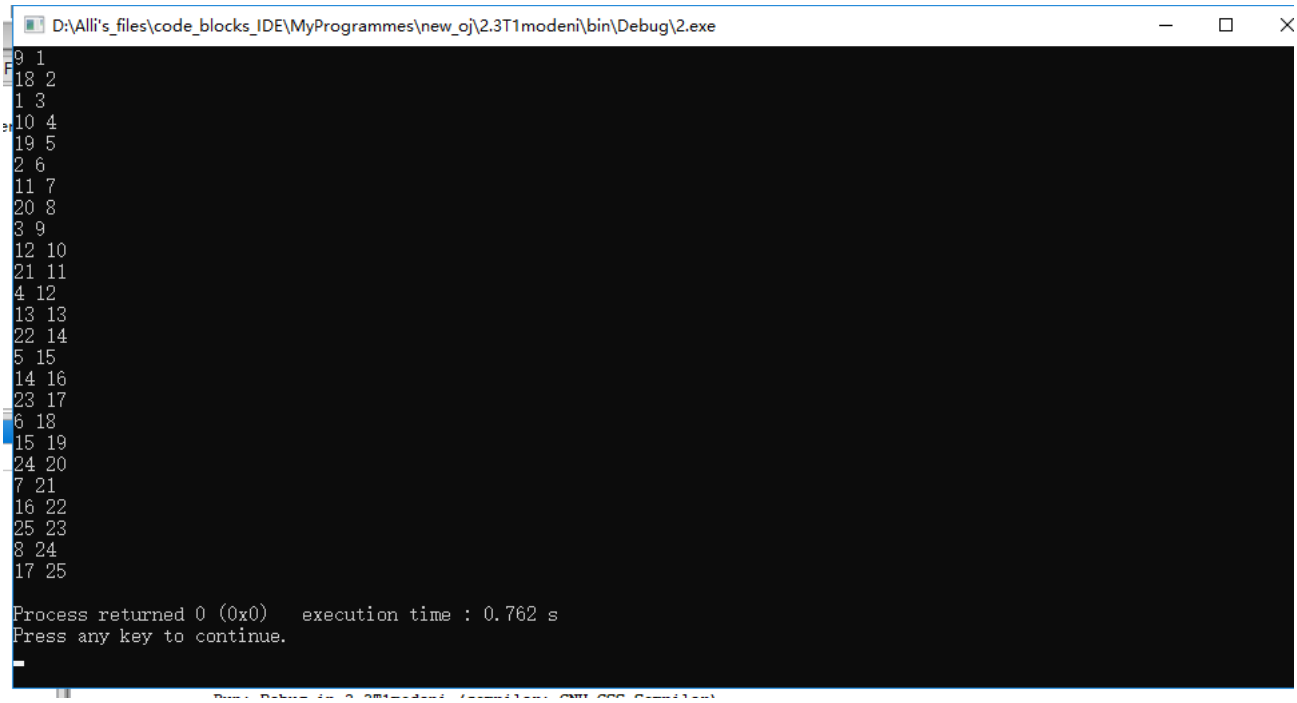


$$\begin{vmatrix} 19 & 7 & 19 \\ 2 & 23 & 2 \\ 18 & 25 & 23 \end{vmatrix} = 2115$$

以及：



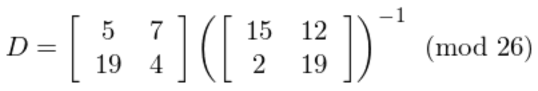
```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int i;
    for(i=1;i<26;i++)
    {
        int a=(2115*i)%26;

            printf("%d %d\n",a,i);

    }
    return 0;
}
```



D:\Alli's_files\code_blocks_IDE\MyProgrammes\new_oj\2.3T1modeni\bin\Debug\2.exe

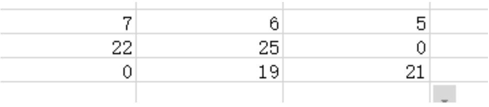
```
9 1
18 2
1 3
10 4
19 5
2 6
11 7
20 8
3 9
12 10
21 11
4 12
13 13
22 14
5 15
14 16
23 17
6 18
15 19
24 20
7 21
16 22
25 23
8 24
17 25

Process returned 0 (0x0)   execution time : 0.762 s
Press any key to continue.
```

找到i=3，乘以伴随矩阵，再像



$$D = \begin{bmatrix} 5 & 7 \\ 19 & 4 \end{bmatrix} \left(\begin{bmatrix} 15 & 12 \\ 2 & 19 \end{bmatrix} \right)^{-1} \pmod{26}$$

一样，mod26，得到模26下的逆矩阵：



7	6	5
22	25	0
0	19	21

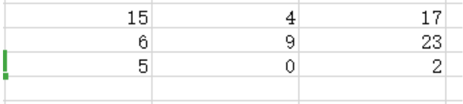
明文*该逆矩阵：

$$\begin{pmatrix} 1 & 24 & 8 \\ 0 & 18 & 11 \\ 1 & 7 & 11 \end{pmatrix} \times \begin{pmatrix} 7 & 6 & 5 \\ 22 & 25 & 0 \\ 0 & 19 & 21 \end{pmatrix} = \begin{pmatrix} 535 & 758 & 173 \\ 396 & 659 & 231 \\ 161 & 390 & 236 \end{pmatrix}$$

再像：

$$\begin{aligned} D &= \begin{bmatrix} 5 & 7 \\ 19 & 4 \end{bmatrix} \left(\begin{bmatrix} 15 & 12 \\ 2 & 19 \end{bmatrix} \right)^{-1} = \begin{bmatrix} 5 & 7 \\ 19 & 4 \end{bmatrix} \begin{bmatrix} 19 & 14 \\ 24 & 15 \end{bmatrix} \\ &= \begin{bmatrix} 263 & 175 \\ 457 & 326 \end{bmatrix} = \begin{bmatrix} 3 & 19 \\ 15 & 14 \end{bmatrix} \pmod{26} \end{aligned}$$

mod26一样后得到：



15	4	17
6	9	23
5	0	2

即：密钥的逆矩阵D，根据公式D*密文=明文，解出：BABYSHILLZCCEDHMQH...显然，没啥意思...，故不是正解

于是重复相同步骤求解BABYSHILL对应后九个字母的情况：发现伴随矩阵的行列式为：324，又根据：

> 当a与f互素时，a关于模f的乘法逆元有解。如果不互素，则无解。如果f为素数，则从1到f-1的任意数都与f互素，即在1到f-1之间都恰好有一个关于模f的乘法逆元。

即：（324，26）不互素也就没有模26的模逆元，所以该情况不成立...

再尝试三个字字母三个字母的划分：以BAB对应HXZ，重复上述冗长的步骤...终于得到了：BABYSHILLCIPHERATTACK......即：flag:hgame{BABYSHILLCIPHERATTACK}

3.Vigener~

百度vigener:

vigener

百度为您找到相关结果约69,700个

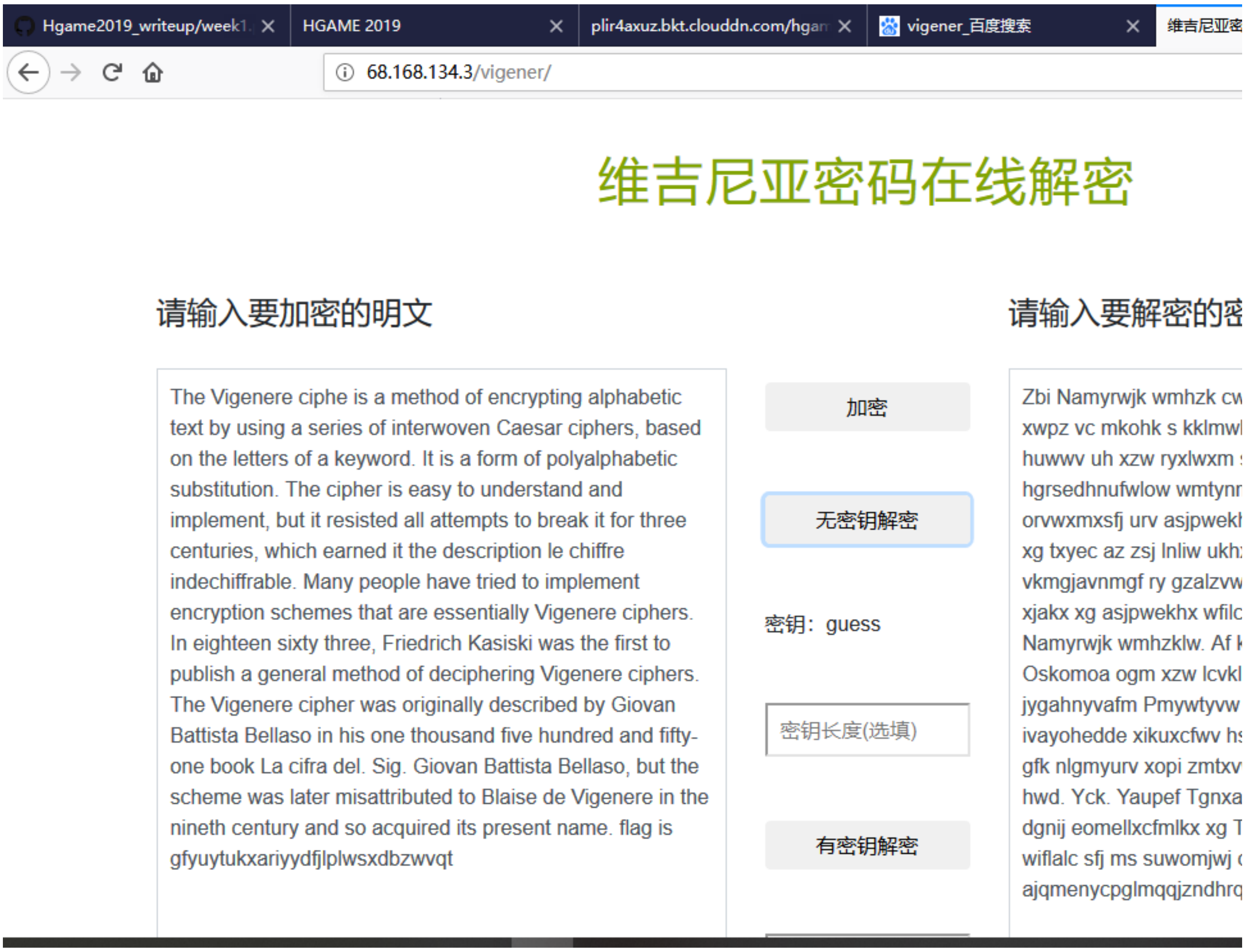
您可以仅查看：英文结果

C语言-数据结构-循环链表实例-维吉尼亚(vigener)密码源... _CSDN博客
2016年7月3日 -
CSDN - 百度快照

vigener viginer in c# code encryption 联合开发网 - pudn.com
2018年12月6日 - 说明: viginer in c# code encryption 文件列表:[举报垃圾]vigener\vigener\App.config, 187 , 2018-11-28 vigener\vigener\bin\Debug\vigener.exe, 6...
www.pudn.com/Download/... - 百度快照

维吉尼亚密码(vigener)在线加密解密 – 孤鸿影的博客
2018年10月8日 - 维吉尼亚密码(vigener)在线加密解密于2018年10月8日由孤鸿影发布在线解密站点 分类: 未分类 发表评论 电子邮件地址不会被公开。 必填项已用*标注 ...
www.zjzhhb.com/archive... - 百度快照

在线解密：



The Vigenere ciphe is a method of encrypting alphabetic text by using a series of interwoven Caesar ciphers, based on the letters of a keyword. It is a form of polyalphabetic substitution. The cipher is easy to understand and implement, but it resisted all attempts to break it for three centuries, which earned it the description le chiffre indechiffrable. Many people have tried to implement encryption schemes that are essentially Vigenere ciphers. In eighteen sixty three, Friedrich Kasiski was the first to publish a general method of deciphering Vigenere ciphers. The Vigenere cipher was originally described by Giovan Battista Bellaso in his one thousand five hundred and fifty-one book La cifra del. Sig. Giovan Battista Bellaso, but the scheme was later misattributed to Blaise de Vigenere in the nineth century and so acquired its present name. flag is gfyuytukxariyydfjlplwsxdbzwvqt

密钥：guess

Zbi Namyrwjk wmhzk cw
xwpz vc mkohk s kklmwl
huwwv uh xzw ryxlwxm
hgrsedhnufwlow wmtynr
orvwxmxsfj urv asjpwekl
xg txyec az zsj lnliw ukh
vkmgjavnmgf ry gzalzvw
xjakx xg asjpwekhx wfilc
Namyrwjk wmhzklw. Af k
Oskomoa ogm xzw lcvkl
jygahnyvafm Pmywtyvw
ivayohedde xikuxcfwv hs
gfk nlgmyurv xopi zmtxv
hwd. Yck. Yaupef Tgnxa
dgnij eomellxcfmlkx xg T
wiflalc sfj ms suwomjwj
ajqmenycpglmqqjzndhrq

找到flag:hgame{gfyuytukxariyydfjlplwsxdbzwvqt}