Selected Methods of File Carving and Analysis of Digital Storage Media in Computer Forensics

Fryderyk DARNOWSKI[1], Andrzej CHOJNACKI[2]

[1]Doctoral student of Cybernetics Faculty
Military University of Technology
Gen. S. Kaliski 2 St., 00-908 Warsaw, Poland
doktorro@gmail.com

[2] Cybernetics Faculty
Military University of Technology
Gen. S. Kaliski 2 St., 00-908 Warsaw, Poland
andrzej.chojnacki@wat.edu.pl

ABSTRACT: Article describes the most common methods of data recovery in modern forensic science. Typical file carving problems are explained. The methods are compared and their advantages and disadvantages explained. Proposition of new file carving method for NTFS is presented and explained.

KEYWORDS: computer forensics, data recovery, file carving, NTFS, MFT

1. Introduction

Computer (or digital) forensics (CF or DF) is an overall term for actions aimed at securing and examining digital storage media. Computer forensics investigations are usually summarized in the form of a report describing the disclosed material which was of interest to the person or institution ordering the analysis. Nowadays, people are surrounded by electronic devices at an unprecedented scale. Smartphones now resemble computers more than phones. Most of the information is produced and processed digitally, which leads to the increasing importance of digital evidence. Computer forensics tools are also used in civil proceedings. CF science is most frequently used to:

- to provide evidence of crime, to identify perpetrators based on the data found in computers, mailboxes, instant messengers, etc. (digital evidence to use in court),
- to provide evidence of data theft by disloyal employees, to detect sabotage, to evaluate system security, e.g. after hacker attacks (business purposes)
- to recover lost data (personal purposes).
- Overall, CF investigations can be divided into three basic stages [1]:
  - acquisition,
  - analysis,
  - reporting.

The days when one man could search an entire digital storage device (e.g. a 1.44MB floppy disk) are gone forever. Nowadays, specialist software is used to analyze digital media storage content, with EnCase [2] and FTK [3] as the most commonly used programs. X-Ways [4] The SANS Investigative Forensic Toolkit (SIFT) [5] and The Sleuth Kit (+ Autopsy) [6] are especially noteworthy. Sleuth Kit is a set of tools for analysis (operating on Windows and Linux) and SIFT is an operating system based on Linux (Ubuntu) with tools for analysis installed. The basic levels of analysis are presented below [7]:

- *media analysis* – used to analyze data stored on a data storage device assuming the absence of a hierarchy or a file system – as in some recorders (continuous recording),
- *media management analysis* – used especially to analyze RAID arrays and the contents of FLASH memory drives,
- *file system analysis* – used to analyze disk partitioning in order to extract files, even the deleted ones,
- *application analysis* – used to analyze the data inside the file, it draws from the fact that every file format is specific; it is a large enough category so it can be divided into the following subcategories:
  - analysis of the operating system – the system settings, network settings, installed software, authorization, etc.,
  - analysis of programs – applies to both data generated by the application as well as logs recorded by the system; it is especially useful in post break-in analysis,
  - analysis of the multimedia – for example, a disclosed image in itself can constitute evidence (e.g. photo of a document).

The rest of this article outlines the most commonly used methods of analysis at the level of the file or application system. The analysis can be performed on the original data storage device, however, the time required to carry out the analysis (days), and the waiting time before the start of the analysis (weeks, months) make it seem reasonable to produce a copy and send it for analysis whereas the original storage device is returned to the owner.

Verification of the hash is always the first step in the analysis performed on a binary copy. Further steps may vary in order depending on the software used, but usually these are:

a) File signature analysis – search of specific file headers.
b) Hash analysis – when the exact content of the file in question is known, its hash can be calculated and the storage device can be searched in order to identify areas with the same hash value.
c) Keywords analysis – useful when analyzing text documents with known content.
d) Statistical method – search based on statistical analysis.
e) Content analysis – when the exact structure of certain files is known, the storage device can be searched for known patterns.

The above mentioned methods of analysis are the most popular ones (especially signature analysis and hash analysis which are the fastest). There are so many methods of analysis and software solutions that nowadays we often talk of the end of the golden age of computer forensics [8]. The reason for this is the plethora of types of storage devices and file types, which makes many methods of analysis valid only for specific file types. The purpose of this article is to provide information about the current state of the field of methods of analysis. These methods will be outlined and the problems that specialists in computer forensics encounter will be discussed. Later in this article we will refer to drives with NTFS (New Technology File System), however, general assumptions are made regardless of the file system.

1.1. Signature analysis

A file signature is a sequence of bytes in the header at the beginning of each file. It is not obligatory, for instance text files do not have it. Most common files used in computer systems may consist of a header, content data and a footer. The header and the footer are not obligatory. Customarily, the header stores information specific to the file format immediately after the signature. Metadata can indicate parameters such as file size, data format, software version that was used to generate the file, and the like. The most frequently analyzed file types such as jpg, doc, rtf, bmp and their signatures are shown in Table 1. The comparison of a signature with file extension allows to quickly identify files with an invalid (changed) extension. For example, while downloading files a web browser first creates a temporary file, such files can go unnoticed if we take into account only file extensions.

Tab. 1. Some of the most frequently analyzed file type signatures [9]

File format	Signature (hex)
jpg	FF D8 FF
doc	D0 CF 11 E0
rtf	7B 5C 72 74 66 31
pdf	25 50 44 46
bmp	42 4D
avi	52 49 46 46

In case of deletion in NTFS partition and the loss of the corresponding $MFT information (Master File Table), the deleted file is no longer visible to the system. The file data can be 100% complete, however, they are stored in the unallocated space. One method for retrieving a file in this case is to look for its signature. This involves searching the entire unallocated space. In order to retrieve the file, the moment the signature is found, it should be extracted with some data following it. Unfortunately, in many cases the header files do not store information about the file size. Therefore, the amount of data to extract is determined arbitrarily by a specialist before starting the retrieval procedure. File fragmentation is yet another issue. If you find a fragmented file signature you will find its first block and the remaining parts will be lost (Figure 1). Since the size of the fragments is not known, we do not know how many clusters after signature need to be recovered. In this case, it is not known whether the clusters 4-7, belongs to the analyzed file or not.

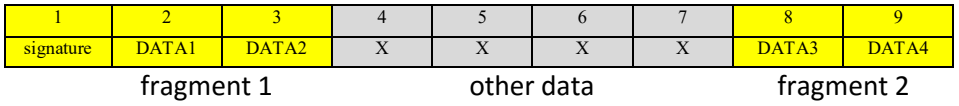

Fig. 1 File fragmentation

The analysis of the unallocated space for signature causes a number of other problems. The most serious problem are the so-called false positives. Since the signature is short, there is a high probability that during the search an array of bytes will be found which has the value of the signature in question, but which is not the signature itself. In addition, many different types of files with different content have the same signature. The types of files with a signature identical to MS Word files are presented in Table 2.

The figure below shows the result search for PDF and Office files in unallocated space. EnCase v7 identified 1,200 files. In the case of the files for which EnCase was unable to find the end of the file, the file size was assumed at 4 096 000B. Extraction of a smaller file was not synonymous with finding the end of the file by the program. It is possible that the recovered file was on the

border of the unallocated space and the program simply could not export more data. This meant the need to manually browse each file for its readability. In this case, out of 1,200 files only one file was correct.

Tab. 2. Types of files with the same signature [9]

File format	Name of the program
DOC, DOT, PPS, PPT, XLS	MS Office 2003
DB	MS Works Database
MSC	Microsoft Common Console Document
MSI	Microsoft Installer Package
OPT	Developer Studio File
VSD	Visio File

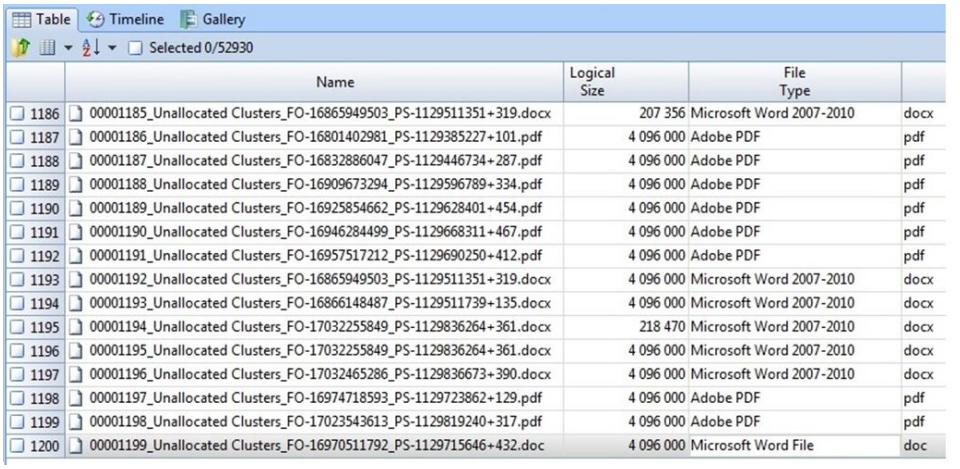
Table Timeline Gallery

Selected 0/52930

	Name	Logical Size	File Type	
1186	00001185_Unallocated Clusters_FO-16865949503_PS-1129511351+319.docx	207 356	Microsoft Word 2007-2010	docx
1187	00001186_Unallocated Clusters_FO-16801402981_PS-1129385227+101.pdf	4 096 000	Adobe PDF	pdf
1188	00001187_Unallocated Clusters_FO-16832886047_PS-1129446734+287.pdf	4 096 000	Adobe PDF	pdf
1189	00001188_Unallocated Clusters_FO-16909673294_PS-1129596789+334.pdf	4 096 000	Adobe PDF	pdf
1190	00001189_Unallocated Clusters_FO-16925854662_PS-1129628401+454.pdf	4 096 000	Adobe PDF	pdf
1191	00001190_Unallocated Clusters_FO-16946284499_PS-1129668311+467.pdf	4 096 000	Adobe PDF	pdf
1192	00001191_Unallocated Clusters_FO-16957517212_PS-1129690250+412.pdf	4 096 000	Adobe PDF	pdf
1193	00001192_Unallocated Clusters_FO-16865949503_PS-1129511351+319.docx	4 096 000	Microsoft Word 2007-2010	docx
1194	00001193_Unallocated Clusters_FO-16866148487_PS-1129511739+135.docx	4 096 000	Microsoft Word 2007-2010	docx
1195	00001194_Unallocated Clusters_FO-17032255849_PS-1129836264+361.docx	218 470	Microsoft Word 2007-2010	docx
1196	00001195_Unallocated Clusters_FO-17032255849_PS-1129836264+361.docx	4 096 000	Microsoft Word 2007-2010	docx
1197	00001196_Unallocated Clusters_FO-17032465286_PS-1129836673+390.docx	4 096 000	Microsoft Word 2007-2010	docx
1198	00001197_Unallocated Clusters_FO-16974718593_PS-1129723862+129.pdf	4 096 000	Adobe PDF	pdf
1199	00001198_Unallocated Clusters_FO-17023543613_PS-1129819240+317.pdf	4 096 000	Adobe PDF	pdf
1200	00001199_Unallocated Clusters_FO-16970511792_PS-1129715646+432.doc	4 096 000	Microsoft Word File	doc

Fig. 2. Retrieving files from the unallocated space

One solution in such a situation is to make use of the characteristic structure of a header and information stored in it (e.g. software version) and thus to "artificially" increase the header length. Such an operation requires tracking changes in each new version of the program, as later distributions may differ in information stored in the headers.

1.2. Hash analysis

Hash values are used in computer science as a method to verify data integrity and authenticity. The most commonly used are MD5 and SHA1 hash values. In general, the hash value is a form of function, which converts the data

into a sequence of bytes. The size of this string is constant for each function and it is, for example, 32B for MD5 and 40B for SHA1. Changing one bit in the input data will generate a different hash value. We can scan any file on the storage device in order to detect known files using hash values, thus we are able to greatly narrow the scope of the analysis. The same method can be used to reveal dangerous or illegal files. A database of software and known system files can be obtained, for example from the internet in the form of downloadable libraries ready to be imported into Encase and FTK [10].

The algorithm for calculating hashes for data blocks is an extension of this method [11]. The author suggests dividing the selected file into fragments (usually the size of 1 cluster, but the size is arbitrary) and then to calculate hash values for each of the fragments separately. The hash value of each element and its location in the file will be saved in a special database. Then you should proceed with scanning the whole unallocated disk space. The search process consists of loading the information about the size of a single file fragment, reading the unallocated area in blocks and then calculating it's the hash value and checking whether the database contains such hash value. As the hash values for all fragments of the file in question are known, the method allows to retrieve all data, regardless of whether the file had been fragmented or not. It is also possible to retrieve partially overwritten files (Figure 3).

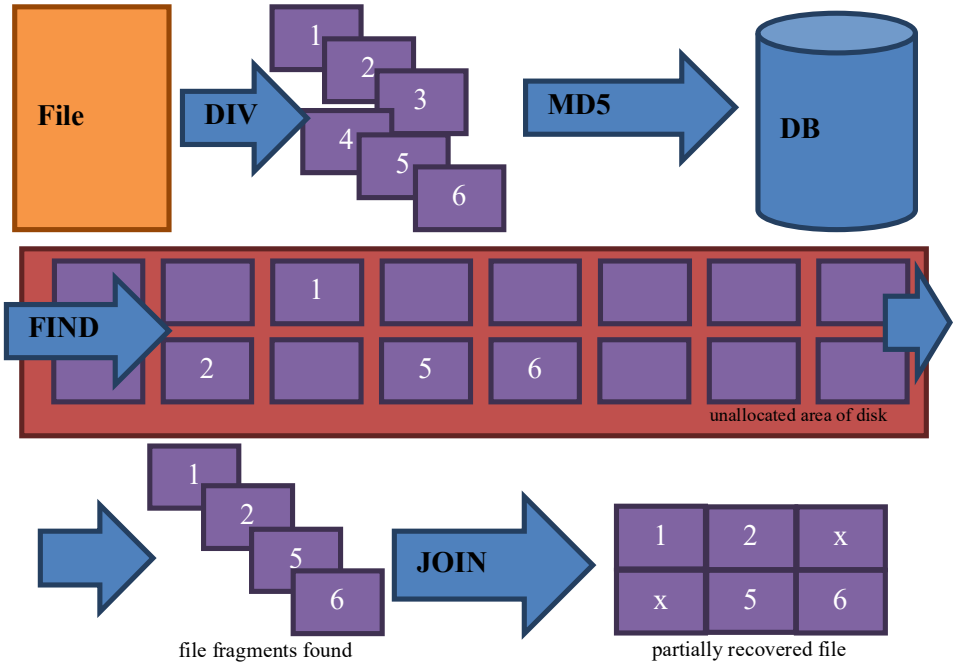

Fig. 3. File Block hash analysis

The basic requirement for hash analysis is the availability of hash values for the file in question. You cannot find the file that is not known, which significantly narrows the usefulness of this method. If the file in question has been changed (e.g. a JPG image was scaled or the range of colours was changed), this method will not provide satisfactory results. Even if the change to a text in a MS Office 2007 document (these are compressed files) was very little this method will fail.

1.3. Keyword analysis

The quality of the results of such an analysis depends on the quality of keywords. It is advisable to avoid keywords such as, e.g. user name or the name of the computer, because this generates thousands of hits both in documents as well as in the system registry as system continuously adds and deletes items. Inappropriate choice of keywords results in a large number of hits in the files and in the unallocated space. It is much better to enter search patterns in the form of whole sentences, but then you must know the content of the document (e.g. its printed version). Previous search methods treated each file as a text file. Such an approach was insufficient in the case of document scans and non-textual documents in general. Compressed files were also excluded from the search process. This was a big problem, because the latest MS Office documents and OpenOffice documents are, in fact, compressed XML files. Current versions of EnCase and FTK can analyze documents (MS Office, OpenOffice and Adobe PDF) not on the basis binary data (which can be compressed), but based on the actual content (text). Moreover, FTK can find keywords among photos using a built-in OCR module. Keywords search is useful in the case of deleted files with lost signatures. Retrieving the desired file boils down to the analysis of the vicinity of keywords. Due to the considerable narrowing of the search scope file recovery can be performed manually.

1.4. Statistical analysis

The method is based on a statistical analysis of data, usually the size of a sector or a cluster. The simplest method of analysis is to measure the frequency of occurrence of characters in the file. The possible results of such an analysis are presented in a confusion matrix (Table 3). The results of a statistical analysis are given with certain values of the likelihood of error. Rows present the actual file format and columns present file format to which given data was classified. The matrix shows the frequency of the recognition of the file formats in rows to file formats in columns. Ideally, the algorithm would show the value of 1 for the

same types of file formats, and the 0 value in other fields, which would mean the correct classification of all file formats. The presented method enables a high level of correct classification for JPG files and HTML files, which means that the error rate for false positives is low. At the same time, in this method, the rate of true negatives is high, e.g. 21% of DBX files are identified as HTML, which indicates low precision.

Tab. 3. Confusion matrix [12]

	AVI	BMP	DBX	DOC	EXE	GIF	HTML	JPG	PDF	PPT	ZIP
AVI	0.56	0.01	0.01	0.04	0.07	0.01	0.00	0.14	0.03	0.07	0.06
BMP	0.20	0.36	0.00	0.03	0.18	0.00	0.03	0.05	0.04	0.05	0.05
DBX	0.00	0.00	0.45	0.01	0.23	0.00	0.21	0.01	0.08	0.01	0.00
DOC	0.01	0.06	0.01	0.39	0.07	0.00	0.02	0.33	0.00	0.09	0.03
EXE	0.02	0.04	0.02	0.04	0.78	0.00	0.01	0.03	0.01	0.03	0.03
GIF	0.00	0.00	0.00	0.00	0.00	0.35	0.00	0.08	0.02	0.35	0.20
HTML	0.00	0.00	0.00	0.00	0.00	0.00	0.99	0.00	0.00	0.00	0.00
JPG	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.01	0.00
PDF	0.01	0.00	0.01	0.00	0.01	0.01	0.00	0.18	0.59	0.10	0.08
PPT	0.00	0.00	0.00	0.04	0.05	0.00	0.00	0.36	0.01	0.38	0.15
ZIP	0.00	0.00	0.02	0.02	0.06	0.00	0.03	0.39	0.07	0.22	0.18

Statistical analysis has been developed into byte-distribution analysis – where a file type fingerprint is created for particular file types [13], [14], [15]. Statistical methods are suitable for the analysis of files stored on a disk. For some file formats, the existing algorithms can, with almost complete confidence, classify the files appropriately. Statistical analysis is often used to find encrypted files (e.g. TrueCrypt containers). Such files do not have their own signature; they can have any name, extension, and size. Problems arise when trying to use these methods in the recovery of deleted files, and the problem of ambiguity is a key issue.

1.5. Content analysis

One of the methods of content analysis involves combining pieces of image files, for example BMP files [16] or JPG files [17]. The idea is to find the first element of the file, and match it against other possible fragments. The matching process is based on comparing the image content rather than file content. Comparison is done on the edge of the analyzed fragments, using known compression mechanisms of JPG files. This way it is possible to recover a fragmented file. There are more content analysis methods using a specific file structure as a basis for data recovery [18].

Content analysis based methods have a number of limitations. Firstly, it is necessary to carry out a preliminary analysis of the entire disk (signature search, statistical analysis). Such an analysis, although time-consuming, is necessary to define the locations that may contain the files in question. The method does not scale well with increasing distance between the fragments. Computational complexity grows exponentially. Current algorithms enable to recover a file that is split into two parts. A greater degree of fragmentation not only increases the computational complexity, but it can also prevent the recovery of a file if any of the fragments is missing. Erroneous classification of file fragments is yet another serious problem causing a situation in which the recovered files, although they are correct (i.e. they open in the associated programs), contain invalid data - for example, a JPG file where half of the image is blurred (Figure 4 below).

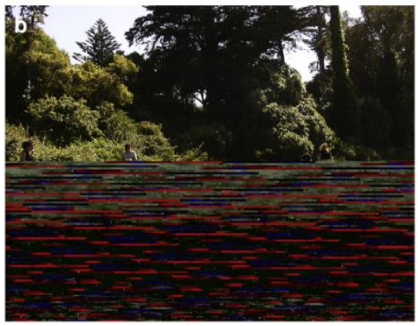

Fig. 4. Inappropriately carved JPG file [17]

2. A case study

Let us consider the example shown in Figure 5: 202,993 is the number of files and folders on the disk. The number of files is typical to modern home computers.

Example of use of file signature and hash analysis.

We will look at image files only. In this case, there are 23,090 image files. First let us proceed with signature analysis and thus exclude invalid or overwritten files. The number of files is further reduced to 19,157. Then, we exclude system files and known files, using, among others, hash and NSRL [10] databases. As a result, the number of files is reduced to 7401. The last step is to filter out duplicates. Due to the relative cheapness of storage space, users generally do not care about the order on the drive. Many files and even entire directories are stored on the drive in multiple copies in different locations. Since

each file has its MD5 hash calculated, we will use this information to detect duplicates - two identical files will have the same MD5 hash.

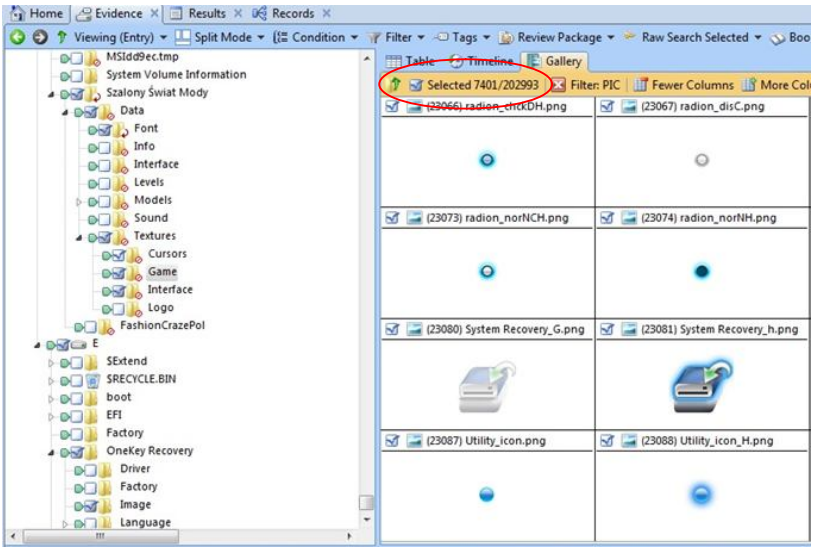

Fig. 5. Screenshot of EnCase 7.10

After applying the appropriate filter the number of files to analyse is reduced to 6,196. The use of signatures and hash analysis reduces the number of files for further analysis on average by 10-30%.

Example of use of keyword analysis.

EnCase has an in-built script to extract the internet browsing history. The script is based, among others, on searching the disk for the term “url: //”. Older versions of Internet Explorer (versions 5 to 9) keep the history in the index.dat file format. Mozilla and Chrome use database files in the SQL file format. They all, i.e. IE, Mozilla and Chrome, store the browsing history in an easily viewable file (in the form of text), hence the possibility to use keyword analysis. Using keyword analysis as a search tool, we do not even have to recover the deleted files with browsing history. It is enough to extract a single data record containing, e.g. the phrase "url: //". Figure 6 shows 21,096 data records, of which most are hits in the unallocated space.

Example of use of statistical analysis.

Encrypted files can be singled out by means of statistical analysis. For that purpose FTK program was used. It calculates the entropy and indicates possibly encrypted files. The program has dozens of in-built types of encryption, so it not only indicates encrypted files, but it also displays information about the encryption method used.

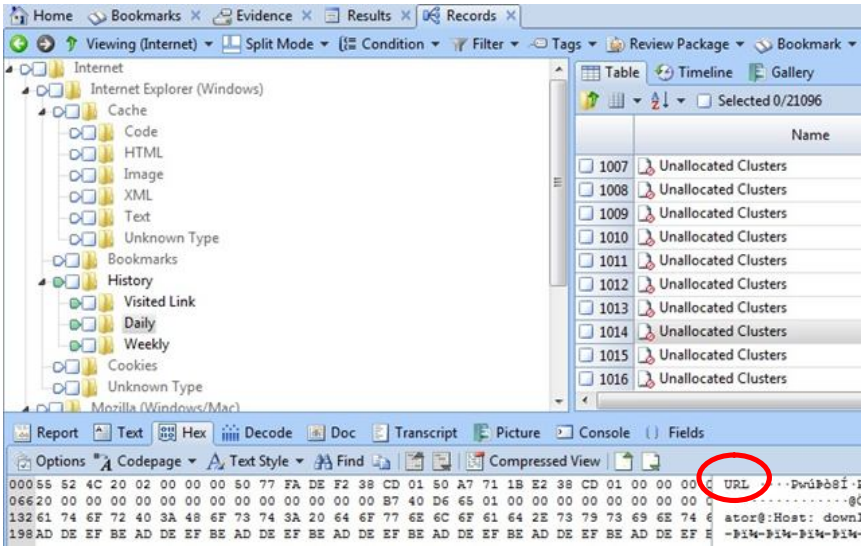

Fig. 6. Sample keyword search

Example of use of content analysis.

FTK's PhotoDNA function [19] detects similar images, therefore it can serve as an example of content analysis. We can find identical files using the hash function (MD5/SHA1). If the colour or the size of the image changes, the hash of the file will also change (due to the change in file content). PhotoDNA does not analyse the bytes in the file, but their representation, that is the whole picture. This method is based on resizing an image to a standard size and changing colours to black and white. Then the data region is divided into sub-areas for which the histogram is calculated. This method allows to find not only identical but also similar files. Nowadays, every photo uploaded to Facebook, Google or Microsoft is scanned using PhotoDNA to detect prohibited content. As shown in the figure below, a standard Windows wallpaper (koala.jpg) was changed, i.e. sketched-in glasses were added. FTK correctly classified the file as similar to the original. The calculated hash functions and the parameter specifying the distance from the original are presented in Figure 7.

3. In search of new methods

The above described data detection techniques are not mutually exclusive. Their combined use can significantly reduce the number of files to be analyzed, and at the same time increase the amount of recoverable data. This paper does not

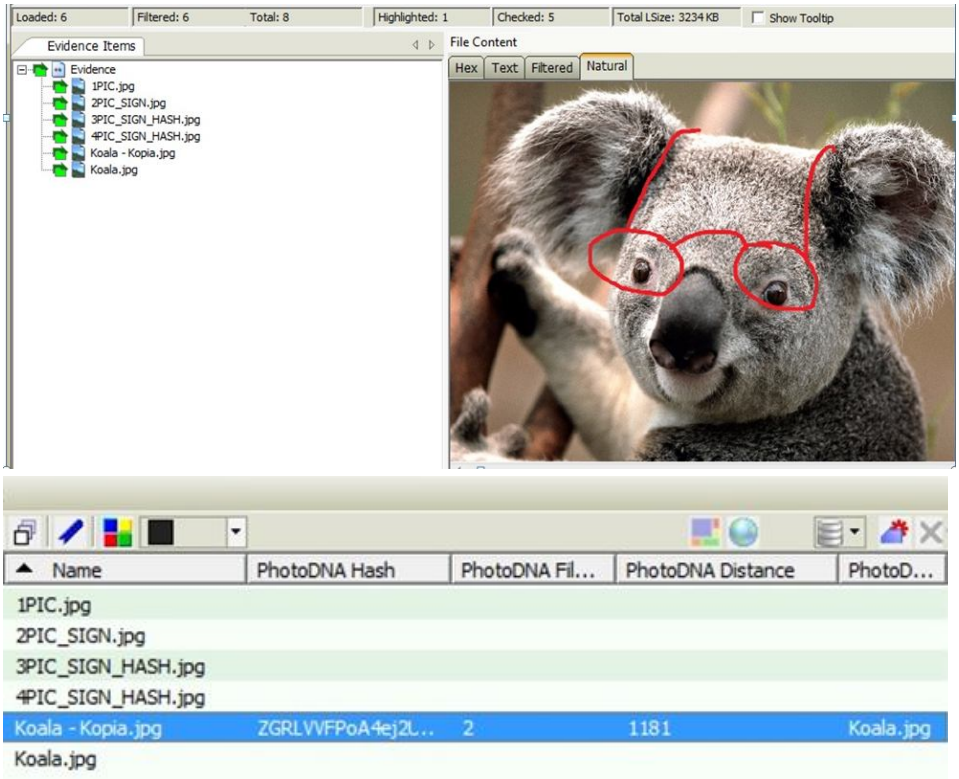

Fig. 7. FTK PhotoDNA function

outline all possible methods of data analysis. The methods using statistical algorithms differ in file/block feature vectors (histograms, entropy, algorithmic complexity, PCA features, etc.) and content classifiers (LDA, neural networks, etc.). Keyword analysis enables full drive content indexation (occurrence of all words) and a drive indexed in such a way can later be searched using, e.g. GREP regular expressions. In short, there are as many solutions as there are tools. One common feature of all these methods is to focus on data area and disregard file system principles. For a file recovered from the unallocated space, it remains unclear when the file was recorded, when it was erased or, in the case of fragmented files, where the other parts are located. Analysis of partitions not only for data but also for meta-data, and for file system operating principles can provide additional information.

3.1. NTFS file system

Let us focus on the NTFS file system, which is most common in home computers. A detailed description of this file system is beyond the scope of this

paper. For the purpose this paper, it is sufficient to define the following system properties:

- MFT file (Master File Table) stores information about each file contained on the drive.
- An MFT file can be treated as an array of records. Each file has one record describing it.
- Each MFT record has a variable that defines how many times the record was previously used by other files. Let us call this variable *L*.
- Data is saved to the drive following the best-fit algorithm (BFA), i.e. the smallest space is chosen from all free spaces equal or larger than the file size (Figure 8).

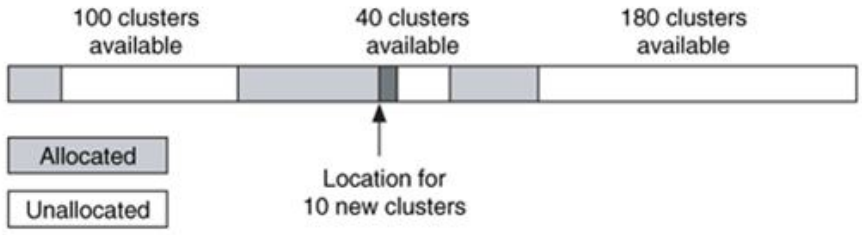

Fig. 8. Best-fit algorithm [20]

Let us now consider MFT File representation as in table below. Each row corresponds to one file.

Tab. 4a. MFT file representation

record	file name	L	occupied clusters
1	A	0	1
2	D	1	9
3	C	0	5-8
4	E	0	2-3

Data distribution on the drive based on MFT stored information is presented in table below.

Tab. 4b Distribution of files on an NTFS partition

cluster	1	2	3	4	5	6	7	8	9	10
file	A	E	E	x	C	C	C	C	D	x

MFT tells us that one file has been deleted. Record No.2 (related to file D) has L=1, which means that some other file had previously occupied that space.

Assuming that subsequent letters of the alphabet correspond to how recent the file is, one can tell that file E was saved last. By removing the file E, that is removing its MFT record and the data, we obtained the look of MFT file and partition as in the tables below.

Tab. 5a. MFT file representation

record	file name	L	occupied clusters
1	A	0	1
2	D	1	9
3	C	0	5-8
4	-	-	-

Tab. 5b. Distribution of files on an NTFS partition

cluster	1	2	3	4	5	6	7	8	9	10
file	A	x	x	x	C	C	C	C	D	x

Now file D is the last saved file. By removing it we removed its data and MFT record. As for this record L=1, it is necessary to add a new file (labelled as B). In accordance with the BFA it may exist only in the 1-3 space.

Tab. 6a. MFT file representation

record	file name	L	occupied clusters
1	A	0	1
2	B	0	2-4
3	C	0	5-8
4	-	-	-

Tab. 6b. Distribution of files on an NTFS partition

cluster	1	2	3	4	5	6	7	8	9	10
file	A	B	B	B	C	C	C	C	x	x

Further operations would involve removing files C, B and A. As a result, we would recreate the sequence of operations from the creation of the partition:

Save file A -> Save file B -> Save file C -> Delete file B -> Save file D -> Save file E

Tracking the full sequence of entries/deletions revealed that the desired data was located in clusters 2-4. In addition, it was found that the file in question was saved after file A, and erased before saving file D and after saving file C. The file in question was partially overwritten (lost signature) and yet the analysis is able to indicate the location of the remaining and recoverable data from the file (cluster 4).

Clearly, the example used to demonstrate the process was an oversimplification, however, the authors hold a view that under certain assumptions it is possible to use NTFS file allocation algorithm for efficient data recovery.

References

[1] CASEY E., *Digital Evidence and Computer Crime*, Second Edition, 2004.

[2] *EnCase Forensic v7*, [Online]. Available: http://www.digitalintelligence.com/software/guidancesoftware/encase7/

[3] *FTK*, AcessData, [Online]. Available: http://accessdata.com/

[4] *X-Ways Forensic*, X-Ways AG, [Online]. Available: http://www.x-ways.net/

[5] *SANS Investigative Forensic Toolkit (SIFT*), [Online]. Available: http://digital-forensics.sans.org/community/downloads

[6] CARRIER B., *The Sleuth Kit*, [Online]. Available: http://www.sleuthkit.org/

[7] CARRIER B., Spafford E.H., *An Event-based Digital Forensic Investigation Framework*, Proceedings of Digital Forensics Research Workshop, Baltimore, 2004.

[8] GARFINKEL S.L., *Digital forensics research: The next 10 years*., Digital Investigation, tom 7, 2010, pp. 64-73.

[9] KESSLER G.C., *File signatures table*, 2014. [Online]. Available: http://www.garykessler.net/library/file_sigs.html.

[10] *National Software Reference Library* (NSRL), [Online]. Available: http://www.nsrl.nist.gov/.

[11] KEY S., *File block Hash Map Analysis*, in Computer and Enterprise Investigations Conference (CEIC), 2011.

[12] VEENMAN C., *Statistical Disk Cluster Classification for File Carving*, Third International Symposium on Information Assurance and Security, 2007, pp. 393-398.

[13] MCDANIEL M., HEYDARI M., *Content based file type detection algorithms*, in Proceedings of the 36th IEEE Annual Hawaii International Conference on System Science (HICSS'03), 2003.

[14] LI W., WANG K., STOLFO S., HERZOG B., *Fileprints: Identifying file types by n-gram Analysis*, Proceedings of the 6th IEEE Systems, Man and Cybernetics Information Assurance Workshop, 2005, pp. 64-71.

[15] AMIRANI M.C., TOORANI M., SHIRAZI A.B., *A new approach to content-based file type detection*, Proceedings of the 13th IEEE Symposium on Computers and Communications (ISCC'08), 2008, pp. 1103-1108.

[16] MEMON N., PAL A., *Automated Reassembly of File Fragmented Images Using Greedy Algorithms*, IEE Transactions on Image Processing, tom 15, nr 2, 2006, pp. 385-393.

[17] PAL A., SENCAR H. T., MEMON N., *Detecting file fragmentation point using sequential hypothesis testing*, Digital investigation, No. 5, 2008, pp. 2-13.

[18] WEI Y., ZHENG N. I XU M., *An automatic Carving Method for RAR File Based on Content and Structure*, Second International Conference on Information Technology and Computer Science, 2010, pp. 68-72.

[19] *Photo DNA*, Microsoft, [Online] Available: http://www.microsoft.com/global/en-us/news/publishingimages/ImageGallery/Images/Infographics/PhotoDNA/flowchart_photodna_Web.jpg

[20] Carrier B., *File System Forensic Analysis*, Addison Wesley, 2005.

Wybrane metody odzyskiwania plików oraz analizy cyfrowych nośników danych stosowane w informatyce śledczej

STRESZCZENIE: Artykuł opisuje najczęściej obecnie wykorzystywane metody odzyskiwania danych z nośników cyfrowych. Przedstawione metody są omówione pod względem swoich wad i zalet. Propozycja nowej metody odzyskiwania plików jest przedstawiona w oparciu o zasady alokacji zasobów w NTFS.

SŁOWA KLUCZOWE: informatyka śledcza, informatyka kryminalistyczna, odzyskiwanie danych, NTFS, MFT

Received: 20.12.2013 r.