一个base64自定义字符表加密

然后这个函数

动调发现base64函数后的打印字符串隐藏着一个函数

```
; Attributes: bp-based frame

sub_400747 proc near
; __unwind {
push    rbp
mov     rbp, rsp
mov     edi, offset aTheEncryptionI ; "The encryption is done!"
call    _puts
mov     rsp, offset off_603110
retn
sub_400747 endp
```

Instruction Data Unexplored External symbol

IDA View-A | Pseudocode-A | Hex View-1 | Structures

```
1  size_t sub_400686()
2  {
3    size_t result; // rax
4    int i; // [rsp+Ch] [rbp-14h]
5
6    for ( i = 0; ; ++i )
7    {
8      result = strlen(s);
9      if ( i >= result )
10       break;
11     if ( s[i] > 96 && s[i] <= 121 || s[i] > 64 && s[i] <= 89 )
12     {
13       s[i] ^= 0x20u;
14       ++s[i];
15     }
16   }
17   return result;
18 }
```

之后是一个crc8的加密，之后与存在的数组比较

解题脚本

```
#include <stdio.h>
#include <string.h>

void crc8decode(char * str)
{
    int i,j;
    unsigned char dst;
    for (i = 0; i < strlen(str); ++i)
    {
        dst = str[i];
        for (j = 0; j < 8; ++j)
        {
            if (!(dst & 1))
                dst /= 2;
            else
                dst = ((dst ^ 0x31u) >> 1) + 0x80u;
        }
        str[i] = dst;
    }

}
char data[40] = {
    0x4E, 0x1C, 0x6E, 0xF7, 0x4E, 0x96, 0xF6, 0xE6, 0x9A, 0xD7, 0x4D, 0x30, 0x9A, 0xD7, 0x8A, 0x51,
    0x41, 0xA7, 0x01, 0x6E, 0x33, 0xF4, 0x4D, 0xD7, 0x2C, 0x1C, 0xAB, 0xF9, 0xD9, 0xE8, 0x9B, 0x96,
    0x1D, 0xD9, 0x4D, 0xA5, 0x51, 0x5E, 0x89, 0x89
};

void change(char*eninput)
{
    int i;
    for (i = 0; i < strlen(eninput); ++i)
    {
        if (eninput[i] >= 'a' && eninput[i] <= 'z')
        {
            eninput[i] = ((eninput[i] ^ 32) - 'A' - 5+260) % 26 + 'A';
        }
        else
        {
```

```
            if (eninput[i] >= 'A' && eninput[i] <= 'Z')
            {
                eninput[i] = ((eninput[i] ^ 32) - 'a' - 5+260) % 26 + 'a';
            }
            else
                eninput[i] = eninput[i];
        }
    }
}
int main(void)
{
    crc8decode(data);   //crc8解密
    puts(data);
    change(data);       //隐藏变换解密
    puts(data);
    return 0;
}

/*
输出

eRCme3YJrK95rKakx24Ci19KgRsDbcF3fb9Zkv==
ZmxHZ3teMf95MfVFS24xD19fBmNyWXa3AW9uFQ==

*/

```

base64解密

```
import base64
enstr = "ZmxHZ3teMf95MfVFS24xD19fBmNyWXa3AW9uFQ=="
biao2 =
str.maketrans("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/","ab
cdefghijklMNOPQRSTUVWXYZABCDEFGHIJKLmnopqrstuvwxyz0123456789+/")
str_decode = base64.b64decode(enstr.translate(biao2).encode('utf-8'))
print(str(str_decode,'utf-8'))

#输出
# flag{D0_y0U_Kn1w_EncrYp7ion}
```