[zip伪加密](http://blog.csdn.net/etf6996/article/details/51946250)

1. [简述](http://blog.csdn.net/ETF6996/article/details/51946250#t0)
2. [原理](http://blog.csdn.net/ETF6996/article/details/51946250#t1)
3. [背景](http://blog.csdn.net/ETF6996/article/details/51946250#t2)
4. [实例](http://blog.csdn.net/ETF6996/article/details/51946250#t3)
5. [猜想](http://blog.csdn.net/ETF6996/article/details/51946250#t4)
6. [测试一](http://blog.csdn.net/ETF6996/article/details/51946250#t5)
7. [测试二](http://blog.csdn.net/ETF6996/article/details/51946250#t6)

简述

在上一篇博客：[实验吧/隐写术/小家伙挺可爱](http://blog.csdn.net/etf6996/article/details/51925383)中遇到了zip伪加密这个概念，于是乎找了很多资料，来简单记录下我的心路历程吧~^\_^~

原理

zip伪加密是在文件头的加密标志位做修改，进而再打开文件时识被别为加密压缩包。

背景

首先，我们来看zip文件头协议。这里以中文资料为证，但是仍然建议大家去[http://www.pkware.com](http://www.pkware.com/) 阅读官方文档，虽然英文读起来有些困难。

**一个 ZIP 文件由三个部分组成：**

压缩源文件数据区+压缩源文件目录区+压缩源文件目录结束标志   
[查看详情](http://blog.csdn.net/wclxyn/article/details/7288994)

实例

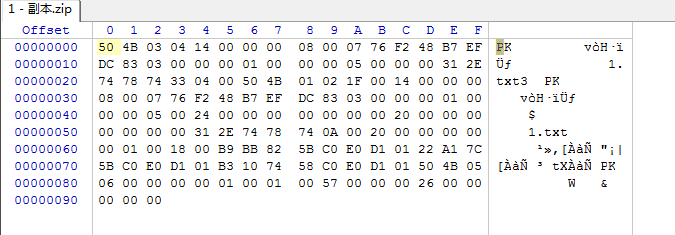
下面给出西普的一个实例：   
这里写图片描述   
**压缩源文件数据区：**   
50 4B 03 04：这是头文件标记（0x04034b50）   
14 00：解压文件所需 pkware 版本   
00 00：全局方式位标记（有无加密）   
08 00：压缩方式   
5A 7E：最后修改文件时间   
F7 46：最后修改文件日期   
16 B5 80 14：CRC-32校验（1480B516）   
19 00 00 00：压缩后尺寸（25）   
17 00 00 00：未压缩尺寸（23）   
07 00：文件名长度   
00 00：扩展记录长度   
6B65792E7478740BCECC750E71ABCE48CDC9C95728CECC2DC849AD284DAD0500   
**压缩源文件目录区：**   
50 4B 01 02：目录中文件文件头标记(0x02014b50)   
3F 00：压缩使用的 pkware 版本   
14 00：解压文件所需 pkware 版本   
00 00：全局方式位标记（有无加密，这个更改这里进行伪加密，改为09 00打开就会提示有密码了）   
08 00：压缩方式   
5A 7E：最后修改文件时间   
F7 46：最后修改文件日期   
16 B5 80 14：CRC-32校验（1480B516）   
19 00 00 00：压缩后尺寸（25）   
17 00 00 00：未压缩尺寸（23）   
07 00：文件名长度   
24 00：扩展字段长度   
00 00：文件注释长度   
00 00：磁盘开始号   
00 00：内部文件属性   
20 00 00 00：外部文件属性   
00 00 00 00：局部头部偏移量   
6B65792E7478740A00200000000000010018006558F04A1CC5D001BDEBDD3B1CC5D001BDEBDD3B1CC5D001   
**压缩源文件目录结束标志：**   
50 4B 05 06：目录结束标记   
00 00：当前磁盘编号   
00 00：目录区开始磁盘编号   
01 00：本磁盘上纪录总数   
01 00：目录区中纪录总数   
59 00 00 00：目录区尺寸大小   
3E 00 00 00：目录区对第一张磁盘的偏移量   
00 00：ZIP 文件注释长度

猜想

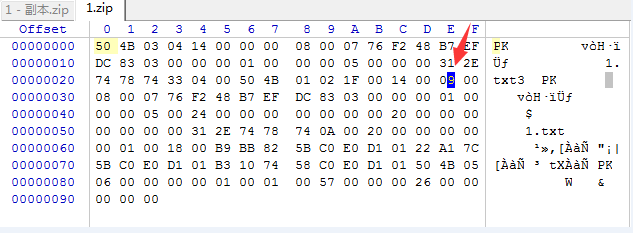
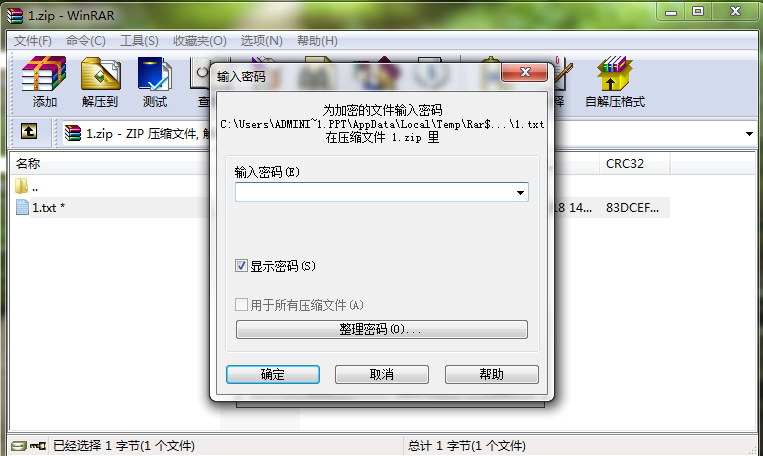
所以如果把一个zip文件的文件头或者加密标志位进行适当修改，那就可能会改变文件的可读性了呗！

测试（一）

亲测例子：先压缩一个1.zip，再用Winhex打开，如下图：



**压缩源文件数据区：**   
50 4B 03 04：这是头文件标记   
14 00：解压文件所需 pkware 版本   
00 00：全局方式位标记（有无加密）   
08 00：压缩方式   
07 76：最后修改文件时间   
F2 48：最后修改文件日期   
……   
**压缩源文件目录区：**   
50 4B 01 02：目录中文件文件头标记(0x02014b50)   
1F 00：压缩使用的 pkware 版本   
14 00：解压文件所需 pkware 版本   
00 00：全局方式位标记（有无加密，这个更改这里进行伪加密，改为09 00打开就会提示有密码了）   
08 00：压缩方式   
07 76：最后修改文件时间   
F2 48：最后修改文件日期   
……

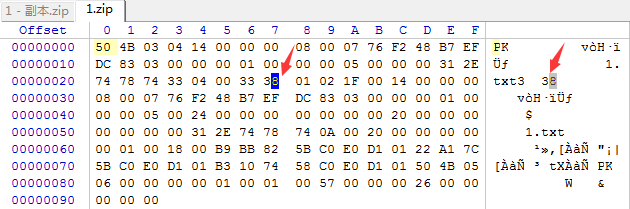
综上所述，如果把第二个加密标记位的00 00改为09 00，打开就会提示有密码：   
   


于是乎，这样就达到了zip伪加密的目的，可是这里又有一个疑问了，你可能会问，为什么改成09，而不是其他的数字呢，其实改成09只是举的一个例子，只要末位是奇数，就代表加密，反之，末位是偶数代表未加密（这是我猜想的，具体操作也符合我的猜想，可我不知道这个原理是什么，如果你了解的话，欢迎来分享分享~^\_^）

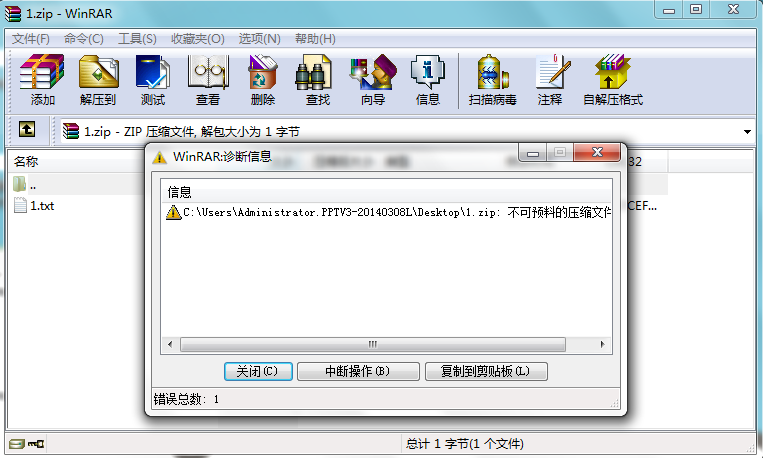
测试（二）

**猜想：**

既然加密标志位可以修改，那头文件标记位应该也能修改吧，试试咯~：



把文本区原来显示的PK位的50 4B改成其他数据，再打开zip文件：

   
显然，这样也使zip文件不可读！