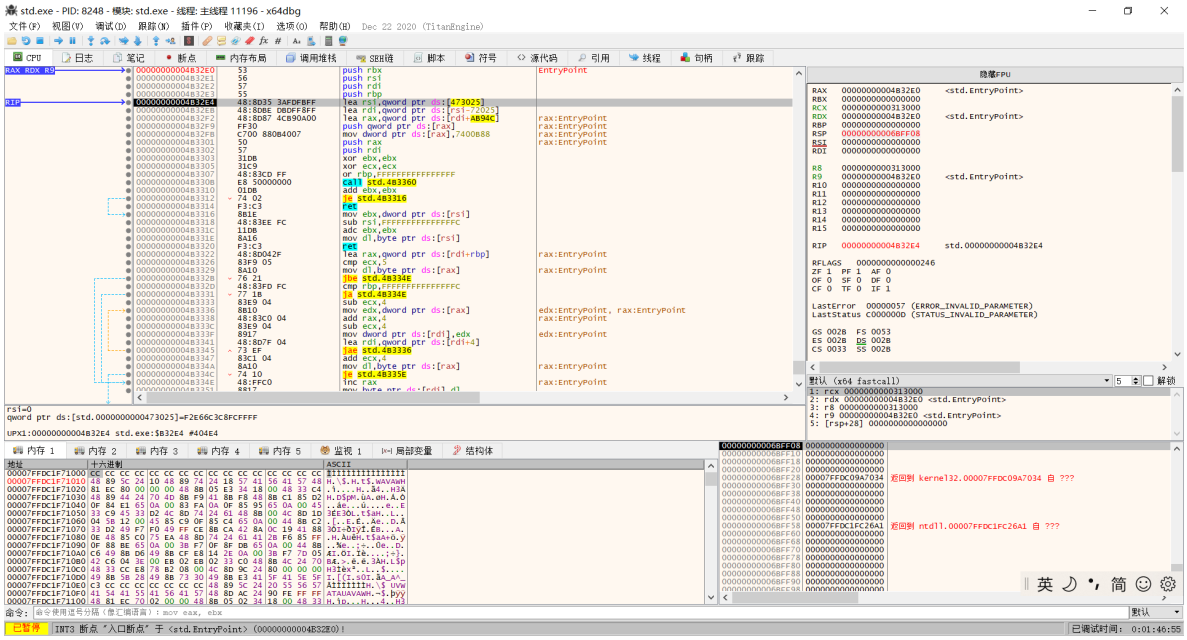
月色真美wp

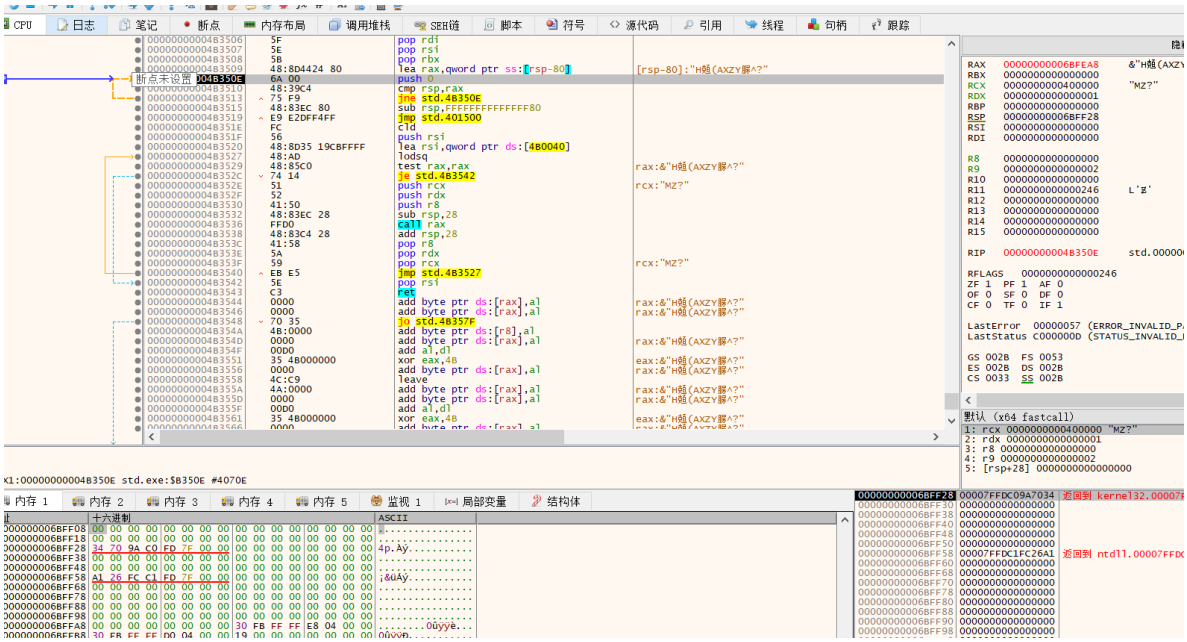
upx的工具脱壳可以网上看，很简单的命令

https://zhuanlan.zhihu.com/p/34263050 , 不等怎么脱壳的都可以看这个博客

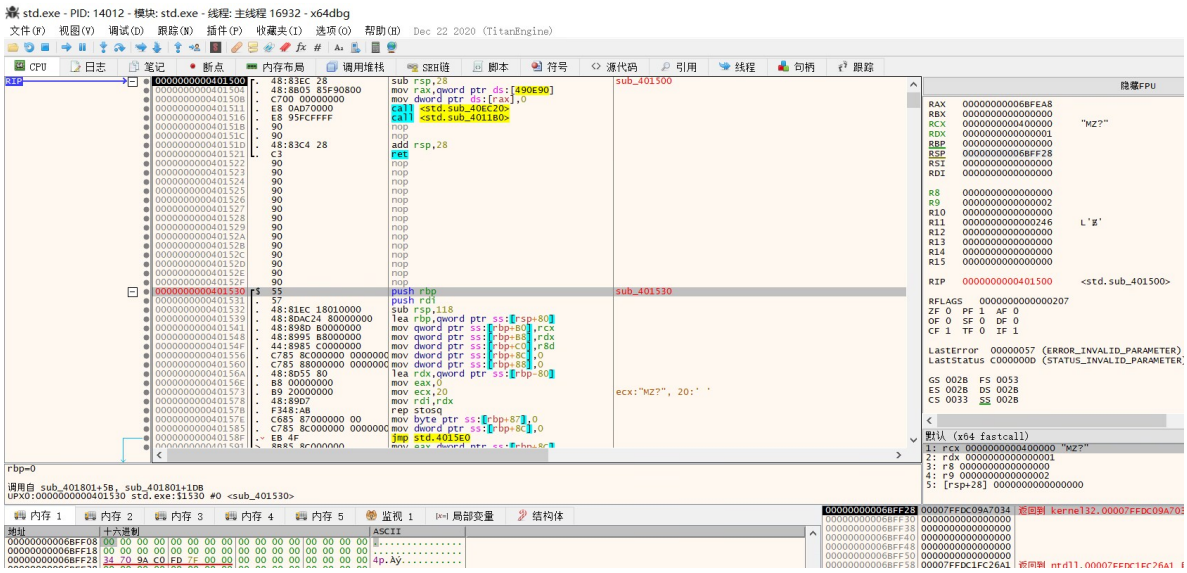
这里演示一下x64dbg脱壳



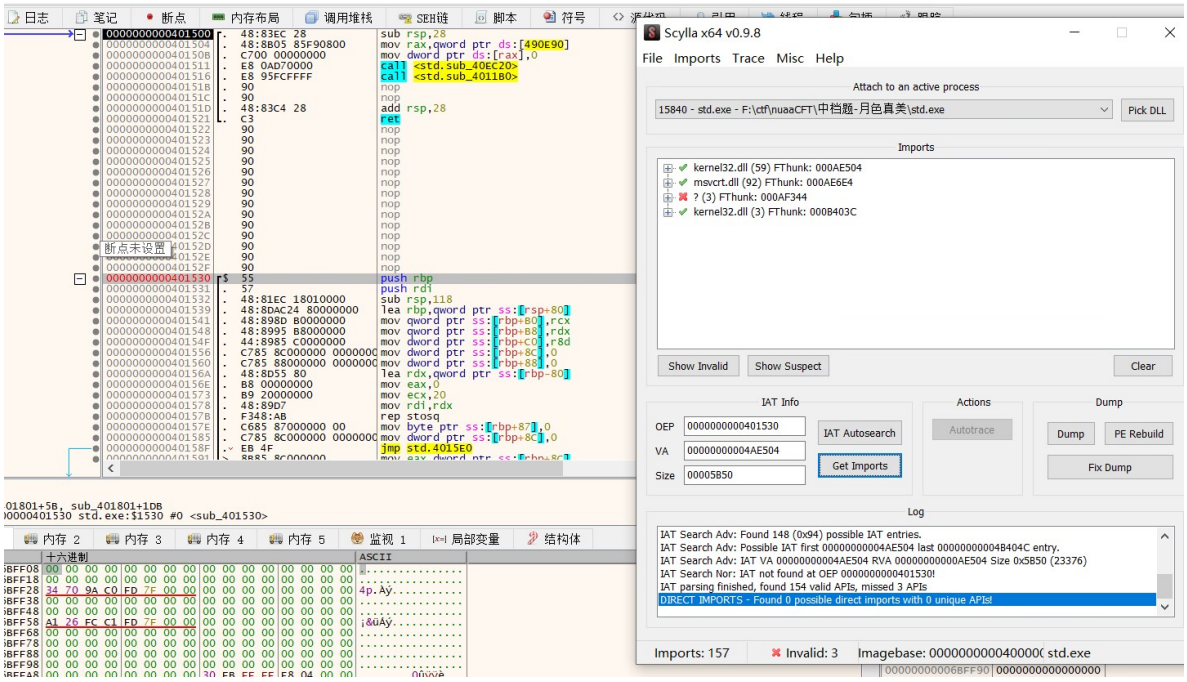
利用esp定律，esp发生变化，下好硬件断点，



然后运行，就会来到这里，接着单步运行



发现401530很像入口点，于是dump

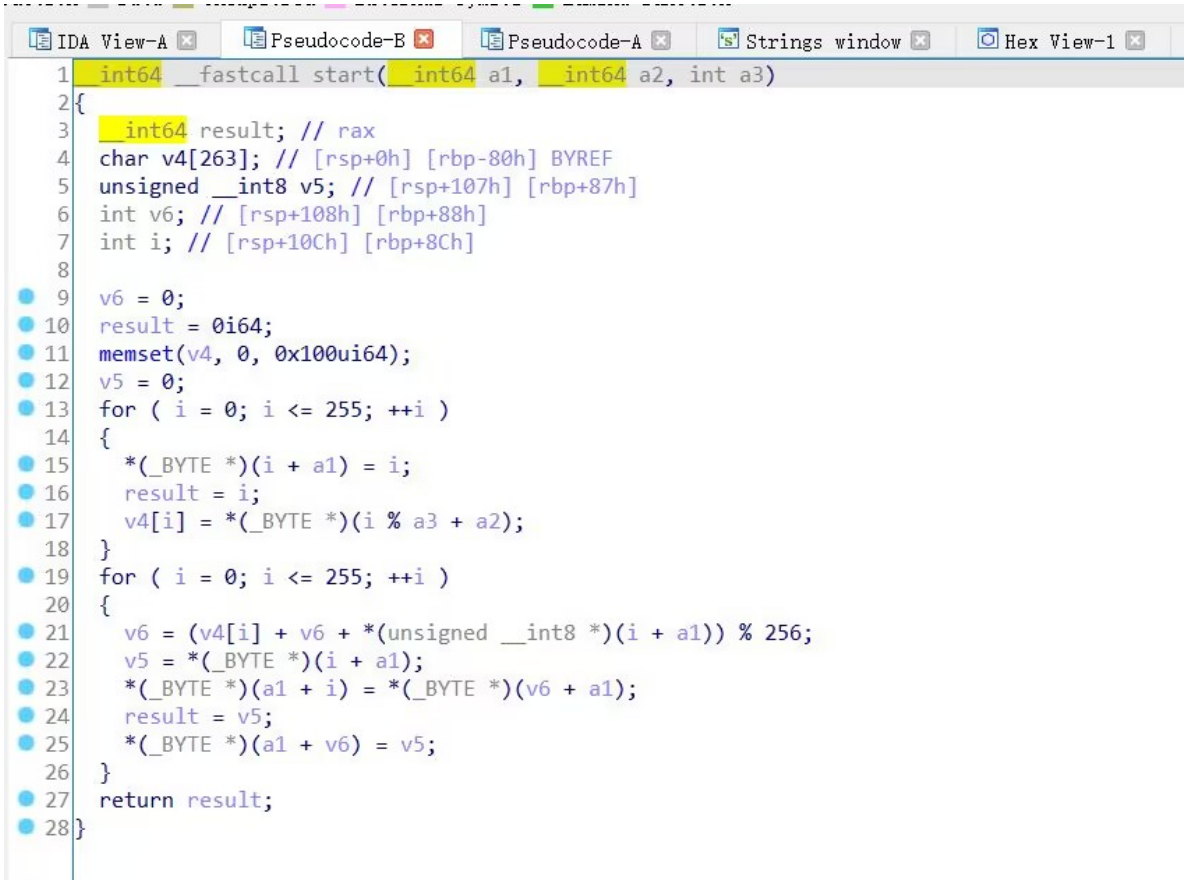


打开dump好的exe文件

```
sub_40EC00();
strcpy(Str, "AttackonTitan");
v0 = j_strlen(Str);
start((__int64)v5, (__int64)Str, v0);
sub_4250A0(aWelcome);
sub_4250A0(aGuessWhatTheTi);
sub_4250A0(aInputYourStr);
sub_425050("%s", ::Str);
v1 = j_strlen(::Str);
sub_4016C9(v5, ::Str, v1);
v6[0] = 71;
v6[1] = -15;
v6[2] = 64;
v6[3] = -33;
v6[4] = -100;
v6[5] = 45;
v6[6] = 108;
v6[7] = -99;
v6[8] = -120;
v6[9] = -53;
v6[10] = 111;
v6[11] = 46;
v6[12] = 68;
v6[13] = -60;
v6[14] = -112;
v6[15] = -55;
v6[16] = -31;
```

```
  v6[15] = -55;
  v6[16] = -31;
  qmemcpy(v7, "`Go", sizeof(v7));
  for ( i = 0; i <= 19; ++i )
  {
    if ( ::Str[i] != v6[i] )
    {
      sub_4250A0(aWrong);
      return 0i64;
    }
  }
  sub_4250A0(aRightHhhh);
  sub_4250A0(byte_48C04E);
  v3 = j_strlen(Str);
  start((__int64)v5, (__int64)Str, v3);
  v4 = j_strlen(::Str);
  sub_4016C9(v5, ::Str, v4);
  sub_4250A0("NUAACTF{%s}", ::Str);
  return 0i64;
}
```

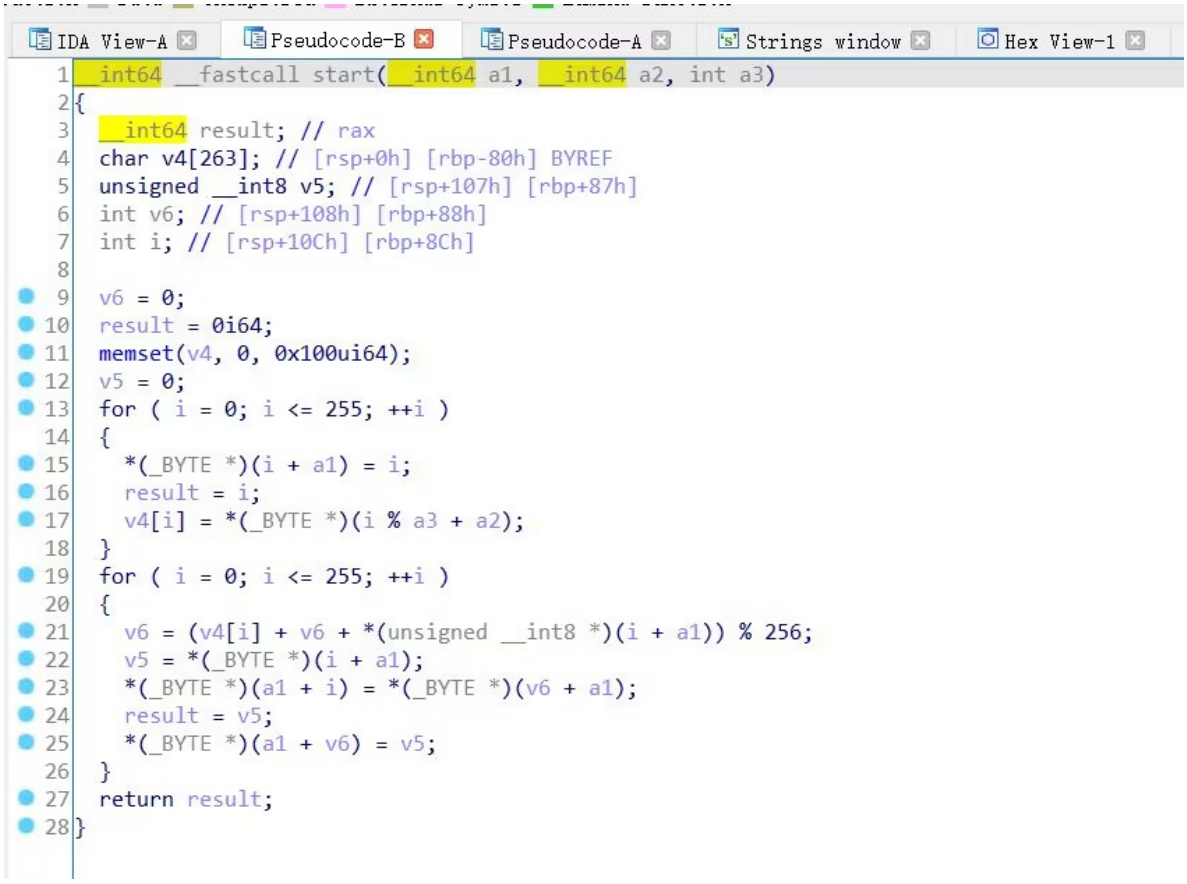
看到全部的代码，发现str先被加密了一次，到输出答案的时候也被加密了一次，怀疑加密等于解密




```
__int64 __fastcall start(__int64 a1, __int64 a2, int a3)
{
  __int64 result; // rax
  char v4[263]; // [rsp+0h] [rbp-80h] BYREF
  unsigned __int8 v5; // [rsp+107h] [rbp+87h]
  int v6; // [rsp+108h] [rbp+88h]
  int i; // [rsp+10Ch] [rbp+8Ch]

  v6 = 0;
  result = 0i64;
  memset(v4, 0, 0x100ui64);
  v5 = 0;
  for ( i = 0; i <= 255; ++i )
  {
    *(_BYTE *)(i + a1) = i;
    result = i;
    v4[i] = *(_BYTE *)(i % a3 + a2);
  }
  for ( i = 0; i <= 255; ++i )
  {
    v6 = (v4[i] + v6 + *(unsigned __int8 *)(i + a1)) % 256;
    v5 = *(_BYTE *)(i + a1);
    *(_BYTE *)(a1 + i) = *(_BYTE *)(v6 + a1);
    result = v5;
    *(_BYTE *)(a1 + v6) = v5;
  }
  return result;
}
```

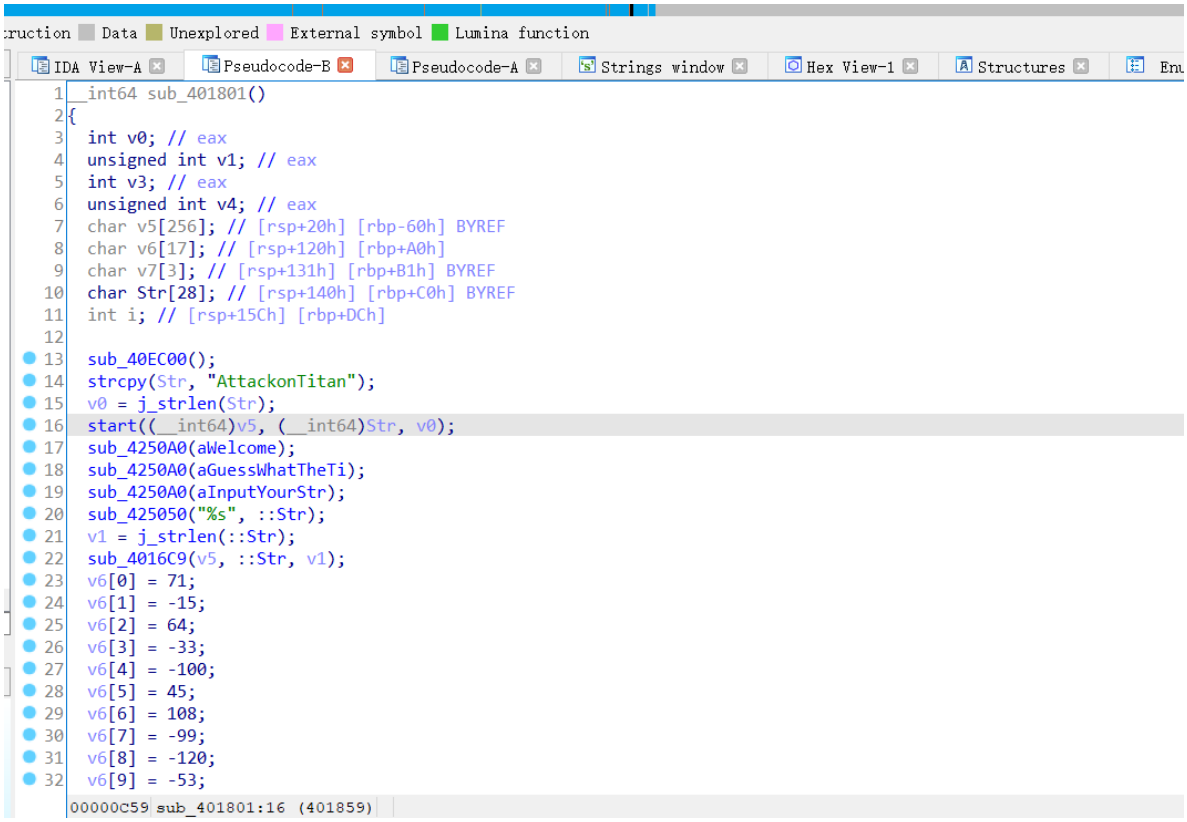
先看到加密函数，256轮加密s盒，




```
__int64 __fastcall start(__int64 a1, __int64 a2, int a3)
{
  __int64 result; // rax
  char v4[263]; // [rsp+0h] [rbp-80h] BYREF
  unsigned __int8 v5; // [rsp+107h] [rbp+87h]
  int v6; // [rsp+108h] [rbp+88h]
  int i; // [rsp+10Ch] [rbp+8Ch]

  v6 = 0;
  result = 0i64;
  memset(v4, 0, 0x100ui64);
  v5 = 0;
  for ( i = 0; i <= 255; ++i )
  {
    *(_BYTE *)(i + a1) = i;
    result = i;
    v4[i] = *(_BYTE *)(i % a3 + a2);
  }
  for ( i = 0; i <= 255; ++i )
  {
    v6 = (v4[i] + v6 + *(unsigned __int8 *)(i + a1)) % 256;
    v5 = *(_BYTE *)(i + a1);
    *(_BYTE *)(a1 + i) = *(_BYTE *)(v6 + a1);
    result = v5;
    *(_BYTE *)(a1 + v6) = v5;
  }
  return result;
}
```

很明显的rc4特征的加密



找到密钥，在14行，v6是加密后得到的密文，rc4的具体说明可以看ctfwiki的reverse中的常见加密算法，根据rc4的特性，我们只需要预处理s盒，然后将v6重新丢到rc4加密函数中即可，这部分就是根据ida然后自己扣下来把代码写到合理能运行就好了

```
#include<bits/stdc++.h>
using namespace std;
char input[21];
void init(unsigned char *s,unsigned char *key,int len){
    int i = 0,j = 0;
    char k[256] = {0};
    unsigned char tmp = 0;
    for (i=0;i < 256;i++) {
        s[i] = i;
        k[i] = key[i%len];
    }
    for (i=0;i < 256;i++) {
        j = (j + s[i] + k[i]) % 256;
        tmp = s[i];
        s[i] = s[j];
        s[j] = tmp;
    }
}

void crypt(unsigned char *s,unsigned char *str,int len){
    int i=0,j=0,t=0;
    unsigned long k=0;
    unsigned char tmp;
    for(k=0;k<len;k++){
        i=(i+1)%256;
        j=(j+s[i])%256;
        tmp=s[i];
        s[i]=s[j];
        s[j]=tmp;
        t=(s[i]+s[j])%256;
```

```
            str[k]^=s[t];
        }
    }

    int main(){
        unsigned char s[256];
        char key[]="AttackonTitan";
        init(s,(unsigned char *)key,strlen(key));
        char s2[]=
    {71,-15,64,-33,-100,45,108,-99,-120,-53,111,46,68,-60,-112,-55,-31,96,71,111};
        crypt(s,(unsigned char *)s2,20);
        printf("NUAACTF{%s}",s2);
        return 0;
    }
```

运行出来NUAACTF{0704Aremy3weeTe3T0ne}

ps:后面的为彩蛋

于是小丫姐姐的flag就是

NUAACTF{gift_F0R_yOu4O5@hyq2}

双方都把对方的生日加到了flag里面，很可惜没有人做出她的题，只有人写了我的狗粮题，表示很伤心没有人发现这个双向的彩蛋。

希望还有后来人