

ExFAT 文件系统 DBR 的分析与重建

赵振洲

(北京政法职业学院 信息技术系, 北京 100024)

摘要: ExFAT 是一个支持更大的存储容量, 又具有更快的存取速度的文件系统, 特别适合大容量移动存储的需要。随着 ExFAT 文件系统的广泛应用, 其数据安全问题日益突出, 尤其是 DBR 遭到破坏后, 整个分区都无法正常访问, 给用户的数据安全带来严重的威胁。目前针对 ExFAT 文件系统的数据恢复软件有 R-Studio4.6、D-Recovery For ExFAT, 这两款软件能够对 ExFAT 分区下删除或格式化的文件进行恢复, 但都不能保证 100% 的恢复。而实际上 DBR 遭到破坏后, 只要不点击格式化, 就完全没有必要进行数据恢复, 只需要通过分析、计算, 然后手工重建 ExFAT 文件系统的 DBR 即可。文中首先介绍 ExFAT 文件系统的结构, 然后对其 DBR 进行分析, 最后系统介绍 DBR 手工重建的步骤、参数计算的方法, 结合校验值计算函数开发了校验值计算工具, 并具体应用到 DBR 的重建过程中, 具有较强的实际应用价值。

关键词: ExFAT; 文件系统; 格式化; DBR; 重建

中图分类号: TP31

文献标识码: A

文章编号: 1673-629X(2012)03-0085-04

Analysis and Reconstruction for ExFAT File System DBR

ZHAO Zhen-zhou

(Department of Information Technology, Beijing College of Politics and Law, Beijing 100024, China)

Abstract: The file system of ExFAT supports a greater storage capacity, but also has faster access speed, especially for high-capacity mobile storage needs. With ExFAT file system widely used, the data security issues become increasingly prominent, especially after the destruction of DBR, the partition can not normally access, data security to the user poses a serious threat. Currently ExFAT file system data recovery software, including R-Studio4.6, D-Recovery For ExFAT, both software can recover formatted or deleted user files, but can not guarantee 100% recovery. In fact, after the destruction of DBR, as long as do not click format, it is not necessary for data recovery, only need to analyze, calculate, and then manually rebuild ExFAT DBR file system can be. It describes the ExFAT file system structure, and then analyzes its DBR, finally introduces DBR manual reconstruction steps, parameter calculation method, combined with the checksum calculation functions developed checksum calculation tools, and specifically applied to the DBR's reconstruction process, with a strong practical value.

Key words: ExFAT; file system; format; DBR; reconstruction

0 引言

微软文件系统经历了 FAT12、FAT16、FAT32、NTFS 几个阶段, 其中 NTFS 文件系统以其非常好的安全性及可恢复性得到广泛认可, 也是微软力推的一种文件系统。

但是, 随着闪存容量的不断增大, 使得目前的文件系统无法更好地适应闪存大容量的需求。由于容量不断增大, FAT32 无法很好地管理较大的容量, NTFS 虽然在容量管理上可以胜任, 但由于它的安全性等方面的原因, 会要求频繁地对闪存进行读写操作。对闪存

而言, 频繁的读写不仅会降低性能, 更会大大降低其使用寿命。另外, NTFS 的延迟写入设计也不适用于便携的移动存储设备。所以, 必须推出一种既可以支持更大的存储容量, 又具有更快的存取速度的文件系统^[1]。

为此, 微软对 FAT 系列文件系统做了进一步发展, 推出了 ExFAT (Extended File Allocation Table File System) 文件系统。ExFAT 既继承了 FAT 类文件系统的简单结构, 同时又增大了对容量的支持。所以, ExFAT 是一种为闪存更好地工作而出现的一种折衷方案^[2-4]。

1 ExFAT 文件系统结构

由于 ExFAT 仍然属于 FAT 类文件系统, 所以它的布局结构总体上仍与 FAT12/16/32 大同小异, 如图 1

收稿日期: 2011-08-12; 修回日期: 2011-11-17

基金项目: 北京市教委优秀青年骨干教师资助项目 (KM201010009006)

作者简介: 赵振洲 (1978-), 男, 辽宁辽阳人, 硕士, 讲师, CCF 会员, 研究方向为信息安全教学与研究。

所示。

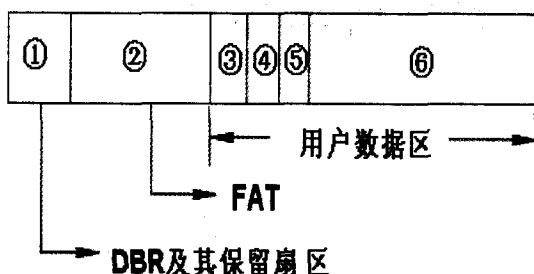


图1 ExFAT文件系统总体结构

ExFAT保持了FAT类文件系统的总体架构,大致分为DBR及其保留扇区、FAT表和用户数据区三大部分^[5]。DBR及其保留扇区是FAT表前的区域,大致可分为三个部分,一是主引导区域,二是备份引导区域,再就是其他保留区域。主引导区域通常占用0~11号扇区,其中0号扇区作为DOS引导记录,简称为DBR;备份引导区域占用12~23号扇区,其中12号扇区是DBR的完整备份;由24号扇区开始至FAT表前一个扇区这部分通常不被使用,权且称其为其他保留区域。

FAT表区域由FAT表组成,不同的是,目前01.00版本的ExFAT文件系统通常只有一个FAT表(事务ExFAT---TexFAT文件系统有两个FAT表),FAT表中的每个FAT项占用4个字节,并全部使用这4个字节的32个bit。

用户数据区则用于存储用户数据。ExFAT的数据区的起始簇号也是2,该簇通常由簇位图文件占用,其位置如图1中③所示,该文件的各种信息记录在根目录下的一个0x81类型的目录项中。跟在簇位图文件后的是大写转换表文件,其位置如图1中④所示,这个文件的大小是固定的,为5836个字节。再后面则是根目录,这是所有文件及文件夹的入口,其位置如图1中⑤所示。根目录之后,如图1中⑥所示,是真正的用户数据区^[6]。

2 ExFAT文件系统的DBR分析

ExFAT文件系统也将0号扇区做为引导记录扇区,除引导信息外,还记录着文件系统的各项参数,如分区大小、FAT表位置及大小、簇起始位置、根目录起始簇号、每扇区大小、每簇扇区数等等^[6]。ExFAT引导记录扇区的主要结构见表1。

3 ExFAT文件系统DBR手工重建

一个ExFAT分区双击打开时出现如图2所示的提示信息。一般情况下,用户会选择“是”,格式化分区,带来的后果是分区下的文件全部丢失,这时用户想要得到原来的文件只能采取分区格式化后的恢复方

法,此方法相对复杂,且不能100%恢复。而实际上出现此种提示,绝大多数情况下是由于DBR遭到破坏引起的,此时应该点“否”,然后重建DBR。

表1 ExFAT引导记录扇区结构

偏移字节 (十六进制)	字节数	含义
00~02	3	跳转代码,“EB 7690”,跳转过0x76个字节,至0x78字节处
03~07	5	分区类型“4558464154”,明文“EXFAT”
08~0A	3	“202020”
0B~3F	53	“00”
40~47	8	分区隐藏扇区数(似乎总是相对于磁盘物理0号扇区)
48~4F	8	分区扇区总数
50~53	4	FAT表起始扇区号
54~57	4	FAT表扇区数
58~5B	4	数据区起始扇区号(注意,虽然只有一个FAT表,但该值并不一定等于FAT表起始扇区号加上FAT表大小扇区数,也就是说为FAT表分配的空间与簇起始扇区之间可能会有未使用的扇区)
5C~5F	4	分区内总簇数
60~63	4	根目录起始簇号
64~67	4	卷ID
68~69	2	文件系统版本
6A~6B	2	卷标志
6C~6C	1	每扇区字节数,假设此处值为N,则每扇区大小字节数为2的N次方个字节,此处值通常为“09”,最大值为“12”
6D~6D	1	每簇扇区数,假设该位置值为N,则簇大小为2的N次方个扇区
6E~6E	1	FAT表个数
6F~6F	1	介质描述符
70~70	1	已用比例
71~77	7	保留
78~1FD	390	引导代码
1FE~1FF	2	签名标志“55AA”

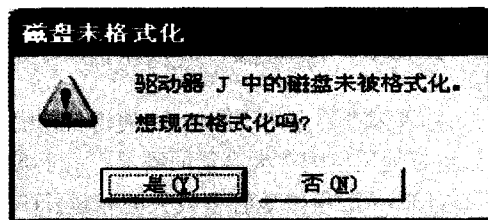


图2 打开分区时出错

重建DBR首先应该考虑到ExFAT分区的DBR有一个备份,在分区的12号扇区,如果备份的DBR没有被破坏,那么直接把备份DBR扇区的数据复制、粘贴到DBR所在扇区即可。而如果备份的DBR也遭到了破坏,那么只能采取手工重建的方法来修复DBR了。

DBR扇区的结构分为以下几个部分:跳转指令、

OEM 代号、BPB 参数、引导代码和结束标志。同一个文件系统类型的 DBR 扇区,除了 BPB 参数有部分不一样以外,其他各部分都是一样的,所以重构 DBR 的方法是从同一个文件系统类型的分区中复制一个 DBR,然后再对 BPB 参数进行修改^[7]。

第1步 复制同版本的 DBR。

先从其它 ExFAT 分区中复制一份完好的 DBR 扇区,写入待修复盘的 0 扇区^[8]。此操作的目的是获取 DBR 扇区里的引导程序信息,因为不同 ExFAT 分区的引导程序是通用的。

第2步 计算 BPB 参数。

ExFAT 文件系统的 BPB 中需要修改的参数有:

- 1) 隐藏扇区数 (Partition Sector Offset)
 - 2) 扇区总数 (Total Sectors in Volume)
 - 3) FAT 起始扇区号 (FAT Location Sector Number)
 - 4) FAT 扇区数 (Size of FAT in Sectors)
 - 5) 首簇起始扇区号 (Bitmap Location Sector Number)
 - 6) 总簇数 (Number of Clusters)
 - 7) 根目录首簇号 (Root Directory Location Cluster Number)
 - 8) 每簇扇区数 (Sectors per Cluster)
- (1) 隐藏扇区数。
- 隐藏扇区数这个参数也就是分区的相对开始扇区号。如果硬盘的分区表没有被破坏,这个参数可以从分区表中查看。

(2) 扇区总数。

扇区总数也可以从主引导记录模板中查看。

(3) FAT 起始扇区号。

FAT 起始扇区号可以通过搜索 FAT 表的方法获得。搜索 FAT 表的具体方法是搜索 FAT 表的头标志“F8 FF FF FF”,搜索到的 FAT 表的扇区位置后,减去分区开始扇区号,既得 FAT 表起始扇区号。

(4) 首簇起始扇区号。

因为簇位图文件一般占用数据区的第一个簇,所以首簇起始扇区号的计算方法是找到簇位图文件的开始位置。同时,数据区开始的一些簇一般都会被使用,所以簇位图文件的前几个字节大多数情况下都是“FF”。所以可以通过搜索“FF FF”来找到簇位图文件的开始位置。搜索到的簇位图文件的扇区位置后,减去分区开始扇区号,既得簇位图文件的起始扇区,这也是分区的首簇起始扇区号^[9]。

(5) 每簇扇区数。

簇位图文件之后就是大写转换表文件了。大写转换表文件的内容是固定的,前 4 个字节是“00 00 01 00”,通过搜索这 4 个字节就能够找到大写转换表文件

的开始地址。搜索到的大写转换表文件的位置后,减去分区开始扇区号,既得大写转换表文件开始扇区号。刚才计算过簇位图文件开始的开始扇区号,用大写转换表文件的开始扇区号减去簇位图文件的开始扇区号,得到簇位图文件的大小。再通过分析 FAT 表当中的 FAT 表项可知,簇位图文件占用的簇的个数,用簇位图文件的大小除以簇位图文件的个数既得每簇扇区数。

(6) 根目录首簇号。

簇位图文件总是开始于数据区的 2 号簇,且通过分析 FAT 表可知簇位图文件占用簇的个数,簇位图文件之后是大写转换表文件,大写转换表文件的大写是固定的 5836 个字节,结合每簇扇区数可知大写转换表文件占用的簇的个数,而大写转换表文件之后就是根目录的开始。

(7) 总簇数。

前面已经知道了首簇起始扇区号、每簇扇区数。用数据区的总扇区数除以每簇扇区数,就是分区的总簇数了。

(8) FAT 扇区数。

FAT 表所占的扇区数不能直接用分区首簇的起始扇区号减去 FAT 表起始扇区号这种方法来计算,因为在 FAT 表之后往往还有很多保留扇区,这样计算就不准确了。

可以用分区的总簇数来计算 FAT 扇区数。因为分区中的每一个簇对应 FAT 表中的一个 FAT 项,假设刚才计算了分区中的总簇数为 63998,而 FAT 表中还有两个保留的 FAT 项,即 0 号项和 1 号项,所以 FAT 表中的 FAT 项总数为 $63998 + 2 = 64000$ 。ExFAT 的每个 FAT 项占 4 个字节,这样就可计算 FAT 表的总字节数为 $64000 * 4 = 256000$,再除以每扇区字节数 512,等于 500,但现在还不能直接把这个数值当做 FAT 表的扇区数。在 ExFAT 文件系统中,FAT 表的大小都是簇大小的整数倍,而 500 并不是当前分区的簇大小 64 的整数倍,再经过一个简单的运算,可以算出 512 是 64 的整数倍,并且是大于 500 又最解决 500 的一个数值,所以当前分区 FAT 表的大小是 512 个扇区。

第3步 修改并保存 BPB 参数。

将计算好的 8 个 BPB 参数填入 DBR 中。可以在 DBR 中直接填写,也可以用模板填写。在 DBR 中直接填写需按照 Little-Endian 字节顺序填写十六进制,用模板需用十进制数填写。填写后存盘。

第4步 计算校验值。

ExFAT 文件系统的主引导区域包括分区的前 12 个扇区,并且最后一个扇区是校验扇区,其内存储的是前 11 个扇区通过校验函数生成的校验值,校验值是 4

个字节的数据,在校验扇区内重复填写^[10,11]。修改前 11 个扇区的任何一个字节都会导致校验值发生变化、校验失效。

因此,经过以上三步后,虽然已经保证 BPB 参数的正确,但由于前 11 个扇区除了 BPB 参数之外的某些字节的数据也发生了变化,已经无法通过校验,分区同样还是会提示格式化。

解决办法就是利用校验函数对分区的前 11 个扇区重新进行校验运算,生成校验值,填入校验扇区。

笔者借鉴资料《Reverse Engineering the Microsoft exFAT File System》中给出的校验值计算函数用 c++编写了一个计算 ExFAT 文件系统校验值计算工具^[12],在该工具窗口中输入存储介质名称及起始扇区位置即可得到 ExFAT 文件系统引导区域的校验值(如图 3 所示)。

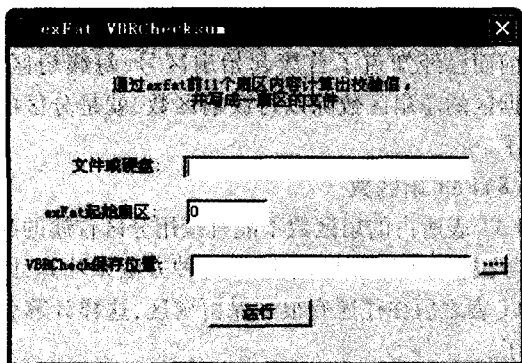


图 3 校验值计算工具

校验值技术函数如下:

```
UINT32 VBRChecksum(const unsigned char octets[], long
NumberOfBytes)
```

```
{
    UINT32 Checksum = 0;
    long Index;
    for (Index = 0; Index < NumberOfBytes; Index++)
    {
        if (Index == 106 || Index == 107 || Index ==
112)
        {
            continue;
        }
        Checksum = ((Checksum << 31) | (Checksum >> 1)) +
(UINT32) octets[Index];
    }
    return Checksum;
}
```

将工具生成校验值填入 11 号扇区,覆盖原来的校验值。覆盖校验扇区后保存,然后再把主引导区域(0

~11 号扇区)的数据复制、粘贴到备份引导区域(12 ~ 23 号扇区)保存,最后在“我的电脑”里双击提示未格式化的分区即可正常打开。

4 结束语

DBR 记录着文件系统的各项参数,包括分区大小、FAT 表位置及大小、簇起始位置、根目录起始簇号、每扇区大小、每簇扇区数等。如果 DBR 遭到破坏,文件系统将无法对分区内的用户数据进行有效的管理,分区将无法正常工作。但 DBR 遭到破坏带来的仅仅是分区的识别问题,用户的数据并没有发生任何的改变。

文中分析了 ExFAT 文件系统的 DBR 结构,给出了 DBR 手工重建的步骤、参数计算的方法,结合校验值计算函数开发了校验值计算工具,并具体应用到 DBR 的重建过程中,取得了良好的效果,具有较强的实际应用价值。

参考文献:

- [1] 钱镜洁. exFAT 文件系统的解析和数据恢复[J]. 电信科学, 2010(11A): 19-21.
- [2] Mamun A A, Guo Guoxiao, Bi Chao. Hard Disk Drive Mecha- tronics and Control[M]. [s. l.]: CRC Press, 2007.
- [3] Microsoft Corporation. File System Functionality Comparison [EB/OL]. 2011-08-12. [http://msdn.microsoft.com/en-us/library/ee681827\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ee681827(VS.85).aspx).
- [4] Microsoft Corporation. exFAT, MSDN Library[EB/OL]. 2007. <http://msdn2.mirosoft.com/en-us/library/aa914353.aspx>.
- [5] 陈潮, 靳慧云. FAT32 文件系统的 DBR 恢复技术研究[J]. 信息安全, 2011(5): 31-32.
- [6] 刘伟. 数据恢复技术深度揭秘[M]. 北京: 电子工业出版社, 2010: 400-404.
- [7] 万亚平. 基于块 I/O 的 RAID 设计[J]. 计算机技术与发展, 2008, 18(3): 135-138.
- [8] 赵振洲. RAID5 数据恢复技术研究[J]. 计算机技术与发展, 2010, 20(6): 121-124.
- [9] 陆莉莉. 基于 DBR 的磁盘文件快速加密安全保障机制研究[J]. 计算机技术与发展, 2011, 21(5): 171-173.
- [10] 黄世权. 网络存储安全分析[J]. 计算机技术与发展, 2009, 19(5): 170-179.
- [11] 钟秀玉, 陈月峰. 基于递归与多线程的丢失文件查找设计[J]. 计算机技术与发展, 2010, 20(9): 98-105.
- [12] Shullich R. Reverse Engineering the Microsoft exFAT File Sys- tem[M]. [s. l.]: The SANS Institute, 2010.