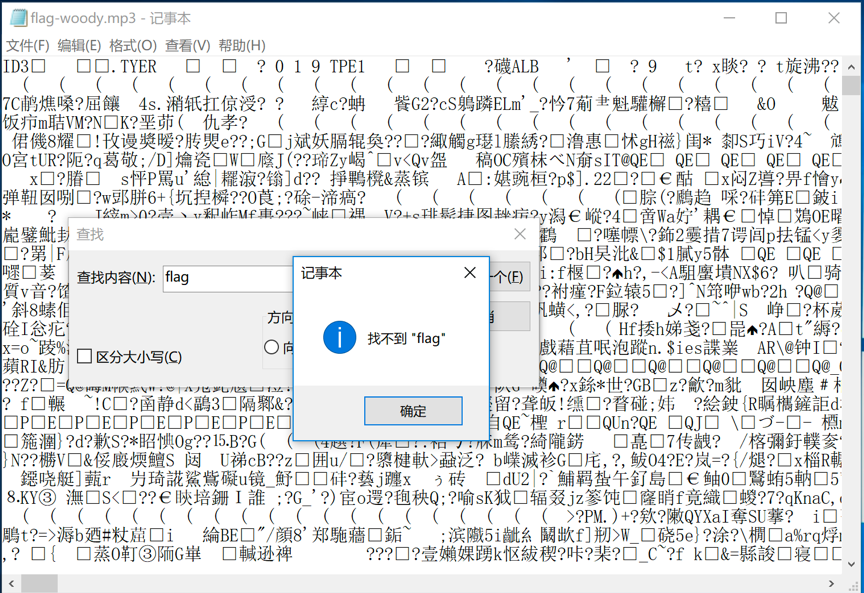


**0x02 曲折**

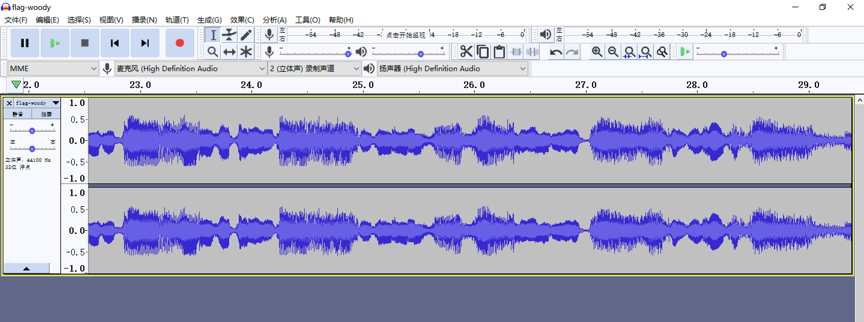
题目很明显，就是一道 MP3的隐写题，按照常规思路，有三种思路：

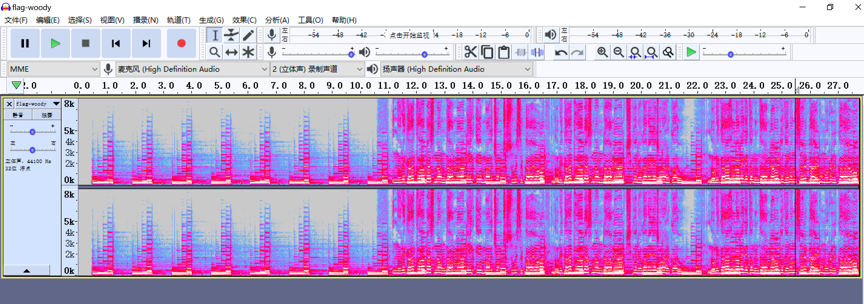
* 直接右击txt打开，查找关键词 flag
* 查看该音频文件的波形图、频谱图，是否存在相关信息可以转化为摩斯电码
* 查看mp3 中是否含有隐藏文件，提取文件

第一种思路没报多大打算，果然没有：

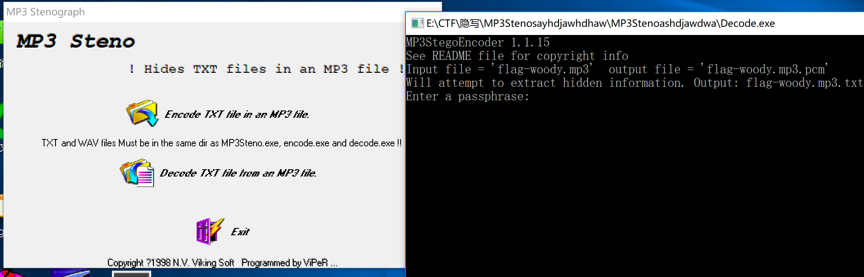


第二个思路：

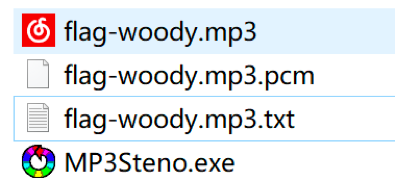




第三种思路：使用MP3Steno提取 mp3 中可能存在的 TXT 文件



猜解出密码为icsc，提取成功





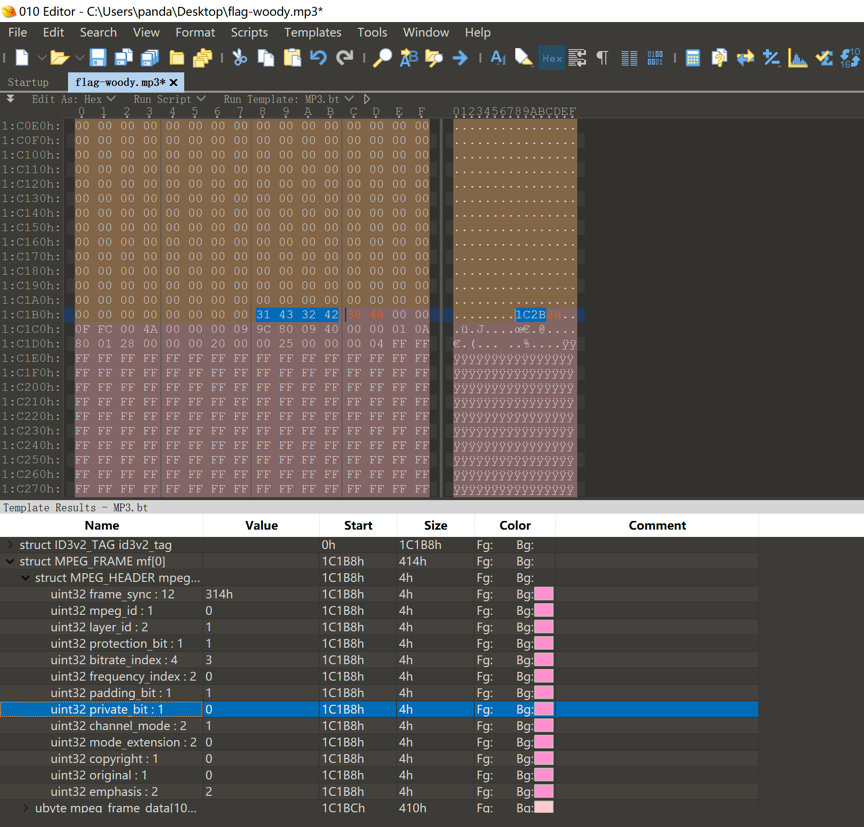
想了半天没有结果，于是换思路。

# 0x03 题解

赛后我问了一些师傅，最终知道了这题的解决方法。

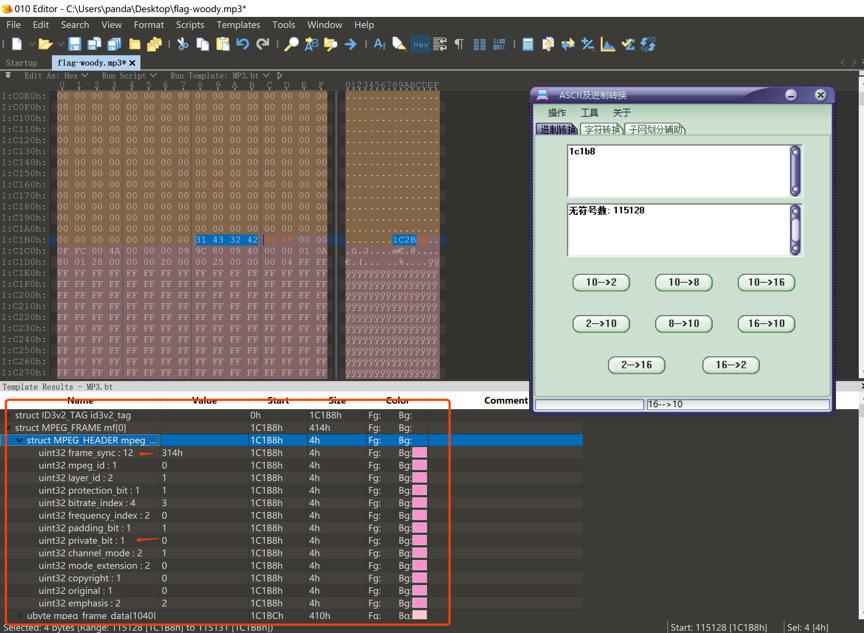
首先回到题目里去，题目提示，通过某种 private 的方式传递信息

而当你用 010editor 打开该 Mp3 文件，并按照提示安装插件后，发现：



存在一个private bit  
因此，只需要提取每一个 mf组中的该字节，组合起来，就是答案。

可以从图中看到 ms 开始位为1 C1B8H，即第 115128 字节



uint32 frame\_sync : 12

uint32 mpeg\_id : 1

uint32 layer\_id : 2

uint32 protection\_bit : 1

uint32 bitrate\_index : 4

uint32 frequency\_index : 2

uint32 padding\_bit : 1

uint32 private\_bit : 1

uint32 channel\_mode : 2

uint32 mode\_extension : 2

uint32 copyright : 1

uint32 original : 1

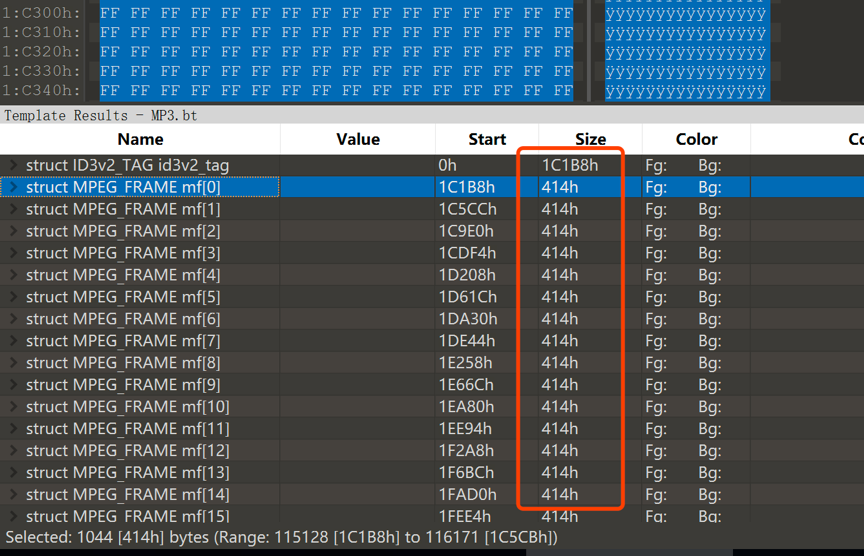
uint32 emphasis : 2

12+1+2+1+4+2+1+1+2+2+1+1+2=32

即 总共 4 字节，private\_bit 为24，所在的字节为第 3 个字节

因此要从前一个，即第二个字节开始提取内容，该字节对应的地址为 115130

观察每一个 mf组



大小都为414h，即1044字节

因此可以得到以下脚本：

# coding:utf-8

import re

import binascii

n = 115130

result = ''

fina = ''

file = open('flag-woody.mp3','rb')

while n < 2222222 :

file.seek(n,0)

n += 1044

file\_read\_result = file.read(1)

read\_content = bin(ord(file\_read\_result))[-1]

result = result + read\_content

textArr = re.findall('.{'+str(8)+'}', result)

textArr.append(result[(len(textArr)\*8):])

for i in textArr:

fina = fina + hex(int(i,2))[2:].strip('\n')

fina = fina.decode('hex')

print fina

