一种基于闪存物理镜像的 FAT 文件系统重组方法

张 丽[1,2],谭毓安[1,3],郑 军[1,3],马忠梅[1],王文明[1],李元章[1]
(1.北京理工大学计算机学院,北京 100081;2.南阳师范学院计算机与信息技术学院,河南南阳 473061;
3.北京理工大学北京市海量信息处理与云计算应用工程技术中心,北京 100081)

摘 要: 文件系统重组是闪存设备取证研究进行数据恢复的主要手段.传统的文件系统重组方法需要同时获取闪存设备在同一时刻的逻辑镜像和物理镜像,该条件在取证实践中常常难以满足,故提出一种仅依赖闪存物理镜像重组文件分配表(FAT)文件系统的方法.在引入统计分析法从物理镜像中提取逻辑地址字段和页状态字段的基础上,给出利用最新页状态值准确重组闪存设备最新 FAT 文件系统镜像的算法.最后以 MTK6229 闪存设备物理镜像的 FAT 文件系统重组过程为例,验证上述重组算法及相关方法是正确的.

关键词: 数字取证; 闪存; 物理镜像; 文件系统重组; 空闲区
中图分类号: TP309.3 **文献标识码:** A **文章编号:** 0372-2112 (2013) 08-1487-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2013.08.006

A Method for Reconstructing the FAT File System from Flash Memory

ZHANG Li[1,2], TAN Yu-an[1,3], ZHENG Jun[1,3], MA Zhong-mei[1], WANG Wen-ming[1], LI Yuan-zhang[1]
(1. *School of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081, *China*;
2. *Department of Computer and Information Technology, Nanyang Normal University, Nanyang, Henan* 473061, *China*;
3. *Beijing Engineering Research Center of High Volume Language Information Processing and Cloud Computing Application, Beijing Institute of Technology, Beijing* 100081, *China*)

Abstract: The file system reconstruction is an effective way of recovering the forensic data from Flash memory. However, the traditional reconstruction methods need a precondition that is there are both the logical image and the physical image of flash memory at the same time and that is usually not satisfied in practice. In this paper, we propose a method for reconstructing the File Allocation Table (FAT) file system of Flash device when only a physical image of Flash memory is acquired. After introducing the statistical methods to identify the logical address bytes and the page state byte from the physical image, we propose the new algorithm to reconstruct the newest FAT file system which is based on the newest value of the page state. At last, take the special flash devices with MTK6229 controllers as examples, we expound the methods related to reconstructing the FAT file system and verify the reconstruction algorithm.

Key words: digital forensics; flash memory; physical image; file system reconstruction; spare area

1 引言

互联网和移动技术的发展将数字取证研究[1]对象从单机取证[2,3]、网络取证[4,5]扩展到以手机为代表的移动设备取证,研究内容也从磁盘通用文档数据的取证[6]发展到手机内存、SIM 卡和闪存数据的取证[7~9].闪存作为一种非易失性的数据存储器件逐渐成为移动设备数字取证领域的研究热点.

闪存设备取证研究主要针对存储在闪存上的数字证据,如:通信记录、音频、视频等信息进行收集、验证、识别、分析、解释以及归档和出示,其目的是重建利用移动设备进行的犯罪活动,为司法机关惩治罪犯提供有效证据.目前获取闪存数据方式主要有两种[10,11]:一种是通过数据线或蓝牙获取移动设备操作系统能访问到的那部分数据,该方式叫做逻辑获取;另一种是利用底层芯片访问技术获取闪存数据的完全拷贝,该方式叫做物理获取.逻辑方式不能获取删除数据以及对取证研究至关重要的数据[10].物理获取方式正好弥补了这一缺陷且还能获取到被移动设备安全代码封锁的数据[11].

物理方式获取到的闪存物理镜像是一个无结构的二进制数据文件,包括闪存固件、文件系统以及闪存管理的元数据.在取证实践中,基于闪存物理镜像获取数

收稿日期:2012-09-10;修回日期:2013-01-31;责任编辑:蓝红杰
基金项目:国家 863 高技术研究发展计划(No.2013AA01A212)

字证据[9~11]的一般方法是利用元数据重组文件系统,再结合传统取证技术中的数据恢复、文件碎片重组[12,13]实现数字证据的重现.

闪存先擦后写的操作特点使得闪存在执行更新操作时,旧数据不会被立刻擦除,而新数据则被重新写入其他数据块[14].结果是在重组闪存文件系统过程中常遇到多个不同物理页具有相同逻辑地址的情况.传统的文件系统重组方法采用实验观察结论,认为上述多个物理页中物理地址最高的页属于当前最新文件系统数据[9,10].但该方法实现的前提是必须同时取得闪存系统在同一时刻的逻辑镜像数据和物理镜像数据.然而在实际取证过程中,该前提条件往往得不到满足.

针对以上问题,作者提出一种新的基于闪存物理镜像的文件系统重组方法.该方法利用统计学的变异系数从镜像空闲区识别出物理页的逻辑地址字段,利用页状态字段唯一标识一个最新物理页的特性,从逻辑地址相同的多个物理页空闲区中定位页状态字段并识别最新页状态值,再根据 FAT 文件系统特征扇区逻辑地址字段与逻辑地址编号的对应关系总结出地址转换公式,最后利用逻辑地址字段和地址转换公式计算每个物理页在文件系统中的逻辑地址编号,按照逻辑地址编号由小到大顺序,结合最新页状态值准确重组出 FAT 文件系统最新的镜像文件.

2 元数据区重要字段的识别

闪存存储空间由块、页、扇区等逻辑单元组成,其中页是最小的可编程单元,块是最小的可擦除单元.每个页单元又被分为数据区(data area)和空闲区(又叫元数据区或备用区,spare area)[10,11].空闲区存储用于闪存管理的元数据.元数据用于记录闪存页的逻辑地址、存储校验码(Error Checking and Correction,ECC)、页状态和坏块信息等.

目前大部分闪存芯片的页大小参数如表 1 所列,其中前一种结构属于小块结构,数据区大小相当于一个扇区,空闲区紧随在数据区之后;后两种属于大块结构,数据区和空闲区大小都是小块结构对应区块大小的整数倍,空闲区集中存放在数据区之后.据此结合物理页大小,可确定空闲区在闪存物理镜像的地址.下面介绍识别空闲区逻辑地址字段和页状态字段位置的方法.

表 1 常见的闪存页大小参数

页大小(Byte)	空闲区大小(Byte)	数据区大小(Byte)
512 + 16/26	16/26	512
2048 + (64/104)	(16/26) * 4	2048
4096 + (128/208)	(16/26) * 8	4096

2.1 识别空闲区逻辑地址字段位置

文献[10]给出一种通过观察统计结果定位空闲区逻辑地址字段的方法.

该方法读取闪存物理镜像中所有的空闲区数据,以 16 字节为一组对每字节数据进行统计(统计值称为频度),统计结果放在一个 256 行 16 列的频度数组中,如图 1 所示.除去字节取值等于 0 和 255 频度值,剩下频度值相同的字节是逻辑地址字段,如第 1 字节;频度值较为平均且在一定范围内随机浮动的字节是 ECC 字段,如第 9 ~ 14 字节.

字节取值	1	2	3-8	9	10	11	12	13	14	15	16
0	1182	788	544	2324	2231	1372	2238	2298	1389	5166	544
1	611	178	0	1835	1764	889	1805	1741	884	4558	0
2	611	678	0	1828	1821	873	1797	1779	923	4590	0
3	611	208	0	1831	1730	916	1805	1874	947	4654	0
4	611	0	0	1785	1789	929	1738	1667	846	4588	0
5	611	0	0	1806	1828	864	1819	1806	842	4589	0
6	611	0	0	1845	1717	857	1733	1742	823	4556	0
7	611	0	0	1817	1816	880	1807	1785	858	4493	0
.....											
16	611	39667	0	1787	1717	892	1793	1772	879	0	0
17	611	39073	0	1718	1810	896	1773	1741	875	0	0
18	611	32992	0	1762	1786	896	1771	1648	929	0	0
19	611	30848	0	1812	1693	822	1813	1746	881	0	0
20	611	0	0	1722	1784	859	1758	1791	830	0	0
21	611	0	0	1831	1707	917	1757	1749	945	0	0
22	611	0	0	1792	1793	864	1818	1728	908	0	0
23	611	0	0	1784	1842	926	1748	1828	884	0	0
.....											

图1 空闲区频度表

上述方法给出了一个识别逻辑地址字段的实验性结论,但人工参与度高.受该方法启发,作者在观察分析大量闪存镜像的空闲区频度值的基础上,发现逻辑地址字段和 ECC 字段字节统计值的典型分布如图 2 所示.由于不同闪存镜像空闲区的字节统计值各不同,因此这里用 a,b,c 等变量表示字节频度(即统计值)的抽象值.经分析可利用统计学上的变异系数(Coefficient of Variation,COV)作为区别逻辑地址字段和 ECC 字段的定量指标.

具体方法如下:

第一步:利用文献[10]的统计法生成空闲区数据的频度数组 N.

第二步:对频度数组 N 中的第 i 列($i = 1,2\cdots\cdots16$)统计非 0 元素的个数 num_i,将频度值等于 0 和字节取值为 0 和 255 的元素从数组 N 中删除.然后对剩余非 0 频度值做相邻元素的减法操作,频度差存放在一个新的二维数组 N'.例如,数组 N' 中第 i 列元素 $x_m = |v_{j+1,i} - v_{j,i}|$, $x_{m+1} = |v_{j+2,i} - v_{j+1,i}|$, $x_{m+2} = |v_{j+3,i} - v_{j+2,i}|$, …,其中 $v_{j,i}, v_{j+1,i}, v_{j+2,i}, v_{j+3,i}$ 分别代表频度数组 N 中第 i 列的非 0 相邻频度值.

第三步:对频度差数组 N' 中的每一列元素计算其

平均值 $\bar{x}$、标准偏差 s 以及每个元素的标准分数 z_i,又称为 Z 分数.计算公式如下:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

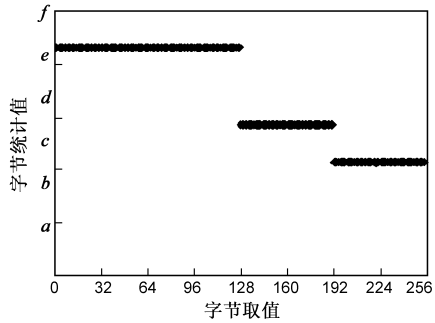
$$s = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n}} \tag{2}$$

$$z_i = \frac{|x_i - \bar{x}|}{s} \tag{3}$$

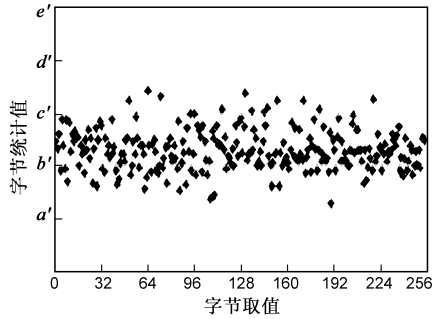
文献[15,16]指出标准分数 $z_i > 3.0$ 的元素属于偏值(outliers).偏值是一个极端数据,在统计分析中需要舍弃以保证统计结果准确性.因此,计算变异系数前要将偏值从频度差数组 N' 中删除.

第四步:对删除偏值后的频度差数组 N' 的每一列元素重新计算其平均值 $\bar{x}$ 和标准偏差 s,最后计算每一列的变异系数 COV,如公式(4).

$$COV = \frac{s}{\bar{x}} \tag{4}$$



(*a*) 逻辑地址字段的典型字节频度值分布



(*b*) ECC字段的典型字节频度值分布

图2

实验表明,逻辑地址字段所属列的变异系数通常都高于 ECC 字段且大于 1.0,同时所有列中变异系数最高的字节一定属于逻辑地址字段.因此,全部逻辑地址字段一定是以变异系数最高的字节为中心的左右列中变异系数大于 1.0 的若干连续列.

2.2　识别空闲区页状态字段

页状态字段标示多个具有相同逻辑地址的物理页中属于当前最新文件系统的唯一物理页.利用该唯一性,识别空闲区页状态字段的方法如下:

①遍历整个闪存物理镜像,基于 2.1 节识别出的逻辑地址字段按照一定规则计算每个物理页的逻辑地址编号(从 0 开始,方法可参考 3.1 节),记录下具有相同逻辑地址编号的多个物理页空闲区的起始地址并将其存储在二维数组 *LSN_list*[*i*][*j*]中,*i* 值等于逻辑地址编号,*j* 表示第 *j* 个具有相同地址编号的物理页,同时记录下地址编号相同的物理页个数 *n*,存储于一维数组 *physical_num*[*i*](*i* 值含义同上);

②查找数组 *physical_num* 中元素值大于 1($n > 1$)的元素下标 *i*;

③读取数组 *LSN_list* 中第 *i* 行的所有数组元素,定位具有相同地址编号的多个物理页空闲区的起始地址,读出这些空闲区的数据存放于二维数组 *spare_data*[*m*][*l*],*m* 表示第 *m* 个具有相同地址编号的物理页,*l* 表示该物理页空闲区的第 *l* 个字节;

④除去 ECC 字段外,将多个空闲区中相同位置的字节数据进行比较,即 *spare_data* 中第 *l* 列的所有元素($m = 0,1,2\cdots\cdots n-1$,*m*、*n*、*l* 含义同上)进行比较.如果所有元素都相同,则继续查找下一列,直到找到第 *k* 列所有元素存在唯一一个不同的数值 *x*,则表明该字节可能是页状态字段并且 *x* 值可能就是最新物理页的状态值;

⑤查找一维数组 *physical_num* 中元素值等于 1($n = 1$)的元素下标 *i*′,读取二维数组 *LSN_list* 中第 *i*′行的 1 个元素,该元素是逻辑地址编号唯一且值等于 *i*′的物理页空闲区地址,读取空闲区第 *k* 个字节数据,如果等于 *x* 则确定空闲区第 *k* 个字节是页状态字段,其取值 *x* 就是最新页的状态值.

⑥如果步骤④不存在有唯一不同值的空闲区字节列,说明该设备未在空闲区设置页状态字段.

上述算法能正确判断一个闪存设备是否使用了页状态字段标示最新物理页.如果使用了该方法,算法还能明确指出状态字段的位置以及最新状态值.当然,并非所有的闪存设备都会设置页状态字段,其他型号的闪存控制器可能采用不同的最新物理页标示方法.

3　基于页状态的 FAT 文件系统重组

利用空闲区逻辑地址字段重组文件系统,需先将逻辑地址字段转换成逻辑地址编号,之后才能根据页状态字段值完成整个文件系统镜像数据重组.

3.1　逻辑地址计算

空闲区的逻辑地址字段并不直接等于上层文件系统的逻辑地址,因为文件系统的逻辑地址属于线性空间(从 0 开始计数),而一个逻辑地址字节表示的地址范

围有限,通常需要多个字节.因此需要先确定逻辑地址字段的存储顺序,即区分最低字节和最高字节,然后推断出由逻辑地址字段计算逻辑地址编号的转换公式.

闪存控制器决定着空闲区逻辑地址字段与文件系统逻辑地址的转换规则.控制器型号不同,地址转换规则有可能就不同.目前市场上闪存控制器型号繁多,因此没有一种统一的算法能直接由逻辑地址字段计算得出逻辑地址编号.这里给出一种适用于文件系统的指导性方法.

①在闪存物理镜像中查找 FAT 文件系统特征扇区 MBR(Master Boot Record)和 DBR(DOS Boot Record)的特征串"000055AA"(十六进制数值串),查找的起始地址是每个扇区的 508 字节处;

②在查找结果中定位一个 MBR 扇区和相近 DBR 扇区的起始地址,分别记做 *addr*1 和 *addr*2;

③从物理镜像的绝对地址 *addr*2 处分别向前向后读取 500 个连续物理页的空闲区数据,如果该闪存镜像同一物理页的多个扇区逻辑地址相同,则将其前 16/26 字节内容存储到二维数组 *spare_area*;否则将其全部空闲区的数据以每 16/26 字节为一组的形式存储到二维数组 *spare_area* 中;

④根据 2.1 节提出的逻辑地址字段识别方法确定二维数组 *spare_area* 中逻辑地址字段的列下标 $j_1, j_2, \cdots\cdots j_n$;

⑤针对逻辑地址字段的每一列 j,执行相邻行元素的减法操作即 $spare\_area[i+1][j] - spare\_area[i][j]$,统计其差值等于 1 和 0 的个数分别存储到一维数组 $num_1[j]$和 $num_0[j]$中;

⑥num_1 值最大的一维数组元素对应的逻辑地址字段列 j_l 是逻辑地址的最低字节,满足 num_0 值最大的数组元素对应的逻辑地址字段列 j_m 是逻辑地址的最高字节;

⑦假设识别出逻辑地址字段 $a_0a_1a_2\cdots\cdots a_n$ 的存储顺序是 a_0 是最低字节数据,a_n 是最高字节数据,由逻辑地址字段计算逻辑地址编号的方法是:

$$LSN = a_0 + a_1 * 256^1 + a_2 * 256^2 + \cdots\cdots + a_n * 256^n \tag{5}$$

⑧根据 *addr*1 和 *addr*2 定位 MBR 扇区和 DBR 扇区,获取其空闲区逻辑地址字段内容(即 $a_0a_1a_2\cdots\cdots a_n$),识别出最低字节和最高字节位置,MBR 扇区在文件系统中的逻辑地址编号(*LSN*)是 0,DBR 扇区的逻辑地址编号(*LSN*)存储于 MBR 扇区的第 454 字节(从第 0 字节开始计数),占用 4 个字节,利用上述信息验证公式(5)的正确性.

上述方法在推导和验证地址计算公式过程中使用了 FAT 文件系统的特征信息,即 MBR 和 DBR 扇区的特征串"000055AA".虽然现有移动设备使用的 FAT 文件系统有 FAT12,FAT16,FAT32 和 exFAT4 种格式之分,但它们 MBR 和 DBR 扇区的特征数据相同,因此不同的 FAT 文件系统不会对上述逻辑地址的计算方法产生影响.

3.2 最新文件系统镜像的重组算法

假设已确定空闲区逻辑地址字段从第 i 个字节起(从 0 字节开始),占用 n 个字节空间,页状态字段位于第 k 个字节位置,最新状态值是 x,另外已知物理页数据区大小为 L 字节,空闲区大小为 S 字节,最新文件系统镜像的重组算法如下:

①打开闪存物理镜像,定位每个物理页的空闲区位置.假设当前读指针位于物理页的起始地址 *start* 处,空闲区首地址是 $start + L$;

②读取空闲区页状态字段内容 p,地址是 $start + L + k$,长度 1 个字节;

③如果 $p = x$,则读取逻辑地址字段内容,地址是 $start + L + i$,长度 n 个字节,根据式(5)计算该物理页的逻辑地址编号,记做 *LSN*,同时记录下物理页的物理地址 *start*,将 *start* 存储到一维数组 *LSN_list* 中,下标即 *LSN* 值;

④如果 $p != x$,则重复执行步骤①~④定位下一个物理页的空闲区位置,读指针 $start = start + L + S$,直到物理镜像文件读取结束.此刻上层文件系统中所有最新物理页的地址信息已全部存储于一维数组 *LSN_list* 中,同时该数组已经按照逻辑地址编号从小到大的顺序排好序;

⑤依次取出数组 *LSN_list* 的每个元素 *physical_addr*,下标是 LSN_{cur},将读指针定位于闪存物理镜像 *physical_addr* 地址处,读取 L 字节数据存储到新文件 B 的 $LSN_{cur} * L$ 字节处;

⑥如果当前元素是最后一个数组元素,则执行步骤⑧;否则执行步骤⑦;

⑦如果 $LSN_{cur} + 1 < LSN_{next}$,则在新文件 B 的 $(LSN_{cur} + 1) * L$ 地址处填充 $(LSN_{next}\text{-}LSN_{cur}) * L$ 个字节的 $0x00$ 数据,然后读取下一个数组元素,$LSN_{cur} = LSN_{next}$,重复执行步骤⑤~⑦;

⑧数组 *LSN_list* 读取结束,新文件 B 即是最新文件系统的镜像文件.

4 实验结果

为了说明基于页状态字段重组最新文件系统算法的运行效率并验证所有与之相关的方法的正确性,使用 2 款 MTK(主控)手机作为实验对象.这 2 款手机使用

FAT 文件系统,内置闪存芯片的详细信息如表 2 所列.从中可知飞利浦 X800 手机的闪存芯片属于小块结构,由此可判断出空闲区起始地址在每个物理页的第 512 字节处.而 LG KM330 手机闪存是大块结构,其空闲区起始地址在物理页的第 2048 字节处.闪存物理镜像均采用从印刷电路板(PCB)上物理拆除闪存芯片,利用闪存编程器读取数据的方式获取.

表 2　实验对象的闪存信息

设备	闪存型号	控制器型号	物理页大小(Byte)
飞利浦 X800 手机	HY27xA081G1M/A	MTK6229	512 + 16 = 528
LG KM330 手机	TC58NVG0S3AFT	MTK6229	(512 + 16) * 4 = 2112

4.1　逻辑地址字段的识别

按照第 2.1 节提供的逻辑地址字段识别方法计算出空闲区频度差值表 N' 中每列元素的变异系数 COV,其结果如表 3 所列.表中 i 值表示每 16 字节空闲区的第 i 个字节,COV 代表变异系数值,黑色加粗部分是 COV 值大于 1.0 的多个连续字节,其中包含 COV 值最大的字节.由第 2.2 节的结论可知这些字节就是逻辑地址字段.

表 3　空闲区每个字节频度差的变异系统值

飞利浦 X800 手机								
i	0	1	2	3	4	5	6	7
COV	0.71	0.49	0.68	0.81	1.41	0	0	0
i	8	9	10	11	12	13	14	15
COV	0.79	0.69	0.85	0.87	1.35	2.82	6.04	0.83
LG KM330 手机								
i	0	1	2	3	4	5	6	7
COV	0.74	1.11	0.82	0.54	1.02	8.51	1.17	0
i	8	9	10	11	12	13	14	15
COV	0.73	0.66	0.72	0.51	0	0	0.59	1.34

4.2　逻辑地址计算

按照 3.1 节所述步骤计算相邻物理页空闲区逻辑地址字段之间的差值并分析其规律,发现飞利浦 X800 和 LG KM330 手机内置闪存逻辑地址字段的存储顺序都是由低到高,即第 12/4 字节是最低字节,13/5 字节次之,第 14/6 字节是最高字节.分析这些字节数据的变化范围,总结出这 2 款手机闪存利用逻辑地址字段计算逻辑地址编号的公式如下:

$$LSN = spare\_area[12/4] + spare\_area[13/5] * 256 + spare\_area[14/6] * 256 * 256 \quad (6)$$

在上述设备闪存的物理镜像中定位每个 MBR 扇区的第 454 字节读取长度为 4 个字节的 DBR 逻辑扇区编号都为 1,然后再读取 DBR 扇区对应空闲区的逻辑地址字段内容“0x010000”,利用公式(6)计算其逻辑地址编号也等于 1,由此证明逻辑地址计算公式是正确的.另一方面也验证了上述通过计算 COV 值识别逻辑地址字段的方法是正确的.

4.3　页状态字段的识别

按照 2.2 节提供的识别页状态字段步骤,找出 X800 手机和 KM330 手机闪存物理镜像中多个具有相同逻辑地址编号的物理页空闲区数据,如图 3 和图 4 所示,大方框区域是空闲区逻辑地址字段,其中前几个数据块内逻辑地址字段相同的物理页有多个,最后 1 个数据块逻辑地址(扇区)字段相同的物理页只有一个.

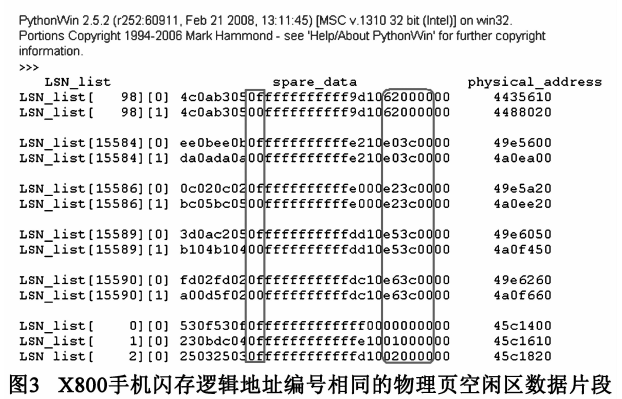


图3　X800手机闪存逻辑地址编号相同的物理页空闲区数据片段

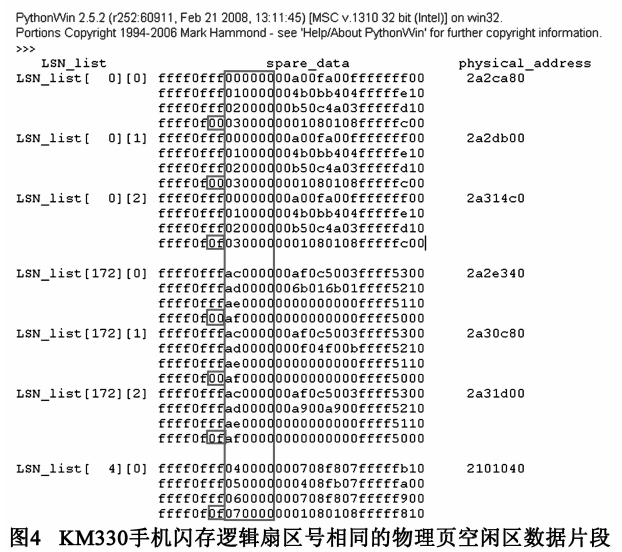


图4　KM330手机闪存逻辑扇区号相同的物理页空闲区数据片段

对比同一个逻辑地址编号的多个物理页空闲区数据发现,X800 手机空闲区除去第 0~3 字节是 ECC 数据不同外,只有第 4 字节(见图 3 窄方框区域)的数据不相同,因此可推断该字节是页状态字段,同时对比逻辑地址编号唯一的最后一个物理页空闲区可知 0x0F 值是最新页状态值;而 KM330 手机整个空闲区只有第 51 个字节存在唯一不同值 0x0F(见图 4 小方框区域),同时对比最后一个物理页空闲区的第 51 字节数据可知,该唯一不同值 0x0F 就是最新页状态值,第 51 字节即页状态字段.

4.4　重组算法的运行效率和准确率测试

在已知逻辑地址字段位置、页状态字段位置、最新页状态值基础上,运行 FAT 文件系统最新镜像重组算

法,测试算法运行效率和准确率.该算法用 python 脚本实现,运行平台使用 Windows XP 操作系统,酷睿双核 CPU(2.10GHz),2GB 内存.另外为了全面反映新重组算法的性能,针对相同实验对象在相同的运行平台上运行文献[10]提供的基于物理地址最高的物理页组成文件系统的传统重组算法.最终两种重组算法的实验对比结果如表 4 所列.

由算法效率的实验数据可知,传统重组算法的运行效率低于新算法.分析原因在于传统算法需要生成一个二维数组用于存储每个逻辑页对应的多个逻辑地址相同的物理页;而新算法只需生成一个一维数组用来存放每个逻辑页对应的唯一一个物理页.另外,针对不同实验对象,新算法的运行效率是不同的.从具体测试数据看,LG KM330 的重组运行效率大约是飞利浦 X800 的 4 倍.分析其原因是由于 KM330 手机闪存的物理页大小是 X800 手机闪存的 4 倍(见表 2),而重组算法是以物理页为单位进行数据处理的.

新旧算法准确性的验证分别针对两类文件.一类是文件系统的元数据文件,一类是用户数据文件.元数据文件利用重组生成的 FAT 文件系统镜像,定位 MBR 和 DBR 扇区等特殊扇区的实际位置,根据这些扇区提供的参数信息,如分区表项、FAT 表个数及大小等内容计算文件系统一些重要数据结构(如 DBR,根目录,FAT 表等)的物理地址,并用实际地址与之进行对比.用户数据文件需事先设置定位标示符并存储于手机可视存储空间的根目录下.准确性验证方法同样利用重组生成的 FAT 文件系统镜像,根据 DBR 参数找到根目录位置,利用文件名查找到对应目录项,从中获知用户文件所在簇号,根据簇号计算文件地址,另利用定位标示符定位文件的实际地址,并与之前计算的地址进行比较.比较结果相同说明重组正确,否则说明重组错误.应用上述验证方法测试新旧重组算法的准确率,其结果如表 4 所示.

表 4 重组算法的运行效率对比

设备	重组算法	物理镜像大小(MB)	最终文件系统大小(MB)	实际处理数据大小(MB)	平均运行时间(s)	效率(MB/s)	准确率
飞利浦 X800	传统算法	128+4	61	65	21.9	2.97	61.2%
	新算法	128+4	61	65	21.4	3.04	100%
LG KM330	传统算法	128+4	91.5	95.5	9.65	9.90	94.1%
	新算法	128+4	91.5	95.5	9.29	10.28	100%

5 结论及进一步工作

传统的闪存文件系统重组需要同时取得闪存设备在同一时刻的逻辑镜像数据和物理镜像数据,该条件在取证实践中常常得不到满足.本文提出一种利用页状态信息准确重组闪存 FAT 文件系统最新镜像的方法,仅仅依靠闪存物理镜像,从中提取物理页逻辑地址和最新页状态值,以重组 FAT 文件系统,实验过程从两种手机的 Flash 镜像中恢复出完整的 FAT 文件系统.上述所有步骤都只使用闪存的物理镜像数据,大大减少了传统重组方法对闪存逻辑镜像的依赖.该重组方法适用于设置了页状态字段并且采用 FAT 文件系统的闪存设备.在今后的研究工作中,将扩展闪存设备的型号和文件系统类型,比如 YAFFS 等.

参考文献

[1] 许榕生.我国数字取证技术研究的十年回顾[J].计算机安全,2011,3:17-19.
Xu Rongsheng. Decade review of digital forensic research[J]. Computer Security, 2011, 3: 17-19. (in Chinese)

[2] 丁丽萍,王永吉.计算机取证的相关法律技术问题研究[J].软件学报,2005,16(2):260-273.
Ding Liping, Wang Yongji. Study on relevant law and technology issues about computer forensics[J]. Journal of Software, 2005, 16(2): 260-273. (in Chinese)

[3] 陈龙,娄晓会,王国胤.基于有限射影几何的细粒度数据完整性检验方法[J].电子学报,2011,39(12):2850-2855.
Chen Long, Lou Xiaohui, Wang Guoyin. An integrity check method for fine-grained data based on finite projective geometry[J]. Acta Electronica Sinica, 2011, 39(12): 2850-2855. (in Chinese)

[4] 孙国梓,耿伟明,陈丹伟,申涛.基于可信概率的电子数据取证有效性模型[J].计算机学报,2011,34(7):1262-1274.
Sun Guozi, Geng Weiming, Chen Danwei, Shen Tao. One validity model of digital data forensics based on trusted probability. Chinese Journal of Computers, 2011, 34(7): 1262-1274. (in Chinese)

[5] 王文奇,苗凤君,潘磊,张书钦.网络取证完整性技术研究[J].电子学报,2010,38(11):2529-2534.
Wang Wenqi, Miao Fengjun, Pan Lei, Zhang Shuqin. The research on integrity of network-based forensic[J]. Acta Electronica Sinica, 2010, 38(11): 2529-2534. (in Chinese)

[6] 钟巍,孔祥维,尤新刚,王波.基于态函数的离散分数余弦倒谱变换在取证话音信息隐藏中的应用[J].电子学报,2012,40(3):595-599.
Zhong Wei, Kong Xiangwei, You Xingang, Wang Bo. Forensic speech information hiding using fractional cosine-cepstrum transform[J]. Acta Electronica Sinica, 2012, 40(3): 595-599. (in Chinese)

[7] Shafik G, Punja, Richard P. Mislan. Mobile device analysis[J]. Small Scale Digital Device Forensics Journal, 2010, 2(1): 1-

15.

[8] Vrizlynn L L Thing, Kian-Yong Ng, Ee-Chien Chang. Live memory forensics of mobile phones[J]. Digital Investigation, 2010,7(1):74 – 82.

[9] 易凌鹰.基于闪存数据恢复的计算机取证技术的研究与实现[D].北京:北京邮电大学硕士学位论文,2009,6.
Yi Lingying. Research and implementation of computer forensics technology based on data recovery from flash memory[D]. Beijing: Beijing University of Post and Telecommunications of China, 2009, 6. (in Chinese)

[10] Marcel B, Martien DJ. Forensic data recovery from flash memory[J]. Small Scale Digital Device Forensic, 2007, 1(1):1 – 17.

[11] C. Klaver. Windows mobile advanced forensics[J]. Journal of Digital Investigation, 2010, 6:147 – 167.

[12] 肖腾,许榕生.基于差异度的 JPEG 碎片重组方法[J].计算机工程,2011,37(10):263 – 265.
Xiao Teng, Xu Rongsheng. Reassembling method for JPEG fragment based on degree of difference[J]. Computer Engineering, 2011, 37(10):263 – 265. (in Chinese)

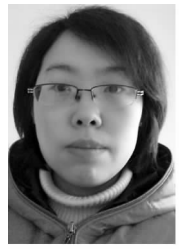
[13] Scott Hand, Zhiqiang Lin, Guofei Gu etc. Bin-carver: automatic recovery of binary executable files[J]. Digital Investigation, 2012, 9:s108 – s117.

[14] 时正,纪金松,陈香兰,等.一种基于差分进化的 Flash 文件系统垃圾回收算法[J].电子学报,2011,39(2):280 – 284.
Shi Zheng, Ji Jin-song, Chen Xiang-lan, Gong Yu-chang. A garbage collection algorithm for flash file system based on differential evolution[J]. Acta Electronica Sinica, 2011, 39(2): 280 – 284. (in Chinese).

[15] Shiffler R E. Maximum Z score and outliers[J]. The American Statistician, 1988, 42(1):79 – 80.

[16] Steven J P. Intermediate Statistics: A Modern Approach[M]. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1990. 456 – 502.

作者简介



张　丽　女,1978 年出生,河南南阳人.1997 年、2001 年分别在河南师范大学、湖南大学获理学学士和工学硕士学位,现为在读博士生,从事数字取证、嵌入式系统等有关研究.
E-mail:hnnyzli@bit.edu.cn

谭毓安　男,1972 年出生,重庆巫溪人.教授、博士生导师.现主要从事为信息安全、嵌入式系统等方面的研究工作.
E-mail:tan2008@bit.edu.cn

郑　军(通信作者)　女,1969 年出生,辽宁鞍山人.副教授、博士.现主要从事信息安全和云计算等方面的研究工作.
E-mail:zhengjun@bit.edu.cn