available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/diin

Digital Investigation

An adaptive method to identify disk cluster size based on block content

*Ming Xu**, *Hong-Rong Yang*, *Jian Xu*, *Ye Xu*, *Ning Zheng*

Institute of Computer Application Technology, Hangzhou Dianzi University, Hangzhou, PR China

ARTICLE INFO

Article history:
Received 15 January 2008
Received in revised form
3 January 2010
Accepted 6 January 2010

Keywords:
Cluster size
Computer forensics
File carving
Disk
Sector
Entropy difference distribution

ABSTRACT

Indentifying the cluster size based on data content, rather than relying on the meta-data of file system, is an important issue in the field of the disk forensics and file caving. When the file system on an evidence disk has been intentionally or accidentally damaged, it is necessary to indentify the cluster size. This paper presents a method to identify the disk cluster size based on data content for various file systems. The main idea is using the difference between the entropy difference distributions of the non-cluster boundaries and the cluster boundaries to identify the cluster size. The χ^2 statistic is adopted to reveal this difference. Experimental results demonstrate that the proposed approach is effective in identifying the cluster size.

1. Introduction

In a file system, a cluster is the smallest logical amount of disk space that can be allocated to hold a file. In order to reduce the overhead of managing on-disk data structures, the file system does not allocate individual disk sectors, but contiguous groups of sectors, called clusters. Typical cluster sizes range from 1 sector to 128 sectors, and typical sector size is 512 bytes.

When file system has been intentionally or accidentally damaged, identifying the size of disk cluster is a fundamental problem for the computer forensics and file caving.

An example was presented by Reust and Sommers in a capital murder case at AAFS 2008. In this case, a hard drive had been reformatted with a new file system, the core components of the prior file system had been overwritten, and the file system metadata was unavailable. It was necessary to recover file system metadata from a reformatted hard drive in order to locate and reconstruct DNA digital files from a Macintosh computer. The first cluster of the target files was located by keyword searching for a known header signature, and then the second cluster was found by performing a calculation involving the cluster size, the characteristics of the DNA file format, and inspecting each cluster on the hard drive for an expected pattern at the calculated offset. In order to know the number of clusters used by the file, and the offset of an expected pattern in the next cluster, it is necessary to know how many sectors or bytes are in a cluster. In this case, the meta-data information about the original file system was luckily obtained by the Volume Recover tool of Norton Utilities, because the tool had been installed on disk before reformatting. Without the help of Norton Utilities to backup of the original volume, this tool will not be able to restore the file system, and an alternate approach to identify the cluster size is an important step to recover file data from a reformatted disk. Another example of the damaged file system for forensic analysis was documented by Garfinkel and Shelat (2003) during their research. They acquired 158 hard drives on the secondary market from November 2000 to

* *Corresponding author.*
E-mail address: mxu@hdu.edu.cn (M. Xu).
1742-2876/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.diin.2010.01.003

August 2002. 46 hard drives of them could not be mounted due to unavailable file system metadata.

Some storage areas of the volatile data, such as swap space or RAM, may contain file fragments data. Identifying the cluster size or the page size will help us acquire more useful data.

Some anti-forensic technologies have been applied to make computer forensic more difficult by destroying the file system meta-data information (Casey, 2000). For example, the meta-data of a file system has been erased or otherwise altered in order to protect incriminating information (Victor and Richard, 1998). To illustrate this scenario more clearly, a simple case was developed for this work based on our experience in digital forensics. Particularly, we altered the cluster size metadata in an NTFS file system at offset 0x0D within the boot sector, which represents the number of sectors per cluster, and changed the byte at this location from '0x08' to '0x02'. We then created a duplicate of this file system partition, and examined it by forensics tools. The alteration of cluster size in this test scenario prevents the forensic tools from accurately interpreting the file system. For instance, the tool *dls* (from The Sleuth Kit, available at http://www.sleuthkit.org) was used to list the details about data units in the unallocated space in a file system. But in this case, this tool only returned ''Error in metadata structure''. Another tool named Directory Snoop, which is a cluster-level search tool that allows Windows users to analyze and view files in FAT and NTFS file system partition, reported ''Error loading $MFT file for drive *'' when attempt to open this test partition image. Using WinHex specialist version, some metadata was successfully extracted from the test partition image. However, WinHex reported the bytes per cluster as 1024 bytes, which were the altered value and not the original cluster size. Furthermore, using the Create Drive Contents Table option on the Specialist menu (Casey, 2004) to list existing and deleted files and directories were unsuccessful, and nothing was listed.

These examples demonstrate some challenges faced by the digital investigators when dealing with the damaged, corrupted, altered or destroyed file systems. When the file system metadata is unavailable, it is necessary step to identify the cluster size for further analyzing. In general, the cluster size can vary from 512 bytes to 64K bytes in a variety of file systems. In addition, the IDEMA (2007) (International Disk Drive Equipment and Materials Association) recently authorized a new criterion , which summarizes the discussion about increasing the traditional sector size of 512 bytes to 4K bytes.

This paper presents a method of identifying disk cluster size based on data content for different file systems. The proposed method is independent of the file system. Identifying cluster size is also an important technology for file carving, which involve recovering a file from unstructured disk data. The main challenges in file carving are file fragmentation and carving files with missing fragments. In the file carving taxonomy, proposed by Garfinkel and Metz, based on characteristic and based on block carving are two primary approaches, but both take little account of the cluster size (Metz and Mora, 2006). If the cluster size can be identified based on data content, we can use cluster-based carving to replace sector-based carving to improve the efficiency of file carving.

This paper proposes an adaptive method to identify the cluster size based on the content of consecutive sectors without the help of the file system metadata. The main idea is using the difference between the entropy difference distributions of the non-cluster boundaries and cluster boundaries to identify the cluster size. The χ^2 statistic technology is adopted to reveal this difference. Experimental results demonstrate that the proposed approach is effective in identifying cluster size.

This paper is organized as follows: Section 2 describes related work about the disk forensics and file carving; Section 3 discusses the proposed methodology; Section 4 describes our experiments; the discussing is proposed in Section 5; and Section 6 concludes this paper and outlines future work.

2. Related work

Existing work has shown that many discarded hard disks contain confidential and recoverable information (Garfinkel and Shelat, 2003). Unfortunately, to the best of our knowledge, there is little work in literature talking about hard disks with damaged, corrupted, altered or destroyed file systems. Victor and Richard (1998) introduced the tools and techniques for recovering information from physically damaged hard disks. But, special equipment and the most pristine of environments are needed to repair the damaged disk, and the effect was still not perfect.

File carving was the topic of the 2006 and 2007 DFRWS challenges (Carrier et al., 2006, 2007). REVIT assumes that data at a specific location can only be part of one file. It searches every block for every file type definition, and used the smart carving and deep carving techniques to recover file (Metz and Mora, 2006). Practically, if the cluster size can be identified in advance, the cluster-by-cluster carving can be used in their method to enhance the efficiency of their method. A disk cluster classification method, proposed by Veenma (2007), uses entropy of cluster content to design the classifier. The first place in the submissions to the DFRWS 2006 challenge uses the block-wise entropy to detect file boundaries, and to identify blocks which do not belong to the same file (Monroe et al.).

In our work, the entropy difference between neighboring blocks has been used as a basic statistic means, and the χ^2 (Chi-square) statistic has been used to differentiate the difference between the entropy difference distributions of the non-cluster boundaries and the cluster boundaries. The χ^2 statistic is used extensively in the hypothesis testing field, such as simulating standard distribution. In Westfeld and Pfitzmann's (1999) work, a χ^2 statistic was employed to estimate probability of embedding information into image with EzStego.

3. The method to identify the disk cluster size

3.1. *Cluster characteristics*

Before describing our approach, it is necessary to explain the characteristics of a cluster.

A cluster is the basic unit of disk space allocation for files and directories in file systems. In order to reduce the overhead of managing on-disk data structures, the file system does not