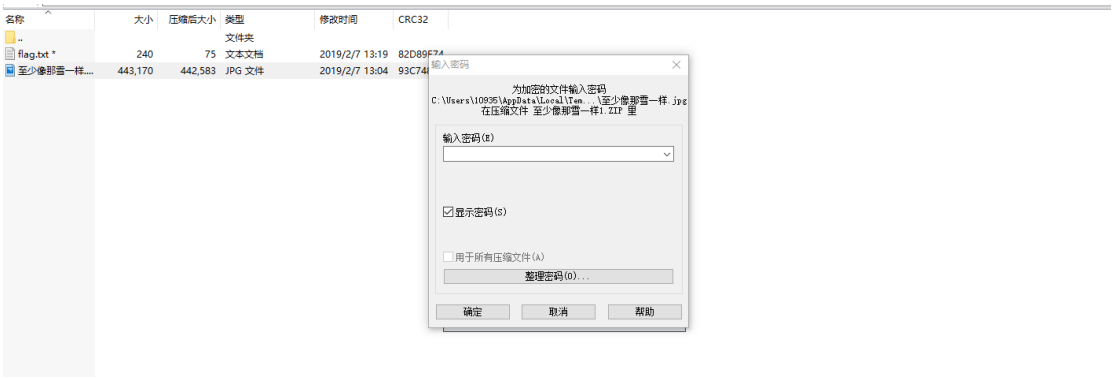
[Adobie] HGAME 2019 week-3 writeup

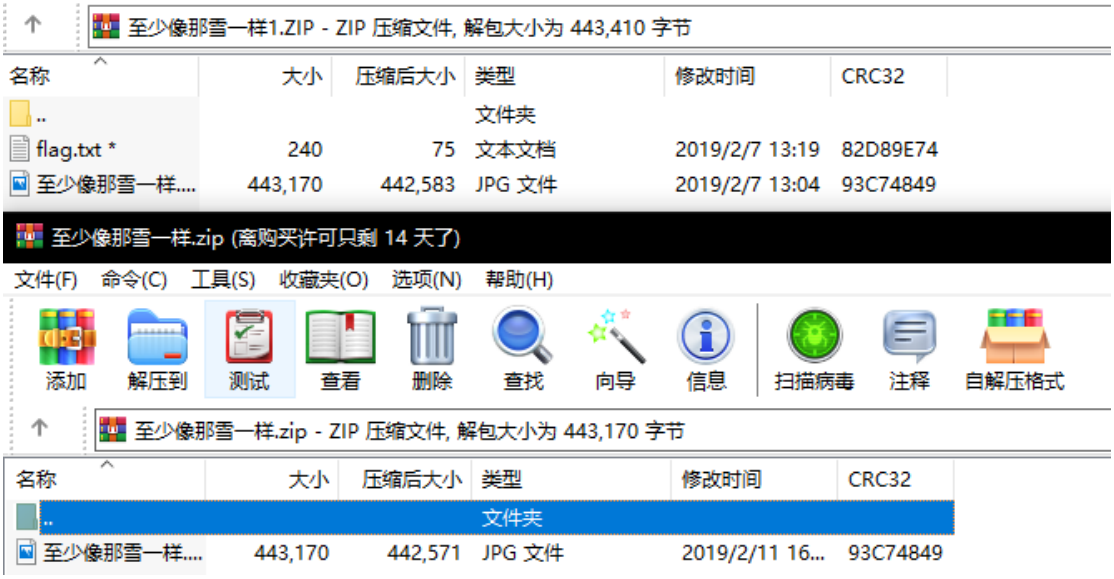
Misc

1. 至少像那雪一样

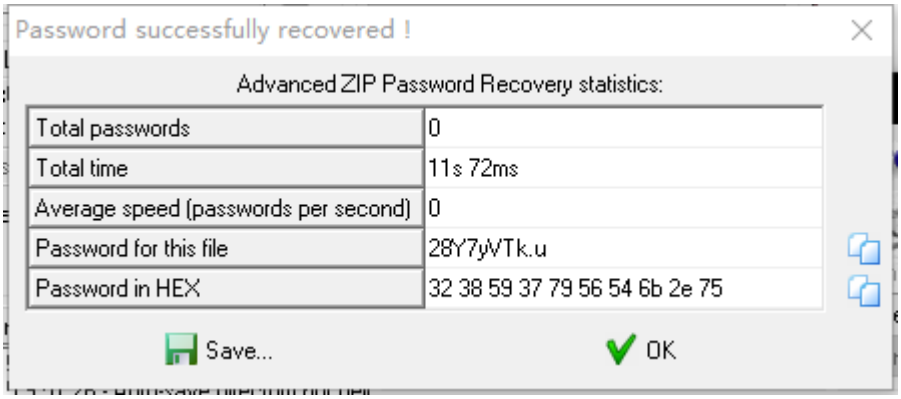
下载文件，是一张后缀为.jpg.的图片。拖入 Hxd 查看 16 进制编码，发现尾部不是 jpg 文件应有的 FF D9，所以还有其他文件，编辑 16 进制编码，手动分离出两个文件，一张 jpg 图片，一个压缩包。先打开压缩包，发现有一张图和一个 txt 文件，但需要密码。



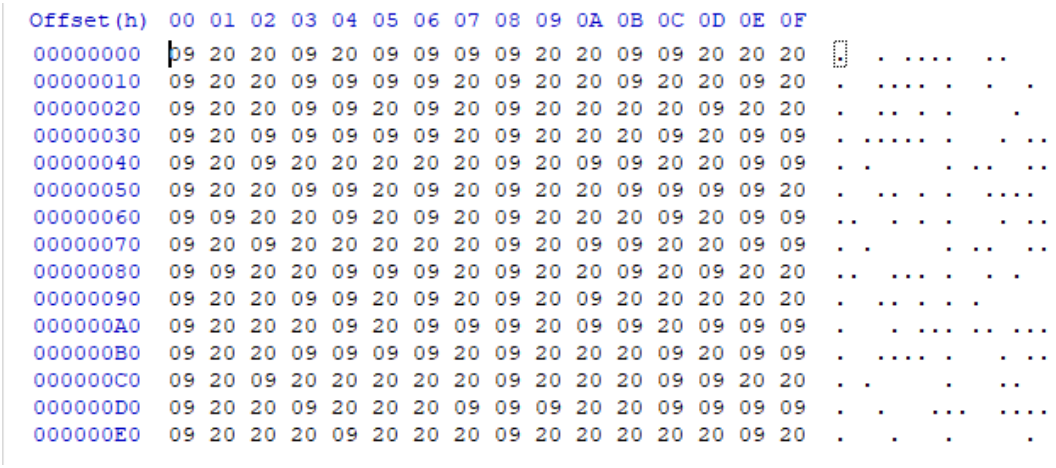
很容易看见，压缩包中的图片名称和分离出的图片名称相同，想到了压缩包破解中常见的明文攻击，于是尝试压缩分离出的图片，发现两者 CRC32 校验码相同，开始使用明文攻击。



破解得到密码，



打开压缩包，点开 flag.txt 竟然是一片空白。沉思良久，看一看它的 16 进制编码。



```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   09 20 20 09 20 09 09 09 09 20 20 09 09 20 20 20   . . ....  ..
00000010   09 20 20 09 09 09 09 20 09 20 20 09 20 20 09 20   .  .... .  .  .
00000020   09 20 20 09 09 20 09 20 09 20 20 20 20 09 20 20   .  .. . .    .
00000030   09 20 09 09 09 09 09 20 09 20 20 20 09 20 09 09   . ..... .   . ..
00000040   09 20 09 20 20 20 20 20 09 20 09 09 20 20 09 09   . .     . ..  ..
00000050   09 20 20 09 09 20 09 20 09 20 20 09 09 09 09 20   .  .. . .  ....
00000060   09 09 20 20 09 20 09 20 09 20 20 20 09 20 09 09   ..  . . .   . ..
00000070   09 20 09 20 20 20 20 20 09 20 09 09 20 20 09 09   . .     . ..  ..
00000080   09 09 20 20 09 09 09 20 09 20 20 09 20 09 20 20   ..  ... .  . .
00000090   09 20 20 09 09 20 09 20 09 20 09 20 20 20 20 20   .  .. . . .
000000A0   09 20 20 20 09 20 09 09 09 20 09 09 20 09 09 09   .   . ... .. ...
000000B0   09 20 20 09 09 09 09 20 09 20 20 20 09 20 09 09   .  .... .   . ..
000000C0   09 20 09 20 20 20 20 20 09 20 20 20 09 09 20 20   . .     .   ..
000000D0   09 20 20 09 20 20 20 09 09 09 20 20 09 09 09 09   .  .   ...  ....
000000E0   09 20 20 20 09 20 20 20 09 20 20 20 20 20 09 20   .   .   .     .
```

右侧的小黑点很神秘，一定是破解的关键。开了很久脑洞，我尝试让有黑点的地方代表 1，无黑点的代表 0，打算使用二进制转字符串得到 flag，但失败了。于是再进行尝试，让有黑点的地方代表 0，无黑点的地方代表 1，解码得到 flag。

2. 旧时记忆

看着图片一脸懵逼，等了好几天出了 hint，而且还是第二个 hint...我才找到了方向。搜索存储器相关资料，看到一种叫打孔卡的东西很相似。进而搜索打孔卡的工作原理。

另外，一些投票机也运用打孔卡。

1928年，IBM发明的80列、矩形孔卡片，成为事实上的标准。
其工作原理如下：编号为0至9，总计10行；以及一块区域，用于第11、第12行（注意，没有编号为第10的行）。
每列的穿孔组合用于表示单个字符：

- 数字通过在行0至行9直接打1个孔来表示。
- 空格符的表示，不需要打孔。
- 字母用2个孔表示：一个孔在第11、第12、第0行；另一个孔在第1至第9行。字母表被依次分为由9个字母组成的区（"zones"），每个区的字母依次在第1至第9行打孔。每个区分别在第11、第12、第0行打孔。第3区第1个字符保留未使用。
- 一些特殊字符使用了额外的单孔表示，或者双孔表示。
- 大多数特殊字符（如标点符号等）用3孔表示：第8行被穿孔；第0、第11、第12行有1个穿孔；第1到第7行有1个穿孔。第9行保留未使用。

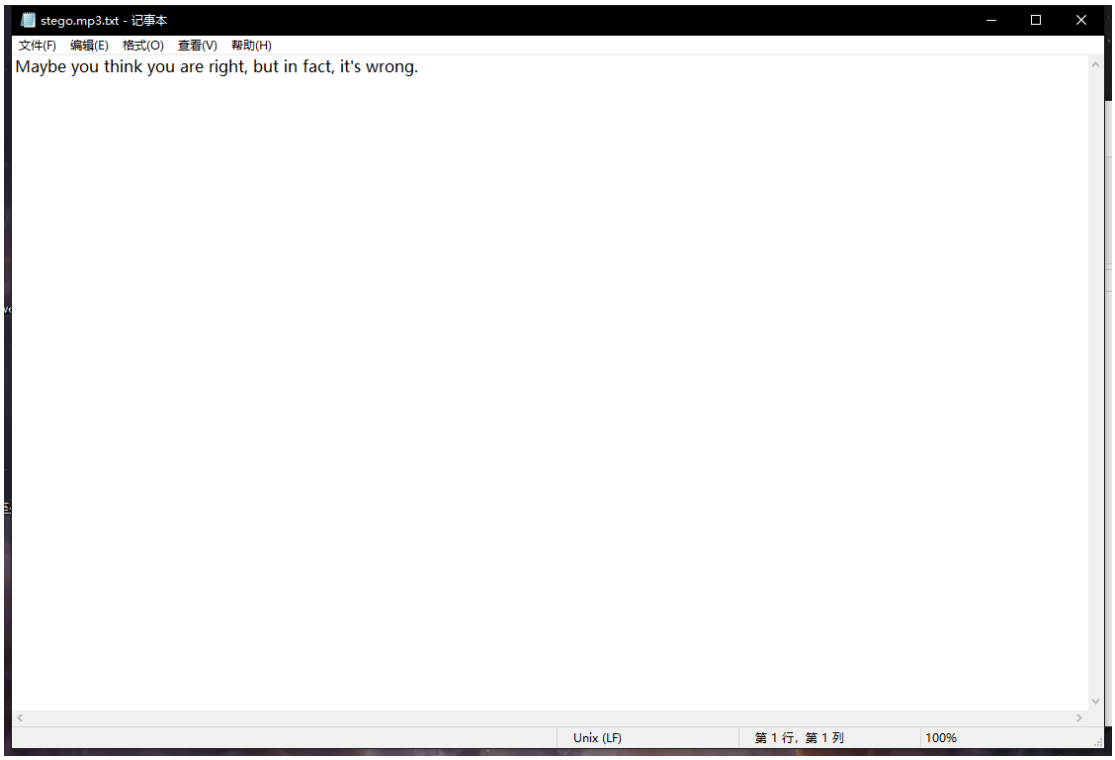
总计表示了67个字符。

进入到数字计算机时代，上述穿孔卡片字符表示方式发展为6比特的字符编码：用4比特表示第0行至第9行的哪一行被穿孔；用2比特表示第11、第12行的哪一

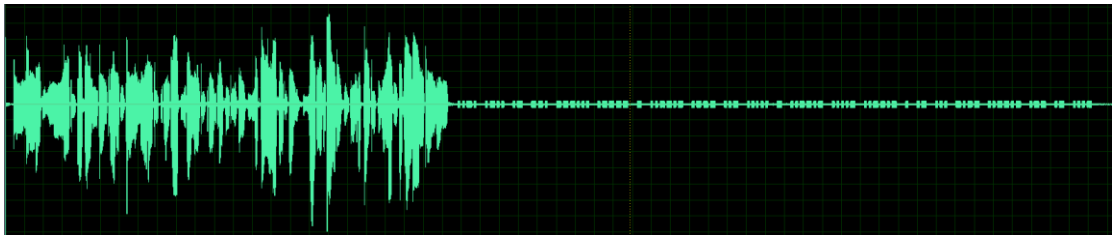
开始进行翻译，但是中间有两个特殊符号也太难搞了。我猜两个都是下划线，提交失败。于是开始了疯狂提交的尝试阶段，工作量太大，调整了一下思路，我猜第一个一定是根下划线，于是只改变第二个特殊符号，挨个试...在%的时候，提交成功了。这题我太靠运气了。。。

3. 听听音乐？

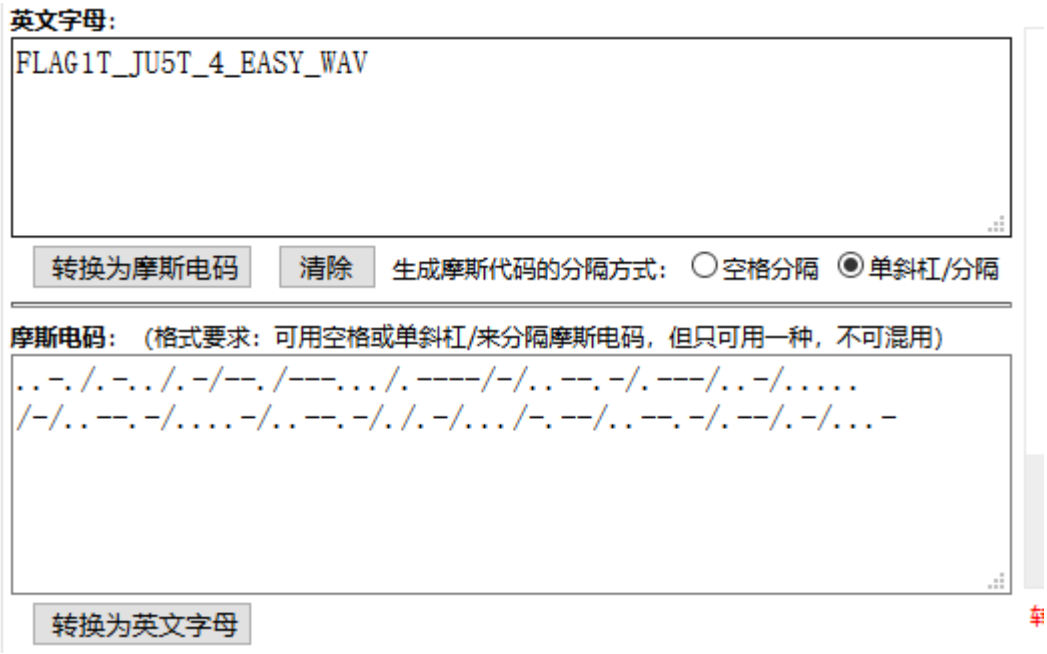
打开是一段 MP3，音频隐写当然使用 MP3stego，却得到了这么一段话。



那就回头听听音频吧。发现歌曲放完之后是一段连续的 didi 声，应该是摩斯电码吧。要想靠听记录我是做不到，于是使用 Cool Edit 查看波形。



看见后面一段，放大后便可得到摩斯电码。记录后进行翻译。



后下划线是我自己添加的，因为之前遇到过摩斯电码的下划线不能被翻译出的情况，所以留意了一番。提交成功。