第 35 卷 第 9 期 计 算 机 工 程 2009 年 5 月
Vol.35 No.9 Computer Engineering May 2009
·安全技术· 文章编号：1000—3428(2009)09—0176—03 文献标识码：A 中图分类号：TP309.2

基于 FAT32 文件系统的计算机取证研究与实现

王中杉，刘乃琦，秦 科，郝玉洁
(电子科技大学计算机学院，成都 610054)

摘 要：针对 FAT32 文件系统，分析离散存储碎片，提出一种基于部分匹配预测算法 PPMC 来重构文件碎片的模型。采用 PPMC 算法确定出任意 2 个碎片的相邻性概率值，通过剪枝技术逐步加工处理，重构出一个有完整顺序的原文件，并分析系统中的隐藏文件 index.dat。
关键词：数据恢复；文件碎片；取证系统；文件重构；index.dat 文件

Computer Forensics Research and Implementation Based on FAT32 File System

WANG Zhong-shan, LIU Nai-qi, QIN Ke, HAO Yu-jie
(School of Computer, University of Electronic Science & Technology, Chengdu 610054)

【Abstract】Aiming at on FAT32 file system, this paper emphasizes to analyze scattered fragments of disk files, and proposes a model of reassembling deleted file fragments based on PPMC algorithm. Employed Prediction by Partial Matching(PPM) is used to build a context model and compute candidate probabilities of the possible adjacency of two document fragments, and pruning technology is adopted to process gradually and reassemble a complete file. It also analyzes the hidden file named index.dat in the system.
【Key words】data recovery; file fragments; forensic system; file reassembly; file of index.dat

1 概述

取证分析是对已发生事件进行保存、重构、归档、翻译的一个分析过程[1]。它致力于对犯罪分子如何入侵计算机以及在入侵成功后所做的事情给出一个完整的回答。各种数字证据一般都以网络日志、文本文档、图像、视频等形式存储。因此，取证分析员需要对犯罪分子所使用的计算机进行取证，主要包括其访问过的文档、网站、打印的文件及已删的文件等。这里只谈已删除文件的恢复问题。当磁盘上数据被删除后，如何重新找回这些数据，并对数据加以分析则成了取证分析员的关键步骤。

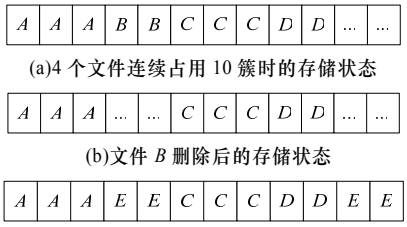
已删除文件在未被其他数据覆盖之前，其在磁盘上的存储方式分为连续存储和离散存储。前者为文件连续占用多个簇。但实际上随着文件的不断创建、修改及删除，离散存储会大量出现，即形成文件碎片。因此，取证分析员提取证据，可能将面临大量文件碎片却没有很好的办法来重构文件。

本文将分析已删文件离散存储恢复的关键技术，给出一种基于 PPM 算法的剪枝重构技术，以加快对离散存储文件的正确重构，并提出了 index.dat 文件对取证中的重要性。

2 已删除文件离散存储恢复技术

2.1 文件碎片成因原理

图 1(a)给出了 4 个文件连续占用 10 个簇的存储状态，即不存在碎片。在图 1(b)中，文件 B 被删除，B 原来占用的簇标记为空簇。当用户再次创建一个大于文件 B(超过 2 簇)的文件时，则其存储方式可能变为图 1(c)中的形式存储。此时图 1(c)中就形成了文件碎片。在面对大量碎片时，取证分析员若采用手工进行碎片重构将是一个十分繁琐且单调的事情，且一些文件并非易读文件，如二进制文件、图像文件等。



(a)4 个文件连续占用 10 簇时的存储状态
(b)文件 B 删除后的存储状态
(c)创建文件 E 后的存储状态

图 1 文件在磁盘上的存储方式

2.2 文件碎片重构问题抽象化

文件碎片离散存储后的重构问题可以形式化地表述如下：假设存在一个集合 $S=\{A_0,A_1,A_2,\cdots,A_n\}$，集合 S 中各元素为文件 A 的碎片，则可以通过一个变换 π，使得

$$A = A_{\pi(0)} \| A_{\pi(1)} \| \cdots \| A_{\pi(n)} \quad (1)$$

那么，通过寻找这样一个变换 π，就可以确定某个碎片 A_i 在原始文件 A 中所处位置。即通过一定的变换法则可将碎片重构成完整的文件。

现在需要确定与任一碎片 A_i 相邻的碎片 A_k，若所有碎片的相邻性全部确定，则文件碎片原始顺序就已经排好了，即文件重构完成。

本文提出碎片相邻性概率问题，**将碎片 A_i 与碎片 A_k 相邻性概率量化为一个权值 $C_{(i,j)}$，若能够正确地量化出每对相邻碎片概率值，则问题就等价于求取所有碎片相邻性概率之和最大值。下面计算基于变换 π 的权值之和 S:**

作者简介：王中杉(1982－)，男，硕士研究生，主研方向：计算机信息系统安全；刘乃琦，教授；秦 科，博士；郝玉洁，副教授
收稿日期：2008-07-30 **E-mail：**kaluo_tech@yahoo.com.cn

$$S=\sum_{i=0}^{n-1}C_{(\pi(i),\pi(i+1))} \tag{2}$$

而寻找 S 的最大值问题，就等同于图解问题，将所有碎片看作图的顶点 V。每个顶点均用一条直线连接，形成一个基于 n 顶点的完全图 G，其中 $C_{(i,j)}$表示碎片 j 在碎片 i 之后的概率。则变换 π 等同于图 G 的一条路径 L，其中 L 为一条遍历 G 中所有顶点 V，且权值之和最大的路径。这就归纳为求图 G 中一条权值和最大的汉密尔顿路径(Hamiltonian path)L。这种优化问题被认为是难以处理的。

实际上碎片重构问题一般发生在多个文件的多碎片问题上，导致复杂性极大地增加。设存在 n 个碎片，其归属于 k 个文件中。若将其抽象为图的问题，则需要在图中寻找 k 条顶点不重叠的路径。图 2 反映了这样一个图的路径问题。其中，路径 *ABE, CDHF* 分别代表 2 条不相交的路径。但寻找图 G 满足上述条件的最优解问题却是很难的，也被认为是 NP-Complete 问题[2]。

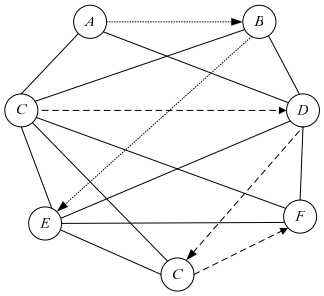


图 2　7 个碎片构成的图

2.3　基于PPMC算法的概率预测

基于上面的分析可知，需要确定 2 个问题:

(1)量化 2 个碎片相邻概率值。

(2)寻找式(2)中最大值问题。

PPMC 算法是一种基于有限序上下文统计模型的技术并已成为一种无损数据压缩技术规范[3]。它采用固定序列的上下文模型，序列值从 0 到指定的最大值 w，用来预测接下来的字符。其中，序列值代表估计下一个字符所依赖的前面字符的长度。这里抽象表述为一个基于一定大小 k 的滑动窗口 S，将 S 置于碎片的末端，某次滑动一个位置，在这个位置上来预测可能出现的字符。

但当某一元素在上下文中未出现时，则必须采用逃逸机制来处理。这个过程循环进行，直到 order-i 能够正确地预测出下一个字符元素。通过这种方式可以求得正确匹配该字符的概率。假定输入字符串 L 为 *abcabcbcacb*，且采用 order-2 顺序处理，则共有 4 种上下文模型(o=2,1,0,-1)。如表 1 所示，给出了 o=2,1 时相应字符的预测概率，在每一种模型中计算下一个字符出现的概率。这个概率的问题已经由文献[4]给出了一个详细的算法编码来描述。

表 1　利用 PPMC 为字符串 *abcabcbcacb* 建立上下文模型的情形

O=2			O=1		
上下文	个数	概率	上下文	个数	概率
ab→c	2	2/3	*a→b*	2	2/5
→esc	1	1/3	*→c*	1	1/5
ac→b	1	1/2	*→esc*	2	2/5
→esc	1	1/2	*b→c*	3	3/4
bc→a	2	2/5	*→esc*	1	1/4
→b	1	1/5	*c→a*	2	1/3
→esc	2	2/5	*→b*	2	1/3

通过上面的描述，可以将文件碎片依次通过 PPMC 算法进行数据加工，形成一系列单一模型 $X_1,X_2,\cdots,X_n$。然后选择一定大小 k 的滑动窗口，将窗口底部置于任一碎片 X_i 的末端，向前滑动一个单位，根据该碎片 X_i 加工后的模型来推测碎片 $X_k(k\neq i)$中第 1 个字符出现的概率 p。在重构磁盘文件碎片时，若没有损失数据，则可认为概率值 p 即为任意 2 个碎片可能相邻的概率。

2.4　启发式剪枝技术

如前文所述，寻找 Hamiltonian path 最优解问题为 NP-Complete 问题。现假定文件的首个碎片可确认，则可将文件首个碎片看作树的顶点，建立树模型，其树中各顶点代表碎片。通过 WinHex 工具分析可知，文件的首部内容一般为特定字符，用于描述文件的格式。如 bmp 图一般是字符串“BM”开始等。因此，假定文件首个碎片可确认是符合实际的。

在树模型中寻找最优解问题归为在树模型的各条路径中寻找一条权值之和最大且解最优的路径。现假设存在 4 个文件碎片，则图 3 描述了该树模型的建立。

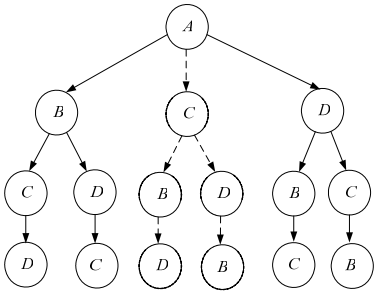


图 3　4 个碎片可能顺序形成的树模型

从图 3 可以看到，当碎片数量增多时，整个树模型将呈指数级别膨胀。因此，排除一些不能满足条件的解路径问题是必须解决的问题。本文采用了极大极小值搜索技术，同时配合 α-β 剪枝技术[5]，对碎片节点构成的树进行优化处理。其优化结果并不是一条单一路径，而是优化出一个具有启发性且重构概率大的路径集。

3　Internet活动记录(index.dat)

当前 IE 浏览器被广泛地应用，也是众多 Web 浏览器中使用最多的浏览器之一。为了使用户能快速地重新访问已经访问过的网址，IE 缓存了已经访问过的网页内容，包括时间、地址及图片等。而这些信息被存放在 index.dat 文件中，该文件为隐藏文件且被系统保护。文献[6]给出了详细的阐述。

此类文件在正常情况下是禁止用户删除的，index.dat 文件包含大量的记录信息。主要类型为 HASH, URL, LEAK 及 REDR。这对取证分析员有极大的帮助，特别是文件 MAC 时间，对文件碎片重构有极大的帮助。同样可以采用磁盘读取技术，将磁盘上的这类文件全部展现出来，包括 MAC 时间、文件类型、用户名等。目前做的比较好软件有 X-ways Trace。

4　实验结果分析对比

虽然数据恢复技术已经引起了人们的大量关注，但国内外并没有很好的数据恢复软件，其恢复的准确率不高。在针对离散存储的文件碎片可读性恢复上，其准确率则更低。

为了验证本文提出的基于 PPMC 算法、α-β 启发式剪枝原理的碎片重构算法，这里将笔者开发的数据恢复软件 SmoothRecovery 同市面上宣称十分优秀的 EasyRecovery professional 软件进行了测试比较，得出的测试对比结果如表 2 和表 3 所示。

表 2　EasyRecovery professional 软件恢复测试

如何删除文件	可否识别	可否恢复	文件大小/KB	读取时间/s	恢复时间/s
深层子目录下的文件	是	常恢复不可读	2.0	18.3	5.3
文件连同目录均删除	部分识别	恢复后不可读	2.0	-	-
深层子目录及文件均删除	部分识别	否	2.0	-	-
已删除的离散文件碎片重构	否	否	320	-	-

表 3　笔者开发的 SmoothRecovery 软件恢复测试

如何删除文件	可否识别	可否恢复	文件大小/KB	读取时间/s	恢复时间/s
深层子目录下的文件	是	恢复后可读	2.0	4.3	3.2
文件连同目录均删除	是	恢复后可读	2.0	3.8	3.6
深层子目录及文件均删除	是	恢复后可读	2.0	4.6	3.6
已删除的离散文件碎片重构	是	可部分重构	320	-	-

从实验分析可知，EasyRecovery professional 并没有很好地解决目录连同文件一并删除这一问题。面对文件首簇号高 16 位被清零的情形，没有提出很好的算法。若待扫描分区很久未进行磁盘碎片整理时，文件删除后即使文件仍连续存储，EasyRecovery 也不能将恢复出来的文件进行正确识别，即文件不可读。

在面对离散储存的文件碎片重构问题时更是没有涉及，没有针对离散的碎片进行扫描、重构等操作，而且其程序的时间复杂度明显偏大。

前文已论述过，在保证证据正确的前提下，取证是尽可能地获取目标机上的所有证据。然而 EasyRecovery 软件给出的已删除文件修改时间并不完全正确，只是粗略地给出了文件修改时间。文件包括文件创建时间、访问时间、修改时间，这里称之为 MAC 时间。MAC 在各种情况下均有很大变化。因此，从磁盘分区读取的文件 MAC 时间并不一定是正确的。如果获取的时间与真实时间发生出入，在取证角度上说，则是致命的。已有学者对 MAC 时间进行了详细的分析[7]。

5　结束语

针对数据恢复的难点热点问题，本文提出了基于 PPMC 算法、α-β 启发式剪枝原理的碎片重构算法。并将国外数据恢复软件 EasyRecovery 同笔者开发的 SmoothRecovery 作了测试分析。下一阶段工作将对移动设备的数据恢复问题进行深入的研究。未来的犯罪将不仅局限于计算机，利用手机等移动设备犯罪率也会不断出现，而且基于移动手机的取证问题已经成为了取证的焦点。

参考文献

[1] Dixon P D. An Overview of Computer Forensics[J]. IEEE Potentials, 2005, 24(5): 7-10.

[2] Vygen J. Disjoint Paths[R]. Bonn, Germany: Research Institute for Discrete Mathematics, University of Bonn, Tech. Rep.: 94816, 1994.

[3] Cleary J, Witten I. Data Compression Using Adaptive Coding and Partial String Matching[J]. IEEE Transactions on Communication, 1984, 32(4): 396-402.

[4] Witten I, Neal R. Arithmetic Coding for Data Compression[J]. Communications of the Association for Computing Machinery, 1987, 30(6): 520-541.

[5] Knuth D E, Moore R W. An Analysis of Alpha-beta Pruning[J]. Artificial Intelligence, 1975, 6(4): 293-326.

[6] Svensson A. Computer Forensics Applied to NTFS Computers[D]. Stockholm, Sweden: Stockholm's University, 2005.

[7] Boyd C, Forster P. Time and Date Issues in Forensic Computing——A Case Study[J]. Digital Investigation, 2004, 1(1): 18-23.

编辑　顾逸斐

（上接第 175 页）

4　结束语

利用 AVL 搜索树的查询与更新(即插入与删除)的最大时间复杂度都保持在 $O(\mathrm{lb}n)$量级是本方案较其他方案的特点，是对 CRT 的最大改进。本方案与 CRTBSHT 一样，其节点数比 CRT 方案少一半，CRT 方案中插入与删除会导致整个树的重新计算，本文提出的 CRAVLST 方案仅需调整相关路径上的节点。而与 2-3CRT 方案相比，CRAVLST 方案为二叉树结构简单，2-3CRT 树的结构相对复杂，且克服了 CRTBSHT 方案及文献[2]给出的证书吊销解决方案中有的叶节点到根节点路径可能很短，有的则可能很长的问题。该方案在查询与更新时最大时间复杂度始终保持在 $O(\mathrm{lb}n)$量级。

正如文献[1]指出的，随着公钥密码体系越来越广泛的使用，CA 发行证书的管理问题以及证书的吊销问题将成为公钥基础设施(PKI)中日益重要的课题。构造快速、高效的证书的管理与证书的吊销机制具有重要意义。CRAVLST 方案较与其他方案相比可以说是一种较完善的方案。

参考文献

[1] 王尚平, 张亚玲, 王育民. 证书吊销的线索二叉排序 Hash 树解决方案[J]. 软件学报, 2001, 12(9): 1342-1359.

[2] 贾续摘, 王彩芬, 于成称, 等. 自根向下压缩的二叉排序证书吊梢树方案[J]. 计算机工程, 2007, 33(17): 181-183.

[3] Sartaj S. 数据结构、算法与应用[M]. 孙晓东, 译. 北京: 机械工业出版社, 2000.

编辑　陈　文