基于FAT表重定向的文件隐藏

王玉龙 李清宝 王 炜 牛小鹏

(解放军信息工程大学 郑州 450001) (数学工程与先进计算国家重点实验室 郑州 450001)

摘 要 文件隐藏技术是一种重要的数据安全防护手段。FAT32文件系统具有良好的兼容性,被广泛用于基于Flash芯片的移动存储设备。在分析FAT32文件系统文件组织方式及存储特点的基础上,提出一种通过擦除、转移存储目标文件目录项对文件FAT表项进行重定向改写的方法来实现文件隐藏。理论分析与实验结果表明,该方法具有隐藏容量大、隐蔽性好、鲁棒性强、安全性能出色等特点,但对于大文件执行效率相对较低。

关键词 文件系统,文件隐藏,FAT表,重定向

中图法分类号 TP309.3 **文献标识码** A

File Hiding Based on FAT Redirection

WANG Yu-long LI Qing-bao WANG Wei NIU Xiao-peng

(PLA Information Engineering University, Zhengzhou 450001, China)

(State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China)

Abstract File hiding technology is a kind of significant data security protection method. The FAT32 file system is a good file system for compatibility, and widely used in removable storage service based on flash memory. By analyzing the pattern of files regulation and the features of storage of FAT32, we proposed a new method, which realizes file hiding by erasing and transferring the directory entry of destination files to rewrite the FATs through redirection technology. Theoretical analysis and experimental results show that this method has the characteristics of large capacity, good concealment, strong robustness and excellent security, but low efficiency in hiding large files.

Keywords File system, File hiding, File allocation table(FAT), Redirection

基于Flash芯片的移动存储设备,如U盘、SD卡等,已成为一种被广泛使用的数据交换设备,设备中数据安全性问题日益突出。为保证数据的安全性,通常采用数据加密方法提高数据抵御攻击的能力。但加密数据会引起攻击者更大的兴趣,反而增加被攻击的风险。信息隐藏技术有效解决了这一问题。采用信息隐藏技术能够实现个人重要数据对他人的透明化,通过降低曝光率达到保护数据的目的。

信息隐藏技术通常需借助图像、视频[2]等大容量数据作为宿主,实现信息的隐藏。移动存储设备中数据具有流动性大、随机性强的特点,为待隐藏的信息找到合适的宿主文件较为困难。因此,传统的信息隐藏技术并不适用于移动存储设备数据的隐藏。基于文件系统的文件隐藏技术利用文件系统自身的特点,实现文件的隐藏,能够摆脱对宿主数据的依赖,具有隐藏容量大、鲁棒性强等特点,更适合移动存储设备。

许多人在该领域进行了有价值的研究,并取得一定的研究成果,可归纳为如下几类:(1)对文件内容直接进行隐藏。如:James C Foster[3]提出利用文件系统中已存在的文件的冗余扇区进行信息隐藏的方法,该方法具有较强的隐蔽性,但鲁棒性较差,容量较小,极易受到文件活动的影响。(2)通过过滤驱动对文件访问操作进行控制,实现文件隐藏。如:张慧、吴春欢等[4]提出一种基于磁盘伪装技术的信息隐藏方法,该方法通过改写分区信息在已有磁盘分区中挂载一个隐藏分区,通过添加过滤驱动对磁盘进行访问控制,实现磁盘隐藏;何耀彬、李祥和等[5]通过修改驱动堆栈单元的结构和完成例程,配合修改I/O请求包的传递方法,实现2种驱动级文件隐藏,使操作系统无法查询,也不能通过正常途径访问。这类方法隐蔽性较强,容量较大,但需有操作系统驱动程序进行访问控制,因此更适用于服务器的磁盘阵列,在移动存储系统中并不适用。(3)修改文件在文件系统中的关键线索信息,实现文件的隐藏。如:文献[6]基于FAT32文件系统通过将目标文件目录项属性调整为卷标属性,实现了文件的隐藏;在该技术的基础上,蔡风华[7]提出对目录项中首簇号信息进行修改的方法,该方法一定程度上提高了文件的隐蔽性,但并未对文件的FAT表进行处理;袁杰、江祖敏[8]提出了重构文件FAT表项链的方法,该方法提高了文件被恢复的难度,但对文件的FAT表进行混淆的算法复杂,运算量极大。这一类方法既不需对文件内容进行操作,也不需专门的驱动层程序配合,效率较高,适用范围较广,但隐蔽性较差。

本文在分析FAT32文件系统文件组织方式的基础上,提出一种基于FAT表项重定向的文件信息隐藏(File Hidden

到稿日期:2013-06-13 返修日期:2013-10-15 本文受信息工程大学未来基金(1201)资助。

王玉龙(1990—),男,硕士生,主要研究方向为网络信息安全,E-mail:pldhero2011@hotmail.com;李清宝(1967—),男,教授,博士生导师,主要研究方向为计算机系统结构、信息安全与可信计算;王 炜(1975—),男,博士,讲师,主要研究方向为计算机系统结构、信息安全与可信计算;牛小鹏(1983—),男,博士生,主要研究方向为网络信息安全。

Based on FAT Entries Redirection, FHFR)技术。FHFR通过转移存储文件目录项实现文件隐藏,利用FAT表重定向修改提升了文件还原难度,提高了隐蔽性。本文第1节分析FAT32文件系统的组织方式与特点;第2节着重介绍FHFR的原理、实现算法及恢复过程等内容;第3节通过实验与理论分析,讨论FHFR的性能;最后对本文进行小结,并对下一步工作进行展望。

1 FAT32文件系统组织方式

FAT32文件系统[1,9]是目前使用最为广泛的文件系统之一,具有良好的兼容性与易用性。Windows、Linux、Unix等操作系统及嵌入式系统对其都有很好的支持。该文件系统采用文件分配表来组织管理磁盘的数据空间,每32位作为一个FAT表项,故称为FAT32文件系统。

在FAT32文件系统中,最小的数据单位为扇区(Sector),默认大小为512字节;若干连续扇区组成一个簇(Cluster),是数据存储的最小单位。FAT32文件系统由引导记录区及保留区、FAT表1、FAT表2(表1备份)及数据区组成,如图1所示。引导记录区记录着一些文件系统的关键信息,如根目录首簇号、每个扇区大小、每个簇包含的扇区数等。FAT表由FAT表项组成,每个表项对应数据区的一个簇,每个表项为32位,且每个表项都有一个固定的编号对应数据区的一个簇,表项的不同值代表不同含义。数据区主要存放数据信息,是文件系统的主要区域。

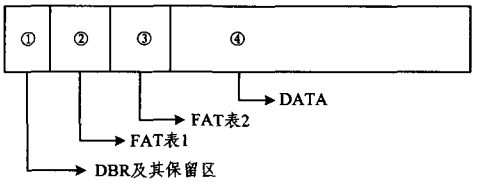


图1 FAT32文件系统结构示意图

为实现对文件的高效管理,FAT32文件系统为每个文件和目录(即文件夹)都分配了一个目录项,用于描述文件或文件夹的属性、大小、起始簇号、创建或修改时间、文件名(或目录名)等关键信息。FAT32的根目录存放在指定的根目录区,根目录位于数据区。通过分析文件的存储路径,逐级查找子目录即可获得目标文件的目录项。目录项中只登记文件实际存储的起始簇号。文件在数据区进行存放,并在FAT表中磁盘簇多对应的表项中记录下该簇所对应的下一簇的簇号,通过这种方式来实现文件的链式存储。图2给出了一个两级路径下的文件访问示意图。目标文件为"tst. txt",存储路径为"G:\wyl\abc\tst. txt"。

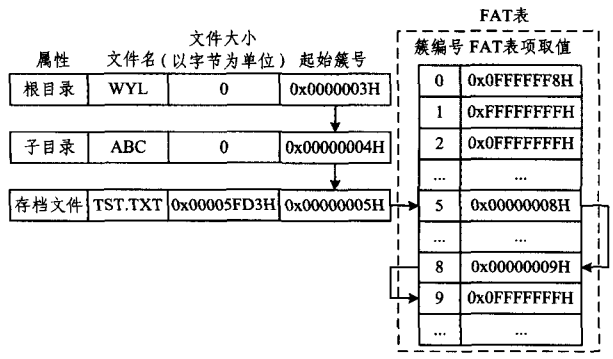


图2 文件访问示意图

2 基于FAT表项重定向的文件隐藏

2.1 FHFR原理

从FAT32的文件组织形式可以看出,FAT32文件系统主要采用链式结构对文件进行管理,其两个关键之处为:目录项管理及FAT表维护。目录记录了文件的存储起始位置、属性等关键信息,而FAT表维护了文件的链式结构。显然,不挪动目标文件在数据区存放的位置,仅修改文件目录项即可实现文件隐藏。但这种隐藏策略隐蔽性较差,通过对目录区目录项进行盲检测即能有效发现异常目录项的存在,进而发现被隐藏的文件。为有效解决上述问题,FHFR通过文件目录项转移存储,增强隐藏文件抵抗目录项盲检测的能力。

具体过程如下:(1)依据文件访问原理,逐级解析文件路径,直到找到文件目录项;(2)查询FAT表,寻找合适的空白簇,将文件目录项内容转移存放到该簇对应的数据区中;(3)擦除文件在目录区的目录项及相关信息。由于文件的目录项已不在目录区存放,因此可有效抵抗盲检测。

FHFR通过对FAT表进行重定向修改,进一步提高隐藏文件的隐蔽性与安全性。具体过程如下:(1)依据文件首簇号,定位在文件FAT中的起始位置;(2)从FAT表中逐项查找文件的簇链,并转移存储簇链,方式与目录项的转移存放一致;(3)对原文件的簇链进行逐项的重定向改写,即将FAT表项随机改写为其它簇编号。修改后的FAT表项不再指向下一个簇。FAT32文件系统依靠FAT表记录文件的簇链,且为单向。若簇链从某处断开,则该簇之后的文件将无法找到。因此,在原簇链结构未知的情况下,由被重定向修改的FAT表恢复出原文件,极其困难。图3为目标文件进行FAT表重定向修改的示意图。

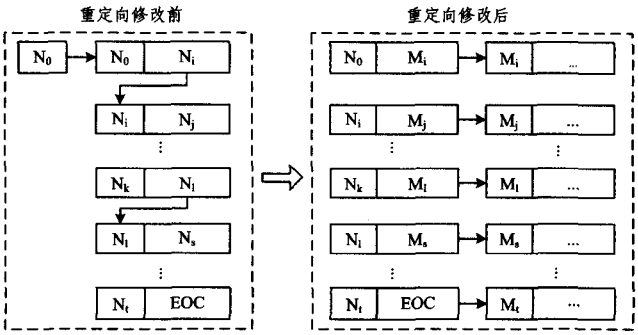


图3 FAT表重定向示意图

2.2 FHFR的算法实现

利用FHFR技术实现文件隐藏,主要分为两个阶段。第一阶段主要实现路径解析,转移存储文件目录项,擦除文件目录项。第二阶段实现FAT表查找,记录目标文件在FAT表中的簇链结构,对FAT表进行重定向修改。

为更好实现文件隐藏,减小被基于内容的取证工具发现的可能,进行文件隐藏前,需对具有明显特征的目标文件采用加密、压缩等手段进行去特征化处理,并使用处理后的文件替换源文件。

表1所列为第一阶段的具体算法实现过程。主要完成逐层解析文件路径,获取文件的目录项,解析目录项获取目标文件的首簇号,记录目标文件的目录项信息。

表 1　第一阶段算法实现

```
Input:文件路径 fileStrPath;
Output:目标文件的目录项 tFileDirEntry、首簇号 tFile1stClstrNum 及所在子
       目录的首簇号 1stClstrNumOfSubFdt;
具体步骤:
(1)pathResult=逐级分割 fileStrPath,记录路径级数 pC;
(2)读取分区的 DBR,获取 1stClstrNumOfRoot,fatNum,sectorNumOfFat,
   ResSector,sectorPerClstr 等参数;
(3)获取 sectorNumOfRoot,目标区域→根目录区;
(4)FOR i=1 TO pC
(5)     依据 pathResult,检索目标区域;
(6)     IF 发现与 pathResult 匹配的字符串
(7)     THEN 获取 fileDirEntry;
(8)          IF 目录项属性为目录
(9)          THEN 获取 1stClstrNumOfSubFdt;
(10)               计算下一级子目录位置;
(11)               目标区域→下一级目录区;
(12)               ELSE IF 属性为文件 & 与目标文件名一致
(13)          THEN 获取 tFile1stClstrNum;
(14)               tFileDirEntry=fileDirEntry;
(15)               fileDirEntry=0;
(16)               BREAK;//跳出循环
(17)                    END IF
(18)          END IF
(19)     ELSE 返回"ERROR!! 未发现匹配路径";
(20)          BREAK;//跳出循环
(21)     END IF
(22)END FOR
(23)加密并转移存储 tFileDirEntry;
(24)返回 1stClstrNumOfSubFdt;
```

表 2 所列为第二阶段的具体实现过程。主要完成定位 FAT 表,在 FAT 表中跟踪查找文件簇链,并转移存储簇链结构,以及对 FAT 表进行重定向修改。

表 2　第二阶段算法实现

```
Input:目标文件的首簇号 tFile1stClstrNum;
Output:簇链存储区域的簇号 clstrNum;
具体步骤:
(1)startSectorNumOfFat1=ResSector,目标区域→FAT1;
(2)求参数 sectorInFat1,offsetInSector;
(3)FOR i=1 TO SIZE
(4)     copyFatChain[i]=TEMP;
(5)          TEMP = 随机选择{0x00000002H to 0x0FFFFFFEH,
             0x0FFFFFFFH};//对簇链进行随机重定向
(6)     IF copyFatChain[i]!=0xFFFFFFFFH
(7)     THEN offsetInSector=(copyFatChain[i]*4)/512;
(8)          IF sectorInFat1!=32+(copyFatChain[i]*4)/512
(9)          THEN
(10)              sectorInFat1=32+(copyFatChain[i]*4)/512;
(11)              依据 sectorInFat1,跳转到新扇区;
(12)         END IF
(13)    END IF
(14)END FOR
(15)检索空白簇 tClstr,将 copyFatChain 写入 tClstr 对应数据区,置 tClstr 对
    应的 FAT 表项为 0x0FFFFFF7H,可将 copyFatChain 与 tFileDirEntry
    存放在同一区域;
(16)定位 sectorInFat2,修改 FAT2;
(17)返回 tClstr;
```

2.3　隐藏文件的恢复

对已隐藏的文件进行恢复是文件隐藏技术的重要一环。FHFR 的恢复过程与隐藏过程相对应。文件恢复算法与文件隐藏算法实现方式相似,过程相反,因此不再赘述。此处只给出基本实现思路。依据文件恢复位置,可将恢复方案分为文件原始位置、根目录下及指定位置下 3 种。

第一种方案,若要求源文件必须恢复到原目录下,则恢复过程需要两个参数 *tClstr*(簇链存放位置)与 *1stClstrNumOfSubFdt*(源文件所在目录的簇号)。

恢复的基本过程如下:

(1)获取 *tFileDirEntry*。已知参数 *tClstr*,可计算出簇编号为 *tClstr* 对应的数据的首个扇区,检索簇编号为 *tClstr* 的数据区,获得 *tFileDirEntry*。

(2)恢复目录项。通过 *1stClstrNumOfSubFdt* 可计算出原 *tFileDirEntry* 所在目录对应簇的首个扇区编号,将 *tFileDirEntry* 恢复到该簇空白处。

(3)修正 FAT 表。检索簇编号为 *tClstr* 的数据区,读取簇链信息,依次恢复对应的 FAT 表项,直到遇见尾簇标志 0x0FFFFFFF。释放编号为 *tClstr* 的簇。

隐藏文件所在目录对应的簇未释放是上述恢复过程能够成功执行的前提。造成对应簇被释放的原因可能是目录被转移或删除。一旦被释放,则无法通过 *1stClstrNumOfSubFdt* 将文件恢复到原目录下,因为子目录已不在该簇中。

第二种方案,若对源文件还原位置无要求,可将文件直接恢复到根目录下,则只需参数 *tClstr* 即可完成:在步骤(2)中直接将由步骤(1)获得的 *tFileDirEntry* 复制到根目录区的空白处。步骤(1)、(3)与上述方法相同。则隐藏文件被恢复到了根目录下。

第三种方案,将文件恢复到指定位置。除了参数 *tClstr* 外,还需提供文件恢复路径参数 *fileStrPath*。在恢复文件时,需要调用路径解析算法(算法实现参考表 1 第一阶段算法实现),通过参数 *fileStrPath* 获取新的目的地址目录首簇号参数 *1stClstrNumOfSubFdt*,通过该参数恢复目录项。

3　实验与性能分析

隐蔽性、隐藏容量、鲁棒性及安全性是衡量文件隐藏技术优劣的 4 个关键指标。隐蔽性是指文件隐藏之后被发现或被曝光的可能性,隐藏容量直观体现为可隐藏文件的大小;鲁棒性是指被隐藏的文件不因文件载体变动而导致无法恢复的能力;安全性是指隐藏文件抵抗攻击者攻击的能力。

为便于讨论,将基于冗余扇区收集的方法[3]称为方法一,将基于目录项内容修改的方法[6,7]称为方法二。对隐蔽性、隐藏容量及鲁棒性进行了实验比较,对安全性、恢复准确率及磁盘开销进行了理论分析。为保证实验具有说服力,以文件大小、类型不尽相同的 11 个文件(见表 3)作为目标测试文件;针对所关心的各项性能指标,采用相应实验方法对目标测试文件进行操作,考察相关指标。具体实验方案在下文中具体阐述。

表 3　目标测试文件列表

文件名	文件类型	文件大小(kB)
tst1	.txt	6
tst2	.doc	18
tst3	.doc	53
tst4	.xls	158
tst5	.jpeg	784
tst6	.jpeg	1106
tst7	.pptx	4412
tst8	.rar	14024
tst9	.rar	30271
tst10	.rar	92673
tst11	.rar	321980

实验环境配置为 Pentium(R) Dual-Core CPU E5400 @ 2.7GHz,2.00G 物理内存,Windows XP SP3 操作系统。

3.1 隐蔽性

本文采用取证工具对隐藏文件所在文件系统的文件隐蔽性好坏进行取证考察。X-Ways Forensics(简称为 XWF 工具)是由德国著名软件公司 X-Ways 公司开发的,是具有较高认可度与权威性的电子取证工具。本文采用的版本为 X-Ways Forensics V13.0。

具体方案:首先,对不同类型目标测试文件分别采用方法一、方法二及 FHFR 进行隐藏操作;然后,对目标测试文件所在磁盘采用 XWF 工具进行相关取证操作(具体取证操作如表 3 所列),并查看测试结果;最后,对文件进行恢复操作。针对不同目标测试文件,重复上述操作 5 次(取证结果见表 4,Yes 表示取证成功,即通过取证能够发现文件存在的痕迹,No 则反之)。

"全局匹配搜索"是将目标测试文件中的某些字段(或编码)作为关键字全局逐扇区进行搜索;"显示全部文件"是利用通常磁盘中文件目录项不会被删除的原理,对磁盘中目录区所有文件进行整理罗列,并给出文件完整评估结果;"文件哈希校验"利用哈希算法为每个文件都创建一个哈希值,通过目标测试文件对比隐藏前后其它文件的哈希值,来判断文件是否被修改;"属性筛选"是磁盘中对特定类型文件进行筛选。

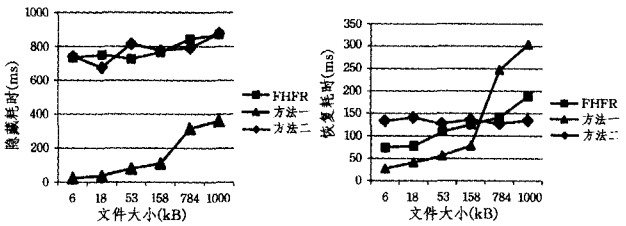
从实验结果来看,3 种方法都无法躲避全局匹配搜索取证。但方法一与 FHFR 都对文件在内容上进行了离散化处理,故即使发现文件的部分片段,也无法以此作为线索发现文件的其它部分,且通过对文件内进行简单加密即可实现关键字的去特征化,可进一步提高隐蔽性。由于方法一严重依赖宿主文件,目标测试文件一旦隐藏成功,一定会改变磁盘中其余某些文件的哈希值,因此,通过比较所有文件前后哈希值的变化,极易发现隐藏信息的存在。由于属性筛选需要从目录区定位文件,比较文件的类型签名,若方法二对目录项中首簇号信息进行处理,则无法被属性筛选取证,否则会被成功找到。

通过上述实验不难发现,相比方法一、方法二,FHFR 技术在文件哈希值校验、显示全部文件等取证手段方面具有更为良好的表现,尽管在全局匹配搜索中文件中包含关键字的某些扇区可能会被发现,但其它内容还能获得很好保护。

3.2 隐藏容量与执行效率

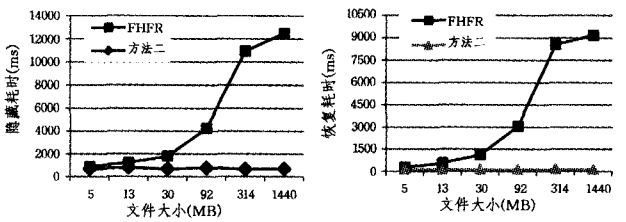
可隐藏文件的大小是隐藏容量的直观反映;执行效率的高低通过执行文件隐藏及执行文件恢复的耗时表征。本文对文件大小递增的不同目标测试文件进行隐藏与恢复操作,并记录文件大小与恢复时间。为保证数据真实性,每个文件测试 10 次,取平均值作为最终测试结果。图 4 所示为对小文件(小于 1MB)进行隐藏与恢复的实验结果;图 5 所示为对较大文件(超过 1MB)进行隐藏与恢复的实验结果。

方法一将目标文件切割为若干片段,存放在已有文件的冗余扇区中,因此隐藏容量与文件系统中的文件数量密切相关。目标文件体积越大,所需的宿主文件越多,耗时越长。当文件体积超过 1M 时,很难找到数目足够多的宿主文件存放目标文件。当目标文件达到 5M 时,在目标分区下无法找到足够多的宿主文件,因此无法被成功隐藏。方法二的执行耗时与文件大小无明显联系,这是因为方法二只对文件目录项进行修改,文件目录项大小是固定的,故采用方法二无论文件大小,在耗时上都差别不大,隐藏容量较大。采用 FHFR 对小文件进行隐藏与恢复,耗时与方法二区别不大。对于大文件,随着文件体积变大,文件隐藏与恢复耗时增加。采用 FHFR 技术最大可实现 GB 级文件的隐藏,但耗时较长(10s 以上),隐藏容量与方法二一致。



(a) 文件隐藏　　(b) 隐藏文件恢复

图 4　对小文件进行隐藏与恢复的实验结果



(a) 文件隐藏　　(b) 隐藏文件恢复

图 5　对大文件进行隐藏与恢复的实验结果

图 6 所示为利用 FHFR 技术实现文件隐藏与恢复耗时与文件大小的关系曲线,由图可知耗时与文件大小正相关。文件越大,曲线斜率越大,增长趋势越明显。这是由于 FHFR 不但需要操作文件的目录项,还需对 FAT 表进行记录、转存、重定向修改等操作,文件越大,所需操作工作量越大。对小文件进行隐藏与恢复时,所需时间相对较短,文件越大,正相关性越显著。

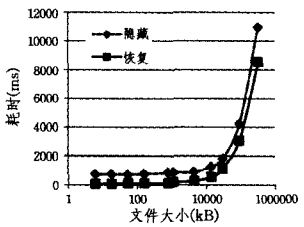


图 6　FHFR 文件隐藏与恢复过程文件大小与耗时关系走势图

3.3 鲁棒性

鲁棒性主要体现为被隐藏的文件抵抗磁盘操作的能力。对磁盘信息的操作可分为如下几类:1)对文件内容的操作,主要包括对某个文件内容的读、增、删等操作;2)文件重命名操作,是指对文件目录项内容进行修改的操作,包括对单个文件在磁盘内进行剪切、移动、属性修改、移入到垃圾站等操作;3)对整个文件的操作,包括文件的复制、创建、彻底删除等;4)对整个文件系统的操作,主要包括磁盘碎片整理、磁盘格式化等。

为检测 3 种方法抵抗上述操作的有效性,采取如下实验方法:首先,对目标测试文件进行隐藏;然后,对目标测试文件所在磁盘(或其中的文件)相应操作多次;最后,对文件进行恢复,并检测恢复文件的完整性。每项过程重复测试 3 次。结果如表 4 所列。

表 4　XWF 取证结果

取证操作	方法一	方法二	FHFR
全局匹配搜索	Yes	Yes	Yes
显示全部文件	No	Yes	No
文件哈希校验	Yes	No	No
属性筛选	No	Yes/No	No

表 5 中，C 表示具有完全抵抗力，P 表示具有部分抵抗力，N 表示无抵抗力。从实验结果不难看出，方法一的鲁棒性最差，这是由于方法一严重依赖于宿主文件，任何对于宿主文件的操作都会对隐藏文件产生影响。宿主文件一旦被修改、删除，隐藏文件的完整性就受到严重影响，从而导致无法恢复。方法二具有较好的鲁棒性，能够完全抵抗类型 1、类型 2、类型 3 及磁盘碎片整理操作，但是无法抵抗格式化操作。

表 5　鲁棒性测试结果

操作	方法一	方法二	FHFR
类型(1)	N	C	C
类型(2)	C	C	C
类型(3)	P	C	C
碎片整理	N	C	C
格式化	N	N	P

FHFR 不仅能够抵抗前 3 类低烈度操作与磁盘碎片整理操作，而且对快速格式化具有一定的抵抗能力。这是因为快速格式化操作只是对文件系统的关键信息进行重新填写，如 MBR、FAT1、FAT2、根目录区等，对数据区的数据并不进行擦除。FHFR 转存了原文件系统的 FAT 表项及目录项，只要格式化后的文件系统与原文件系统一致，通过重新恢复文件目录项，修正 FAT 表就可以完全恢复出隐藏文件。因此，3 种方法中 FHFR 具有更好的鲁棒性。

3.4　安全性分析

文件隐藏的安全性主要体现在两个方面：(1)有效抵抗盲检测的能力；(2)隐藏文件被攻击者恢复的可能性。对于(1)所涉及内容，事实上与隐蔽性检测的目标相一致，不再赘述。此处重点讨论隐藏文件被恢复的可能性。

定义 1　在 FAT32 文件系统下，某簇被标记为占用，但不属于任何目录或文件，则称该簇为异常簇，用 N 表示。对于长期使用的文件系统，异常簇十分常见，多由异常操作导致。被重定向修改的文件簇，其特征与异常簇相同。

定义 2　异常簇中，非由异常操作导致而是由 FHFR 重定向修改产生的异常簇称为伪异常簇，用 P 表示；反之，称为真异常簇 T。

定义 3　从异常簇中成功恢复隐藏文件的概率称文件恢复率，用 PR 表示。可用回复率来表征文件的安全性。显然，恢复率越低，安全性越好。

假设某 FAT32 文件系统下，攻击者能够找出所有 N 异常簇，但不知哪些簇属于隐藏文件，且由于隐藏文件被重定向修改，因此无法获得正确簇排列顺序。只能通过从异常簇中随机挑选 n 个异常簇，恰好为全部伪异常簇，为事件 A；对 n 个簇进行排列组合，恰好与原文件顺序一致，为事件 B。显然，事件 A 与事件 B 相互独立。

其中：

$$P(A)=\frac{1}{C_N^1+C_N^2+\cdots+C_N^N}=\frac{1}{\sum_{n=1}^{N}C_N^n} \tag{1}$$

$$P(B/A)=\frac{1}{n!} \tag{2}$$

则由全概率公式可得，

$$\begin{aligned}PR&=P(B)=P(A)\times P(B|A)\\&=\frac{1}{\sum_{n=1}^{N}C_N^n}\cdot\frac{1}{n!}=\frac{1}{\sum_{n=1}^{N}(C_N^n\cdot n!)}\\&=\frac{1}{\sum_{n=1}^{N}A_N^n}=\frac{1}{A_N^1+A_N^2+\cdots+A_N^N}\end{aligned} \tag{3}$$

由式(3)可得如下结论：

(1)PR 只与 N 有关，即 PR 只与文件的异常簇的数目有关，且 N 越大 PR 越小，意味着安全性也就越高；

又因为

$$N=P+T \tag{4}$$

所以：

(2)对于大文件 $T\ll P$，$N\approx P$，则 PR 与伪异常簇的数量相关；

(3)对于小文件 $T\gg P$，$N\approx T$，则 PR 与真异常簇的数量 T 相关。

下面，对 PR 值的大小进行估计。

因为

$$\frac{1}{N\cdot A_N^N}<PR<\frac{1}{A_N^N} \tag{5}$$

且 $A_N^N=N!>2^{N-1}$，所以

$$PR<\frac{1}{2^{N-1}} \tag{6}$$

由式(5)不难看出，若某文件系统异常簇数目约为 200，则 $PR<6.3\times10^{-60}$，PR 数值极小，这意味着文件被重定向修改后，极难被恢复。事实上，这只相当于一个约 1.5MB 文件的大小。

3.5　恢复准确率

隐藏文件成功恢复是文件隐藏的最基本要求。FHFR 通过对文件目录项转存、擦除，对 FAT 表项进行转存、重定向修改，实现文件隐藏。隐藏过程并不对文件数据区内容进行处理，因此降低了内容完整性受到侵害的风险。恢复过程中，只需两个参数 *tClstr* 与 *1stClstrNumOfSubFdt*(或指定恢复路径 *fileStrPath*)即可实现文件的恢复(详细参考本文 2.3 节)，所需保存的线索信息较少，因此具有较高的恢复准确率。

某些极端条件下，FHFR 仍存在文件无法被恢复的可能。在实际运用过程中发现，导致无法恢复的状况与原因可分为两种：

(1)由于文件所在目录被移除，导致文件无法恢复到原位置。事实上，尽管文件无法恢复到原位置，但文件保持完好，依然可以将文件还原到根目录下或指定位置(参考本文 2.3 节)，因此属于可修正的恢复失败。

(2)磁盘快速格式化、低级格式化、磁盘修复等磁盘全局性操作或其他因素导致文件完整性破坏，而无法被恢复(参考 3.2 节)。磁盘全局性操作，对磁盘中所有文件数据的存放位置、数据完整性及数据组织形式具有极大的影响，其中一些操作(如低级格式化)甚至会对磁盘全数据造成不可逆的损坏。此种情形下，FHFR 隐藏文件具有与普通文件相同的风险。因低级格式化引起的失败为不可修正的恢复失败。

但经快速格式化操作的磁盘上，被隐藏的目标测试文件仍然有可能被恢复，但须满足 3 个前提：(1)经快速格式化前

后文件系统规格一致；(2)与隐藏文件相关的扇区未被使用；(3)隐藏文件大小不能超过 1MB(设每簇大小为 16 个扇区，每扇区 512B)。满足上述条件也属于可修正的恢复失败。由于 FHFR 实现文件隐藏的过程会产生伪异常簇，文件越大，产生异常簇越多。这可能会引起操作系统主动申请调用磁盘维护程序对磁盘进行修复操作，从而影响文件的恢复准确率。因此，FHFR 对大文件进行隐藏与恢复操作时具有更高的风险。

表 6 中示出了本文在隐蔽性测试、隐藏容量与执行效率测试及鲁棒性测试中，FHFR 准确恢复与恢复失败的结果。由于在鲁棒性测试中，50%的测试过程中包含高风险的全局性操作，因此鲁棒性测试中恢复率较低。在实际运用过程中，磁盘全局性操作的使用率极低，因此 FHFR 的实际恢复率远远高于表中的 92.1%。

表 6 历次试验中 FHFR 恢复成功与失败的统计结果

	一次成功	一次失败		可修正失败		成功率
		原因一	原因二	原因一	原因二	
隐蔽性测试	220	0	0	0	0	100%
隐藏容量与效率测试	110	0	0	0	0	100%
鲁棒性测试	132	6	60	6	18	78.8%
合计	462	6	60	6	18	92.1%

3.6 磁盘开销分析

对原文件的目录项，FAT 表链进行转移存储是实现文件隐藏、提高文件安全性与隐蔽性的关键步骤。文中算法实现是将目录项与 FAT 表转移存放在原文件系统数据区之下，因此需要额外的磁盘开销对其进行存放。

假设，FAT32 文件系统中，每簇大小为 16 个扇区，每扇区 512B，则每簇大小为 8kB，那么额外所需开销为 1/2048，因此，当文件小于 1M 时，所需额外开销小于 512B。FAT32 中单个文件最大为 4GB，故额外空间最大需 2MB。

结束语 本文针对 FAT32 文件系统的目录项与 FAT 表两个关键点，提出了通过擦除、转移存储目标文件目录项对文件 FAT 表项进行重定向改写的方法来达到对目标文件的深度隐藏之目的，并将其命名为 FHFR 技术。FHFR 只需较少的额外存储空间用于目录项及 FAT 表项，即可获得较好的隐藏效果。

实验与分析结果表明，在隐蔽性方面，FHFR 能够有效抵抗多种磁盘取证手段的检测；FHFR 不受宿主文件数量大小的限制，最大可隐藏 GB 级文件，但耗时较长；在鲁棒性方面，FHFR 不仅能够有效抵御多类低烈度的文件操作的影响，而且在能够有效抵抗针对整个文件系统进行的高烈度操作对隐藏文件的影响；同时，FHFR 具有良好的安全性与恢复准确率。

参 考 文 献

[1] Microsoft Corporation. Microsoft extensible firmware initiative FAT32 file system specification[S]. FAT: general overview of ondisk format(Version 1.03). December 2000

[2] Lin X, Li Q, Wang W. Information Hiding Based on CAVLC in H.264/AVC Standard [C]// Proceeding of Multimedia Information Networking and Security (MINES), IEEE. Nanjing, China, 2012: 900-904

[3] Burton B J. How to hide file in slack file space with Slacker.exe [EB/OL]. www.jbbrowning.com, 2013.4

[4] Zhang Hui, Wu Chun-huan, Niu Xia-mu, et al. A Disk Disguising and Hiding Method[C]// Convergence and Hybrid Information Technology, Proceedings of Busan. Nov 2008: 520

[5] 何耀彬，李祥和，孙岩. 基于驱动堆栈单元的文件隐藏方法[J]. 计算机工程，2011，37(13)：9-12

[6] 华中科技大学. 基于 FAT32 磁盘文件系统结构的文件隐藏方法[P]. ZL03118544.4. 中国，2003：1-9

[7] 蔡风华. 基于 FAT32 文件系统的文件隐藏研究与实现[D]. 武汉：华中科技大学，2007

[8] 袁杰，江祖敏. 基于 FAT32 的文件隐藏方法及在 Linux 上的实现[J]. 电子设计工程，2012，20(13)：15-18

[9] 刘伟. 数据恢复深度揭秘[M]. 北京：电子工业出版社，2010

(上接第 106 页)

参 考 文 献

[1] Kesner S B, Howe R D. Design Principles for Rapid Prototyping Forces Sensors Using 3-D Printing[J]. IEEE/ASME Transactions on Mechatronics, 2011, 16(5): 866-870

[2] Zafeiriou S, Tefas A, Pitas I. Blind robust watermarking schemes for copyright protection of 3D mesh objects[J]. IEEE Transactions on Visualization and Computer Graphics, 2005, 11(5): 596-607

[3] 李辉，侯义斌，黄樟钦，等. 一种智能攻击模型在 RFID 防伪协议中的研究[J]. 电子学报，2009，37(11)：2565-2573

[4] Kim K, Barni M, Tan H Z. Roughness-adaptive 3-D watermarking based on masking effect of surface roughness [J]. IEEE Transactions on Information Forensics and Security, 2010, 5(4): 721-733

[5] 谢磊，殷亚凤，陈曦，等. RFID 数据管理：算法，协议与性能评测[J]. 计算机学报，2013，36(3)：457-470

[6] Aliaga D G, Atallah M J. Genuinity Signatures: Designing Signatures for Verifying 3D Object Genuinity[J]. Computer Graphics Forum, 2009, 28(2): 437-446

[7] Johnson A. Spin-images: A representation for 3D surface matching[D]. CMU, Robotics Institute, 1997: 1-288

[8] 杨育彬，林珲，朱庆. 基于内容的三维模型检索综述[J]. 计算机学报，2004，27(10)：1297-1310

[9] Besl P J, Mckay N D. A Method for Registration o f 3D Shapes [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992, 14(2): 239- 256

[10] Halma A, ter Haar F, Bovenkamp E. Single spin image-ICP matching for efficient 3D object recognition[C]// Proceedings of the ACM workshop on 3D object retrieval. ACM, 2010: 21-26

[11] Johnson A E, Hebert M. Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes[J]. IEEE Trans: Pattern Analysis and Machine Intelligence, 1999, 21(5): 433-449

[12] Johnson A E, Hebert M. Surface Matching for Object Recognition in Complex 3-D Scenes[J]. Image and Vision Computing, 1998, 16: 635-651