FORSIGS: Forensic Signature Analysis of the Hard Drive for Multimedia File Fingerprints

John Haggerty and Mark Taylor
Liverpool John Moores University, School of Computing & Mathematical Sciences, Byrom Street, Liverpool, L3 3AF. E-mail: {J.Haggerty, M.J.Taylor}@ljmu.ac.uk

Abstract. Computer forensics is emerging as an important tool in the fight against crime. Increasingly, computers are being used to facilitate new criminal activity, or used in the commission of existing crimes. The networked world has seen increases in, and the volume of, information that may be shared amongst hosts. This has given rise to major concerns over paedophile activity, and in particular the spread of multimedia files amongst this community. This paper presents a novel scheme for the automated analysis of storage media for digital pictures or files of interest using forensic signatures. The scheme first identifies potential multimedia files of interest and then compares the data to file signatures to ascertain whether a malicious file is resident on the computer. A case study of the *forsigs* application presented within this paper demonstrates the applicability of the approach for identification and retrieval of malicious multimedia files.

1 Introduction

Due to the increased use of computer and network technologies in certain types of criminal activity, computer forensics is emerging as an important tool in the fight against crime. Computer forensics is defined as the application of computer investigation and analysis techniques to determine potential evidence [1]. Crime involving computers and associated technologies may be classified in three ways [2]; the computer is the target of the crime, a repository of information used or generated during the commission of a crime, or as a tool in committing a crime. Therefore, the investigation and analysis techniques are wide and varied, and often rely on the context surrounding the activity under scrutiny.

A major advantage within the networked world is the speed and volume of information that may be shared between hosts. This has given rise to major concerns over paedophile activity and the spread of multimedia files, in particular indecent

Please use the following format when citing this chapter:

Haggerty, J. and Taylor, M., 2007, in IFIP International Federation for Information Processing, Volume 232, New Approaches for Security, Privacy and Trust in Complex Environments, eds. Venter, H., Eloff, M., Labuschagne, L., Eloff, J., von Solms, R., (Boston: Springer), pp. 1–12.

images of children amongst this community. As with many types of computer crime, whilst an investigation may begin with one suspect, it is rare that it will end with only the same suspect. In addition, due to the greater capacity of today's hard drives (upwards of 200 Gigabytes) and the amount of multimedia information that may be held on a suspect's computer, a large volume of data must be analysed. A major challenge facing law enforcement and national security is accurately and efficiently analysing this growing volume of evidential data [3].

The current practice of searching a hard drive for evidence is a time-consuming, manual process. An image of the hard drive is taken to replicate the original evidence source, and this itself may take 24 – 48 hours to make a robust copy. A forensics tool is then used to recreate the logical structure of the underlying file system. A computer forensic analyst views the files, both extant and deleted, and files of interest are reported with supporting evidence, such as time of investigation, analyst's name, the logical and actual location of the file, etc. As the investigation of the hard drive relies on the analyst viewing files as if part of the file system, this process is laborious. Therefore, attempts by practitioners have been made to improve the speed of the search within the constraints of the tools at their disposal. Some practitioners achieve this by comparing MD5 file checksums from the files on the hard drive under investigation to MD5 checksums of known malicious files recorded from previous investigations.

This paper presents a novel scheme for the forensic application of signature analysis, and in particular, the search of raw data for evidence of illegal or suspicious multimedia files[1] resident or deleted on the hard drive. This approach focuses on the the speed of search and robust identification of multimedia files of interest. It is recognised that other pieces of information would be gained from a manual search using the logical file structure, such as time of file creation, access or modification, etc., once these files are identified and located.

The advantages of the forensic signature approach are fourfold. First, the speed of analysing a suspect's machine may be reduced by automating the search process. This is not to say that a manual inspection may not be required later, but it can direct the forensic analyst to the relevant areas of the hard drive. However, this approach may be used to detect data not discovered by more traditional computer forensic techniques [2]. Second, the application can be extended to investigate related data types other than multimedia files, such as searching for text strings or evidence produced by other applications, e.g. Word, e-mail, etc. Third, current practice by some practitioners of analysing data for known files of interest utilises searches for MD5 checksums produced during previous investigations. Recent research has questioned the reliability of MD5 and SHA-1 hash functions in producing digital signatures [4], and this will have serious ramifications within the legal arena. In addition, a suspect may avoid detection by altering just one byte within the multimedia file of interest which will alter the MD5 checksum produced. Finally, the analyst is not required to look at any images that they may find disturbing and which may have an adverse psychological effect on them.

The remainder of this paper is organised as follows. In section two, related work is discussed. Section three first provides an overview of the way in which

[1] In this paper, the term multimedia file(s) refers to a digital picture(s).

multimedia files are organised on the hard drive that aids forensic signature analysis. It then provides an overview of the digital fingerprint signature analysis approach. Section four presents the results of a case study to demonstrate the applicability of the approach. Finally, section five discusses further work and we make our conclusions.

2 Related Work

A number of computer forensic tools and approaches are used for the detection of suspicious images located on the hard drive. These can be generally divided into *file analysis* and *format specific* approaches.

Commonly used computer forensic tools, such as *Forensic Toolkit* (*FTK*) [5] and *EnCase* [6], provide examples of *file analysis* approaches. These tools are used for storage media analysis of a variety of files and data types in fully integrated environments. For example, *FTK* can perform tasks such as file extraction, make a forensic image of data on storage media, recover deleted files, determine data types and text extraction. *EnCase* is widely used within law enforcement and like *FTK* provides a powerful interface to the hard drive or data source under inspection, for example, by providing a file manager that shows extant and deleted files. These tools have in common the ability to read the data source as a whole, irrespective of the underlying logical structure of the operating system. Whilst these applications provide a robust forensic analysis, they are often time consuming in building a case due to the analyst having to manually read the data, e.g. looking at file contents, recovering deleted files, etc., to determine the relevance of the files to the investigation.

Format specific approaches specifically look for data belonging to particular applications or data types. For example, *Jhead* [7] is an application to extract specific Joint Photograph Experts Group (JPEG) image data, such as time and date a picture was taken, camera make and model, image resolution, shutter speed, etc. Tools such as *Data Lifter* [8] are able to extract files of a multitude of types. These tools support data carving to retrieve files of specific types by searching the disk for file preambles. The main problem with these tools is that they are not designed for robust forensic analysis. For example, *Jhead* enables the user to alter JPEG files. Whilst *DataLifter* extracts files of particular types, it does not differentiate between suspicious, malicious or benign files. Therefore, the user must still manually trawl through the extracted files to determine the nature of the file and its relevance to the investigation.

Recent research has recognised the disadvantages of current practice and has therefore proposed alternative approaches. These approaches attempt to not only identify file types, but also known files of a particular type by utilising statistical data derived from file analysis. For example, [9] posit a method based on intrusion detection to identify files of interest. This method models mean and standard deviation information of individual bytes to determine a *fileprint*, or identification of a specific file. This method is dependent on file header data for file categorisation, and therefore requires that the files are not fragmented and for the file system to be intact [10]. As such, [11] propose the Oscar method, which determines probable file types from data fragments. This approach, unlike the previous one, aims to identify files based on fragmented data, such as that in RAM, and therefore does not require header information or an extant file system. A disadvantage of this approach is that it uses a more computationally exhaustive statistical measure than [9] with not much advantage in detection rate, in order to achieve the identification of data fragments.

3 Digital Fingerprints Using Signature Analysis

Within this section the properties of data resident on a hard drive or other storage media that are relevant to this approach are described and an overview of the digital fingerprint method is provided. The scheme for the *forsigs* (forensic signature) analysis for multimedia files is also posited.

3.1 Organisation of Files on the Hard Drive

The hard drive typically consists of a set of data structures organised into layers for access. At the highest level is the *hard drive* itself, which can be configured into one or more *partitions*. Partitions allow a single hard drive appear to be a number of individual drives and are referenced by the underlying operating system and partition tables. At the next level sits the *filesystem*. The filesystem determines how data is stored on the disk and provides a logical map to the data resident within the partition. A filesystem is typically organised into a set of *directories*. Directories provide a hierarchical organisation and referencing system for *files*. The file is a data structure, created by a person or system, that holds relevant information for both the user and the operating system.

The underlying hard drive on which data resides is organised into a series of memory locations, called *sectors*, which are typically 512 bytes long. The operating system organises these memory locations, 'blocks' in Linux or 'clusters' in Windows, to hold a finite amount (512 bytes to 4,096 bytes) of information. As files are normally much larger than this size, data is segmented and stored in a series of blocks, which may or may not be sequentially ordered. The relevant blocks associated with a file are then referred to by a master file table to allow the file to be seamlessly recreated when accessed by the user in the associated application. As a file is rarely an exact multitude of available bytes, the last block will contain as much information as required before placing an *end of file* (EOF) indicator for the operating system. The operating system reads the data up to the EOF indicator but no further. This gives rise to the condition known as 'slack space'. As this last block may have been used previously, data from the previous file using that block not overwritten for the new file will remain extant after the EOF indicator. Thus, partial file fragments of deleted files may be retrieved.

Signature analysis of multimedia files on the storage media relies on the premise that a significant number of files that may be of interest to the forensic examiner follow a relatively simple structure [2]. The *file header* contains information specific to the file format, for example, whether it is a JPEG, Graphic Interchange Format

(GIF), Microsoft Office, etc., file. The *file body* contains the data pertinent to the file itself. This data is used to reconstruct the multimedia file on the computer, thus enabling the user to view the data as a picture within an application. However, additional information is also stored within this file section that may have a direct impact on the investigation when analysing image files. For example, camera make and model, photograph time, application used, font types, etc. This information may be used with other forms of evidence, such as physical seizure of a camera at the scene of a crime or existence of a particular user application, to help the forensic analyst build their case. The *file footer* indicates information such as EOF.

3.2 Digital Fingerprint Approach

Previous work focused on identifying the file type, for example through the JPEG header, and then comparing the entire block to a signature block [12]. Whilst this approach proved successful, three principal problems remain. First, only the first block of any file is used for signature comparison. Much of this first block reveals supporting evidence such as the application used to create the file, camera make and model, time and date a picture was taken, setting up fonts, etc. The image itself starts later in the block, thus leaving much redundant information to be searched. This is particularly pertinent when looking at Linux blocks, which are substantially smaller than Windows clusters, and therefore the first block holds less picture information. Second, knowing that the approach focuses on matching the entire first block of a file of interest to a signature block requires that the suspect only changes one byte within this data to avoid detection. This is a similar problem to that faced by practitioners today relying on MD5 checksum comparisons. Third, the entire evidence file[2] is loaded into the application leading to additional computational load on the application, and therefore search time. In addition, the application is bounded by the size of the evidence data and in experiments could only read evidence files up to 500 Mb in size.

[2] The term evidence file refers to the entire sequential contents of a hard drive transformed into a file.

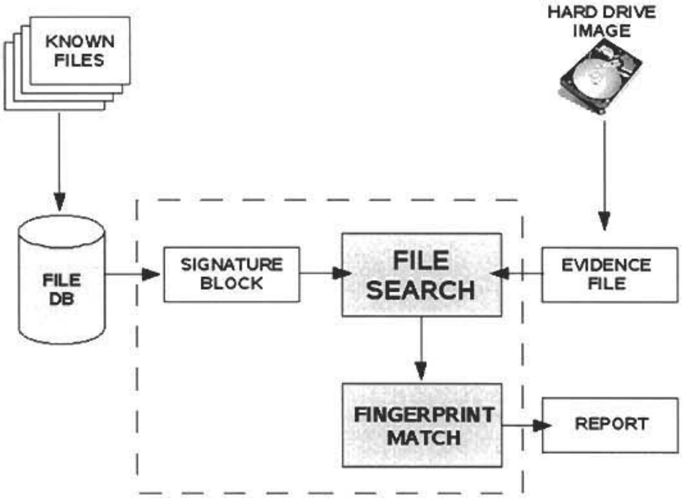

Fig. 1. Application overview.

Figure 1 provides an overview of the signature analysis approach. The dashed line represents the application and its internal components. Known multimedia files collected from previous investigations are collected in raw data form, and are stored in a database. This database views file data in hexadecimal form rather than as an image. Thus, anyone interrogating the database is not confronted with indecent images of children, just hexadecimal values. In this way the analyst does not have to view disturbing images, thereby alleviating psychological pressures involved in this type of investigation. This also has the advantage that signatures, as partial fragments of hexadecimal data, may be shared by authorities without the legal restrictions of disseminating the entire image.

The hard drive seized from the suspect's computer is copied, or *imaged*, in a robust manner in order to protect the original data. This is a requirement of current practice, whereby the analyst does not interrogate the original hard drive, as this may alter evidence located there. Every byte is copied across to a replica hard drive to provide an exact duplicate of the original to protect the integrity of the evidence. The raw data on the imaged hard drive at byte level forms an evidence file which is analysed by the fingerprint application.

The file database provides a signature block(s), which is used by the application for comparison to data in the evidence file. The signature is formed from a single block from the original multimedia file held on the file database and obtained using the *siggrab* application developed by the authors. This may be any part of the file, and the size of the block alters depending on the underlying operating system used by the suspect. By focusing on a single block, the search does not rely on a file being sequentially ordered on the suspect's computer; this block could reside anywhere on the hard drive, and data prior to or after this block may be related to the same file, or not. The choice of signature block will be discussed in section 3.3.

The *forsigs* application reads in both the signature block(s) and the evidence file to conduct the signature search. The file is searched for evidence of known multimedia files of interest. Once the search is complete, a report is generated for the analyst. This process is described in the next sub-section.

3.3 Digital Fingerprint Signature Search

The signatures on which we search will have an impact on the effectiveness of the approach. If we were to search for a whole digital image file, there are four factors that will affect the search. First, digital images may be in the region of Megabytes, thereby requiring a large signature to be transferred from the file database. Second, only one byte in the original data need be altered to give rise to a large number of false negatives. Third, clusters allocated to a digital image are not necessarily sequentially ordered on the underlying storage media. As the data is not sequentially stored, searches using large signatures may be defeated due to fragmentation of the picture in the evidence file. Finally, as has been found in intrusion detection, large signatures can be computationally exhaustive [13], and this is also assumed to be the case with searches of storage media.

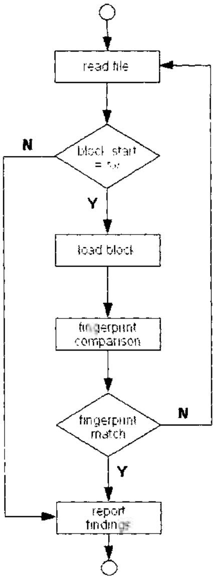

Fig. 2. Digital fingerprint signature search process.

Figure 2 provides an overview of the digital fingerprint signature search process. The application reads in the evidence file. This is conducted a character at a time to ensure that all bytes are analysed. Whilst this may add some computational overhead, this ensures the robustness of the data collection by the application. The beginning of each block within the evidence file is compared to the first byte of the file of interest's signature block. If none of these are found within the file, the application reports that no signatures are matched, and therefore, no multimedia files of interest are resident on the hard drive. If, however, there is a match, the remainder of the block is loaded for comparison. A fingerprint comparison is conducted, and if a match is confirmed, a report is generated. If the fingerprint is not confirmed, the application continues to search the data until the end of the evidence file.

For the signature search, an *ad hoc* block within the original multimedia file is chosen. With many multimedia files being upwards of hundreds of kilobytes in size, the file itself will use many blocks. This ensures that a suspect would be required to alter the start of every block of data stored on the hard drive as they will not know which block is used for the signature. Whilst this scheme has been used as the basis for the case study to demonstrate the applicability of the fingerprint approach, anywhere in the block can be used as the basis for identification of files of interest. This would further complicate attacks on the scheme.

The first and last block of a file are not provided for the signature search. As discussed earlier, the first block may hold generic but redundant data, such as setting up fonts, and therefore is less robust for analysis. The last block will include slack space data, and therefore cannot provide a reliable signature. Using either of these two blocks will lead to false postitives.

The application loads suspected blocks from the evidence file and compares them to the signature block. In order to defeat the possible attack of a suspect altering bytes, 16 points of reference within the signature block are compared to the corresponding points of reference in the evidence block. The positions that are compared can be randomised. If a match is found, it is reported to the analyst.

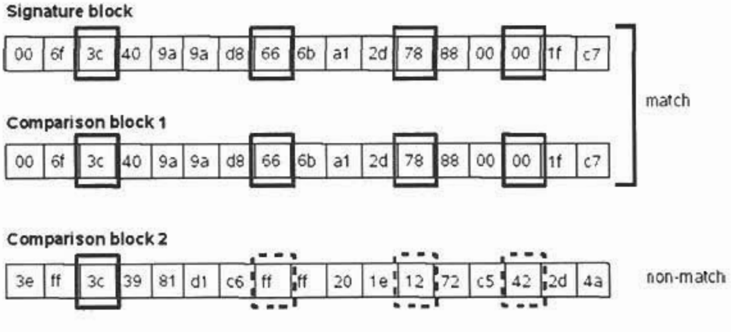
Signature block
00 6f 3c 40 9a 9a d8 66 6b a1 2d 78 88 00 00 1f c7
match
Comparison block 1
00 6f 3c 40 9a 9a d8 66 6b a1 2d 78 88 00 00 1f c7
Comparison block 2
3e ff 3c 39 81 d1 c6 ff ff 20 1e 12 72 c5 42 2d 4a
non-match

Fig. 3. Digital fingerprint signature matching.

Figure 3 illustrates the digital fingerprint search process. The signature block refers to a signature read into the application from the signature file, as illustrated in figure 1. Suspicious blocks from the evidence file that potentially could be from a file of interest are placed into a comparison block. Sixteen points within the block are compared to the corresponding points within the signature block. Highlighted in boxes in figure 3 are the points at which comparisons are made. Comparison block 1 reveals that all points of comparison match, and therefore a fingerprint is found. Comparison block 2 shows that only the first comparison value is matched, whereas the rest do not, as indicated by dashed boxes. All 16 points within the file must be matched to identify a signature.

The probability that all sixteen points within a block will match that of the fingerprint taken from an *ad hoc* block is remote. A single byte can take any one of $2^8 = 256$ distinct values. The probability that a byte triggering a fingerprint comparison will match in arbitrary file with an even distribution of independent values will therefore be 1/256. The probability that all sixteen values will present a perfect match is approximately 2.29×10^{-1532}.

4 Case Study and Results

The previous section presented an overview of the novel *forsigs* scheme; the forensic application of signature analysis, and in particular, the search of raw data for evidence of illegal or suspicious multimedia files resident or deleted on the hard drive. This section provides a case study to demonstrate the fingerprint signature approach and presents its results.

Experiments were undertaken on a 2 Ghz AMD Athlon host with 256 Mb RAM running Suse Linux 10. This represents a similar set up to that which would be deployed in the field on a laptop by a forensic analyst, and therefore provides a useful benchmark for speed and efficiency tests. More computational power could be provided within the forensics lab.

Four files, ranging from 250 Mb to 2 Gb data sets, are used as the basis of the evidence search. These files are images taken from real computer data to form a single evidence (or search) file. The filetypes in the evidence data are wide and varied, including MP3, system files (dat, swf, dll, and Master File Table information, etc.), exe, ogg, pdf, ppt, doc, and jpg files. Amongst this data are a wide range of digital picture files other than the specific file(s) of interest. This ensures that the *forsigs* application has many opportunities to return false positives by incorrectly identifying a digital picture. In addition, many of these pictures were taken with the same camera, of a similar subject and in similar lighting conditions. It should be noted that no malicious images are used in the tests, only benign pictures and consist of images of historical manuscripts from archives. In all the data sets, a single file of interest was placed amongst the data. In all cases, the *forsigs* application successfully identified the signature and location of the file of interest and no false positives were returned.

Figure 4 illustrates the time *forsigs* takes to search evidence files for a single signature ranging from 250 Mb to 2 Gb in size. The time is recorded as both real and user time and returned by the system. Real time represents the actual time between invocation and termination of the program, whereas the user time records the actual CPU time of the application. The real time measure is used to represent a maximum search time. The search and correct identification of a single signature within 250 Mb of data takes approximately 11.5 seconds, as opposed to 95 seconds for 2 Gb. With a 1 Gb search taking approximately 45 seconds, it can be assumed that it may take 75 minutes to search all bytes on a 100 Gb hard drive, which is significantly faster than a manual search of the same scale.

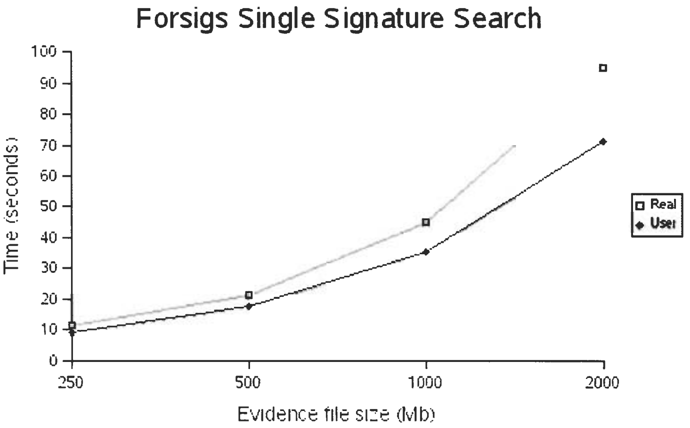

Fig. 4. Digital fingerprint signature matching over time.

In order to evaluate the impact of searching for more than one signature, between one and five signatures are simultaneously searched for within a 250 Mb evidence file. The application reports the detection of any of the signatures, if found. Again, real and user times are recorded, as illustrated in figure 5. Interestingly, the results show that it takes slightly more time (both real and user) to search for a single signature than for multiple signatures. This is despite the number of comparison indicators identifying the possibility of a block of interest. The real times recorded ranged from 12.1 seconds for three signatures (the fastest) to 12.9 seconds for a single signature (the slowest). However, this efficiency may be demonstrated by the average search time per signature; 12.9 seconds for a single signature but just over 2 seconds per signature when searching for five signatures.

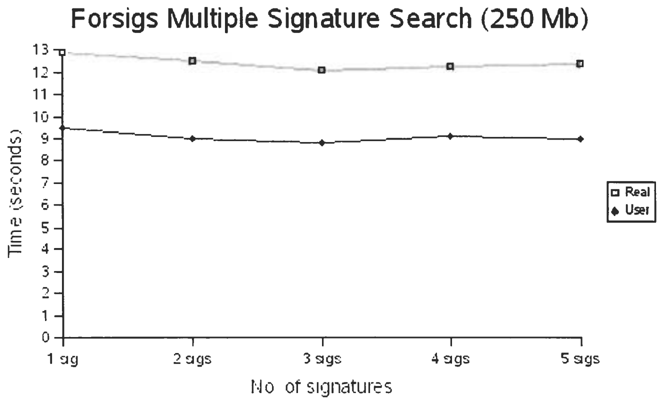

Fig. 5. Impact on search time of multiple signature search.

The efficiency of the *forsigs* approach is also demonstrated by the increased number of comparisons that multiple signature searches require. Figure 6 illustrates the number of comparisons on the 'trigger' byte that indicates the possibility of a block of interest and therefore will be compared to a signature. The similarity in comparisons between three and four signatures is due to the comparison indicator being the same for both signature blocks. However, *forsigs* still correctly identifies the correct signature in both these cases. Therefore, the number of comparisons that the application must make does not have an adverse effect on the time of search, as indicated above.

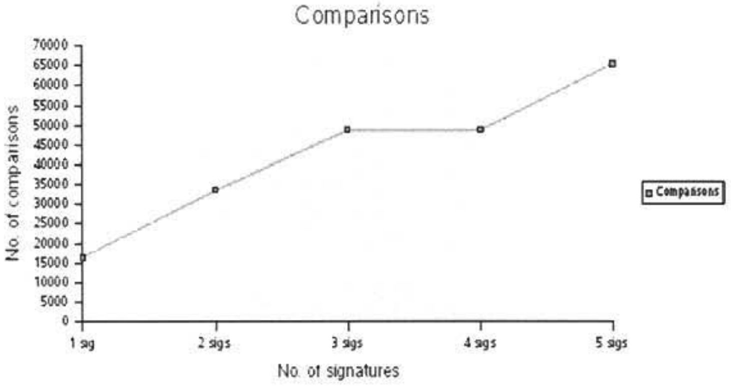

Fig. 6. Number of *forsigs* comparisons on a 'trigger' byte.

The case study demonstrates the applicability of the *forsigs* approach. Despite the number of signatures, and therefore comparisons, the program identifies and locates the signature, and therefore file of interest, correctly and efficiently.

5 Conclusions and Further Work

Further work aims to extend the tests to include larger data sets and wider signature searches to build on the work in this paper. In addition, work is being conducted into the position within a file that provides the optimum signature for *forsigs* searches. Whilst this paper has focused on digital pictures, other file types are of interest to the forensic examiner and the application of the *forsigs* approach to these will be investigated. Finally, compression, resizing or encryption of files of interest has an adverse effect on this approach. Therefore, future work will attempt to address the prediction of these algorithms on a file of interest, and the signature that may be produced.

This paper has presented the novel *forsigs* approach for forensic signature analysis of the hard drive for multimedia file fingerprints. The widespread use of computer and network technologies has given rise to concerns over the spread of digital picture files containing indecent images of children. Current forensic analysis techniques are time consuming and laborious, as well as raising the psychological burden on the forensic analyst by viewing such images. Therefore, the *forsigs* approach provides a means by which hard drives may be searched automatically and efficiently for evidence of malicious images. The approach identifies potential files of interest and compares them to known images to determine whether data contained on a hard drive is malicious or benign. The case study presented in this paper demonstrates the applicability of this approach.

References

1. Li, X. & Seberry, J., "Forensic Computing", *Proceedings of INDOCRYPT*, New Delhi, India, 8-10 Dec 2003, LNCS 2904, Springer, 2003, pp.18-35.
2. Mohay, G., Anderson, A., Collie, B., De Vel, O. & McKemmish, R., *Computer and Intrusion Forensics*, Artech House, MA, USA, 2003.
3. Chen, H., Chung, W., Xu, J.L., Wang, G., Qin, Y. & Chau, M., "Crime Data Mining: A General Framework and Some Examples", *Computer*, April 2004, pp. 50-56.
4. Burr, W.E., "Cryptographic Hash Standards Where Do We Go from Here?", *IEEE Security and Privacy*, March/April, 2006, pp. 88-91.
5. The Forensics Toolkit, available from http://www.accessdata.com, accessed October 2006.
6. Guidance Software Encase, available from http://www.guidancesoftware.com, accessed October 2006.
7. Jhead, available from http://www.sentex.net/~mwandel/jhead/ , last updated April 2006, accessed October 2006.
8. DataLifter Computer Forensic Software, available from http://datalifter.com/products.htm, accessed October 2006.
9. Li, W. J., Wang, K., Stolfo, S. & Herxog, B., "Fileprints: Identifying File Types by n-gram Analysis", *Proceedings of the 6th IEEE Systems, Man and Cybernetics Assurance Workshop*, West Point, NY, USA, June, 2005.
10. Karresand, M. & Shahmehri, N., "Oscar – File Type Identification of Binary Data in Disk Clusters and RAM Pages", *Proceedings of IFIP SEC 2006,* Karlstadt, Sweden, 22 – 24 May, 2006.
11. Karresand, M. & Shahmehri, N., "File Type Identification of Data Fragments by their Binary Structure", *Proceedings of the 2006 IEEE Workshop on Information Assurance*, US Military Academy, West Point, NY, 21-23 June, 2006.
12. Haggerty, J., Berry, T. & Gresty, D., "Forensic Signature Analysis of Digital Image Files", *Proceedings of the 1st Conference on Advances in Computer Security and Forensics*, Liverpool, UK, 13-14 July, 2006.
13. Zhang, Y. & Paxson, V., "Detecting Backdoors", *Proceedings of USENIX Security Symposium*, Denver, CO, USA, 2000.