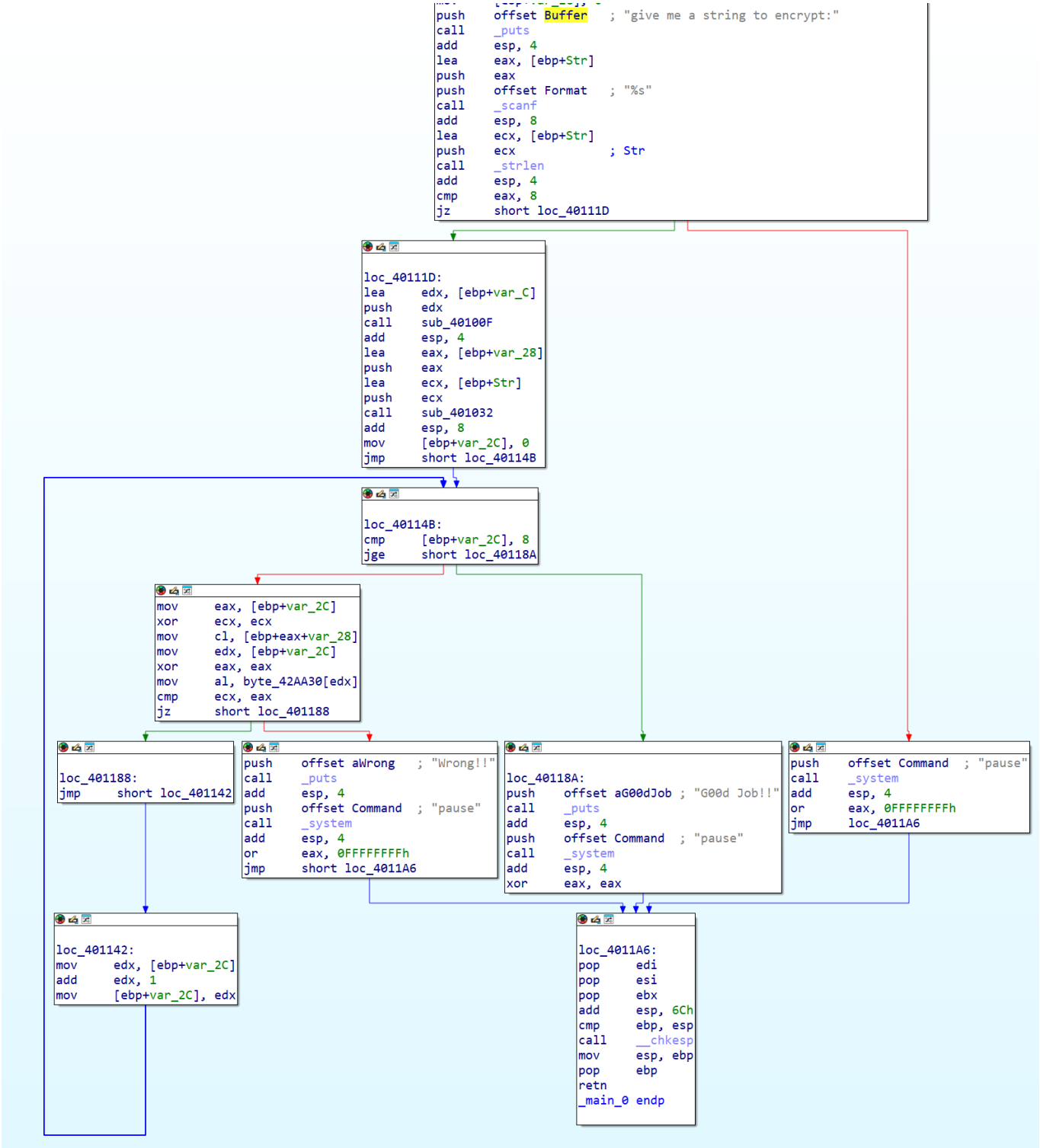
DES算法逆向分析

先直接运行程序，不难发现当输入长度为8个字符时程序会进行判定

使用 IDA 静态分析该程序

main 函数如图所示



可知var_C被传入了函数sub_40100F，推测其为密钥编排函数，DE3_En1C为密钥

输入的str和var_28被传入了函数sub_401032，推测其为加密函数

var_2C为 main 函数中的一个循环的控制变量 i

具体分析该循环

可知其功能是将var_28与byte_42AA30的各位进行比较，若全部相同才能通过

推测byte_42AA30为目标密文，双击查看其值

```
.data:0042AA30 byte_42AA30      db 0EFh
.data:0042AA31                  db  34h ; 4
.data:0042AA32                  db 0D4h
.data:0042AA33                  db 0A3h
.data:0042AA34                  db 0C6h
.data:0042AA35                  db  84h
.data:0042AA36                  db 0E4h
.data:0042AA37                  db  23h ; #
```

现进入具体的加密函数进行分析

进入sub_401032

IDA View-A | Pseudocode-A

```
1  int __cdecl sub_401200(int a1, int a2)
2  {
3    int v3[8]; // [esp+4Ch] [ebp-A4h] BYREF
4    _BYTE v4[32]; // [esp+6Ch] [ebp-84h] BYREF
5    _BYTE v5[32]; // [esp+8Ch] [ebp-64h] BYREF
6    _BYTE Src[32]; // [esp+ACh] [ebp-44h] BYREF
7    _BYTE v7[32]; // [esp+CCh] [ebp-24h] BYREF
8    int i; // [esp+ECh] [ebp-4h]
9
10   sub_40101E(a1, Src, 8);
11   sub_401028(Src, (int)Src);
12   memcpy(v4, Src, sizeof(v4));
13   memcpy(v3, v7, sizeof(v3));
14   for ( i = 0; i < 15; ++i )
15   {
16     sub_401005((int)v3, v5, (int)&unk_42DC9C + 48 * i);
17     sub_401023(v5, v4, 32);
18     memcpy(v4, v3, sizeof(v4));
19     memcpy(v3, v5, sizeof(v3));
20   }
21   sub_401005((int)v3, v5, (int)&unk_42DC9C + 48 * i);
22   sub_401023(v4, v5, 32);
23   memcpy(Src, v4, sizeof(Src));
24   memcpy(v7, v3, sizeof(v7));
25   sub_401019(Src, (int)Src);
26   return sub_40102D(Src, a2, 8);
27 }
```

可知加密一共有16轮

多次用到memcpy函数，v4和v3疑似是用于保存加密中间结果的数组

推测unk_42DC9C为编排后的子密钥

sub_401005疑似加密轮中的函数

进入sub_401005

IDA View-A | Pseudocode

```
1  void *__cdecl sub_4019B0(void *a1, void *a2, _BYTE *a3)
2  {
3    _BYTE Src[32]; // [esp+4Ch] [ebp-50h] BYREF
4    _BYTE v5[48]; // [esp+6Ch] [ebp-30h] BYREF
5
6    sub_40104B(a1, (int)v5);
7    sub_401023(v5, a3, 48);
8    sub_40103C(v5, Src);
9    sub_401037(Src, (int)Src);
10   return memcpy(a2, Src, 0x20u);
11 }
```

进入sub_40104B

IDA View-A | Pseudocode-A

```
1  unsigned __int8 __cdecl sub_401690(void *Src, int a2)
2  {
3    unsigned __int8 result; // al
4    int i; // [esp+4Ch] [ebp-34h]
5    _BYTE v4[48]; // [esp+50h] [ebp-30h] BYREF
6
7    result = (unsigned __int8)memcpy(v4, Src, sizeof(v4));
8    for ( i = 0; i < 48; ++i )
9    {
10     result = v4[byte_42809C[i] - 1];
11     *(_BYTE *)(i + a2) = result;
12   }
13   return result;
14 }
```

查看byte_42809C

```
.rdata:0042809C byte_42809C     db 32
.rdata:0042809D                 db  1
.rdata:0042809E                 db  2
.rdata:0042809F                 db  3
.rdata:004280A0                 db  4
.rdata:004280A1                 db  5
.rdata:004280A2                 db  4
.rdata:004280A3                 db  5
.rdata:004280A4                 db  6
.rdata:004280A5                 db  7
.rdata:004280A6                 db  8
.rdata:004280A7                 db  9
```

```
.rdata:004280A8                 db    8
.rdata:004280A9                 db    9
.rdata:004280AA                 db  0Ah
.rdata:004280AB                 db  0Bh
.rdata:004280AC                 db  0Ch
.rdata:004280AD                 db  0Dh
.rdata:004280AE                 db  0Ch
.rdata:004280AF                 db  0Dh
.rdata:004280B0                 db  0Eh
.rdata:004280B1                 db  0Fh
.rdata:004280B2                 db  10h
.rdata:004280B3                 db  11h
.rdata:004280B4                 db  10h
.rdata:004280B5                 db  11h
.rdata:004280B6                 db  12h
.rdata:004280B7                 db  13h
.rdata:004280B8                 db  14h
.rdata:004280B9                 db  15h
.rdata:004280BA                 db  14h
.rdata:004280BB                 db  15h
.rdata:004280BC                 db  16h
.rdata:004280BD                 db  17h
.rdata:004280BE                 db  18h
.rdata:004280BF                 db  19h
.rdata:004280C0                 db  18h
.rdata:004280C1                 db  19h
.rdata:004280C2                 db  1Ah
.rdata:004280C3                 db  1Bh
.rdata:004280C4                 db  1Ch
.rdata:004280C5                 db  1Dh
.rdata:004280C6                 db  1Ch
.rdata:004280C7                 db  1Dh
.rdata:004280C8                 db  1Eh
.rdata:004280C9                 db  1Fh
.rdata:004280CA                 db  20h
.rdata:004280CB                 db    1
```

发现 DES 的 E 盒

故将`var_C`和`byte_42AA30`输入 DES 解密程序，得到flag

DES 加密/解密

运算模式: CBC (密码块链)　填充模式: None　密钥长度: 64 bits

密钥: Text　DE3_En1C

偏移: Text　null or 64 bits

EF34D4A3C684E423

字符编码: UTF-8　格式: Hex　(格式加密表示输出，解密表示输入)　加密　解密　↕交换

HarDd3s?

将 flag 输入程序