




Jun 11th, 10:25 AM

System-Generated Digital Forensic Evidence in Graphic Design Applications

Enos Mabuto
University of Pretoria, South Africa

Hein Venter
University of Pretoria, South Africa

Follow this and additional works at: <https://commons.erau.edu/adfsl>

 Part of the [Computer Engineering Commons](#), [Computer Law Commons](#), [Electrical and Computer Engineering Commons](#), [Forensic Science and Technology Commons](#), and the [Information Security Commons](#)

Scholarly Commons Citation

Mabuto, Enos and Venter, Hein, "System-Generated Digital Forensic Evidence in Graphic Design Applications" (2013). *Annual ADFSL Conference on Digital Forensics, Security and Law*. 9.
<https://commons.erau.edu/adfsl/2013/tuesday/9>

This Peer Reviewed Paper is brought to you for free and open access by the Conferences at Scholarly Commons. It has been accepted for inclusion in Annual ADFSL Conference on Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.

EMBRY-RIDDLE
Aeronautical University™
SCHOLARLY COMMONS

(c)ADFSL



SYSTEM-GENERATED DIGITAL FORENSIC EVIDENCE IN GRAPHIC DESIGN APPLICATIONS

Enos Mabuto, Hein Venter
University of Pretoria
South Africa

ABSTRACT

Graphic design applications are often used for the editing and design of digital art. The same applications can be used for creating counterfeit documents such as identity documents (IDs), driver's licences, passports, etc. However, the use of any graphic design application leaves behind traces of digital information that can be used during a digital forensic investigation. Current digital forensic tools examine a system to find digital evidence, but they do not examine a system specifically for the creating of counterfeit documents created through the use of graphic design applications.

The paper in hand reviews the system-generated digital forensic evidence gathered from certain graphic design applications, which indicates that a counterfeit document was created. This inference is made by associating the digital forensic information gathered with the possible actions taken, more specifically, the scanning, editing, saving and printing of counterfeit documents. The digital forensic information is gathered by analysing the files generated by the particular graphic design application used for creating the document. The acquired digital forensic information is corroborated to the creation of counterfeit documents and interpreted accordingly. In the end determining if a system was utilised for counterfeiting.

Keywords: Digital evidence, Digital forensic, Digital forensic artifacts, Graphic design applications.

1. INTRODUCTION

Industries including but not limited to advertising, newspaper printing, architecture, fashion and design, project management and manufacturing make use of graphic designs for their corporations. Graphic design applications have enhancing tools like paint brushing, vector drawing, digital pen and pencil drawing, and many more. These graphic design applications are used to facilitate the creation of unique art for company logos, magazine advertising or computer-aided design, to mention but a few. Most industries make use of graphic design applications for visual presentations and use pictorial expressions that aid communication and the expression of ideas.

Forged or counterfeit documents are, however, encountered and in circulation all over the world. The same graphic design applications used in modern industry can also be used for illegitimate purposes like creating counterfeit documents. Due to the exceptional editing and design capabilities of these applications they can easily be exploited and misused to create counterfeit documents like IDs, passports or drivers licences. According to a newspaper report by Ilham Rawoot of the *Mail & Guardian*, terrorist's target fake South African passports because of the ease with which they can be faked [1]. Criminal activities such as these confirm the need for digital forensic investigations.

Similar digital forensic papers have been published that identify image forgery or tampered images [18, 19]. However, not much has been done in such research to identify whether a specific system was used during a counterfeiting exercise. Therefore, if no evidence is available for proving that a counterfeited document exists, counterfeiting criminals can potentially get away with it. It is, thus, relevant to examine a system specifically for the potential existence of counterfeit documents.

The use of graphic design applications leaves behind traces that can be revealed during a digital forensic investigation. A digital forensic investigation generally consists of the following phases consisting of the acquisition, examination, analysis and reporting [2]. Assuming that an individual is

suspected of creating counterfeit documents, the regular process of acquisition is followed. The phases of acquisition and reporting are generally similar in different cases; hence the emphasis is on the examination and analysis phases.

This paper identifies the digital traces left behind when certain graphic design applications had been used. This is achieved by associating the possible actions taken during document creation with the traces left behind. The source of potential evidence referred to above equates to the results of possible actions (i.e. document scanning, editing, saving and printing) taken during document creation. Most of this evidence would originate from the application log files, referred to as system-generated evidence.

The work covered in this paper continues from previously-published work by the authors on “User-generated digital forensic evidence from graphic design applications” [16]. The mentioned paper elaborates on gathering potential evidence on the actual files with counterfeit value created by the counterfeiter intentionally. As opposed to the previous paper [16], the focus of this paper is on the files generated by the graphic design application itself, mostly for the purpose of metadata that would hold potential evidence. Another similar paper published by the authors titled “Finding digital evidence from graphic design applications” [17], presented digital evidence on a high level.

To address the problem, the authors focus on identifying the digital forensic information that shows whether a document was created through the mentioned four actions. In doing so, a link with the potential criminal may be established. However, it is not the aim of this paper to link the crime to an actual person but merely to establish that a counterfeit document was indeed created.

The remainder of the paper is structured as follows: Section two starts off with some background on digital forensics, followed by a brief discussion on graphic design applications. Section three presents the system-generated digital forensic evidence gathered by means of two experiments, while Section four is an evaluation and discussion of the evidence extracted from the graphic design applications. Section five serves as conclusion to this paper.

2. BACKGROUND

In part A, the authors discuss the studied literature on digital forensics, followed by an explanation of digital evidence and a definition of digital forensic artifacts. Part B contains a brief discussion of the three Adobe graphic design applications used for the purposes of this study.

2.1 Digital Forensics

At the Digital Forensics Research Workshop (DFRWS) in 2001, digital forensics was defined as the use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorised actions shown to be disruptive to planned operations [3]. To reconstruct and understand what happened on a system in the past, data has to be gathered and analysed in a transparent manner.

A digital forensic investigation focuses on finding digital evidence when a computer or network security incident has occurred, or locating data from systems that may form part of some litigation, even if such data has been deleted. In this context, evidence is critical and any items that can be considered to be of evidential value should be identified and collected [4].

Computer evidence or digital evidence is defined as any hardware, software or data that can be used to prove one or more of the ‘who, what, when, where, why and how’ questions pertaining to a security incident [5]. Computer evidence furthermore consists of digital files and their contents that are left behind after an incident. Casey defined digital evidence as any data that can be used to establish that a crime was committed or that can prove a link between a crime and its victim or an offender [6]. Digital

evidence consists entirely of sequences of binary values called bits [7]. It is important to keep in mind, however, that the evidence should be presented in its logical form in court or at a disciplinary hearing.

Traces left behind from the use of an application or operating systems are referred to as digital forensic artifacts [8]. An examiner reveals the truth of an event by discovering and exposing the remnants of the event that have been left on the system. Because of the loaded legal connotations binding the term ‘evidence’, the term ‘artifacts’ is preferably used instead to refer to these remnants. When a perpetrator tries to remove these artifacts, it potentially leaves other artifacts behind. For example, in trying to remove log files from a system, one typically might use a removal tool, which leaves additional traces indicating that a log removal tool was used. The scattered evidence inside a system can indicate what has happened for a particular digital forensic investigation.

Application artifacts left by installed applications can be an excellent source of potential evidence when performing an analysis. An artifact, however, does not become evidence unless its ability to prove a fact has been established [9]. Hence it is necessary to reconstruct events that occurred by gathering all the possible digital information from a system.

The amount of research and development that has been undertaken in this field has not, to date, focused on the skills and of graphic design software, which is a particular area that is nearly always exploited for the purpose of creating counterfeit documents and images. Most research work that has been undertaken up till now has concentrated on image forensics, which is the kind of investigation that is able to determine whether or not an image as been forged or tempered [18,19].

Lien [18], proposed a method that uses a pre-calculated resampling weighting table to detect periodic properties in error distribution within an image. The errors in the distribution within an image are used to determine if the image has been forged. Stamm [19] proposed a method to detect contrast enhancement and addition of noise in *jpeg* compression images. Changes in contrast and noise within an image are determined through the use of an algorithm that calculates pixel values within the image. The values are then used to detect forgery within the image. Cohen [20] proposed a method that determines characteristics associated within digital still camera images to determine the origin of the image. The characteristics are compared to the exact replicas and derivatives of other statistical images to detect forgery. These, [18, 19, 20], and other related work focus on determining forgery using statistical data within the image [21, 22, 23, 24].

Very little of the research carried out to date has specifically investigated the ways and means in which documents are counterfeited. These ways also include the methods and procedures that can be used to detect such activities from graphic design applications, which is the focus of this paper.

How and where evidence is located differs, depending on the crime being investigated, the platform (operating systems) and the application used to commit the crime.

2.2 Graphic design applications

Of the many graphic design applications currently available in the industry, Adobe Systems Incorporated is regarded as the largest software maker in the graphic design software category [10] and hence the reason for focussing on graphic design software from Adobe Systems for this research. Adobe Systems Incorporated owns software technologies that are used for online transactions, business applications and social technologies [11]. The case study for the current research was therefore conducted with Adobe graphic design applications, namely Photoshop and In-Design.

3. DIGITAL FORENSIC EVIDENCE GATHERED FROM GRAPHIC DESIGN APPLICATIONS

In this section, the authors start off by explaining the research method used in this study to create the counterfeit documents, referred to as the experiments. Secondly the authors illustrate the results

obtained from the experiments, referred to as the gathered digital forensic artifacts. A summary elaborating on the results concludes this section.

3.1 Experiments

‘System-generated digital forensic artifacts’ refer to those artifacts created by the application without direct user intervention, while ‘user-generated digital forensic artifacts’ refer to artifacts intentionally and directly created by the user. The latter are not analysed in this paper.

The research experiments were conducted in two stages. The first experiment was conducted to simulate the activities that can be performed by an offender and is referred to as the ‘counterfeiter experiment’. The second experiment was carried out to trace the activities of the offender and is referred to as the ‘investigator experiment’. An explanation of the two experiments follows.

3.1.1 Counterfeiter experiment: Creating the counterfeit documents

The researcher created approximately three hundred dummy counterfeit documents by using the graphic design applications that were discussed earlier in this text. The motivation behind the creation of approximately three hundred documents is as follows. These documents were created during the experiment by editing the following four components within a South African Identity Document (ID), passport and drivers license: the barcode, fingerprints, signatures, and photographs of human faces. This required a combination of twenty four options ($4! \text{ (Factorial)} = 24$) on eleven examined file types. The combination for all file types equalled two hundred and sixty four (24×11), and included a few extra repetitions for clarity, yielding almost three hundred documents. This was so that the authors could be able to notice the difference or the changes to the digital forensic artifacts as more documents are created. Different application versions usually bring about more application capabilities and enhanced digital tools which can result in potential changes to digital forensic artifacts. These changes will be explained later in the results section.

Since most graphic design application users prefer the latest editions, the most recent version of Adobe, CS5, was used for this study as the base experiment. Further experiments were carried out on CS3 and CS4 for comparative purposes. Three different computers were used, each with a different Adobe version installed on it. The counterfeit documents were created by performing the actions mentioned before (scanning, editing, saving and printing). The ‘platform’ refers to the operating system on which the counterfeit documents were created. According to software reviews in 2011, the Windows operating system is still ranked most popular [12, 13] and the analysis of digital forensic artifacts was consequently conducted on a Windows 7 platform.

3.1.2 Investigator experiment: Searching for the evidence

Once the counterfeit documents had been created, experiments were carried out to search for pertinent evidence left behind from the use of the graphic design applications. The operating systems’ registry editor tool, ‘regedit’ was used to search for associated registry entries, while a hex editor, Winhex [14] was used for analysing the binary data of the log files.

To respond to the problem stated earlier, that there are no digital forensic investigation software tools available yet to investigate crimes where graphic design applications can be used for creating counterfeit documents; four possible actions taken during the creation of a document were used as a hypothesis to gather digital forensic information related to the graphic design applications. The analysis is formulated to find the digital forensic information that indicates that the actions (scanning, editing, saving and printing) had indeed taken place. By tracking the actions performed, an investigator is able to conduct a systematic investigation aimed at acquiring not only the files used to create the document, but also the actual documents created to be used as potential evidence. For example, if the document was scanned, then the next step would probably be that it was edited. If

never scanned then probably it was edited only. In the end, it becomes possible to state if the document created was a counterfeit document or not.

If none of the four actions were taken, then there is no need to ascertain whether the application was used for document creation. An illustration of the results from the experiments follows.

3.2 Results from the experiments: Gathered digital forensic artifacts

The discussion that follows highlights the digital forensic artifacts found in graphic design applications where the source of the potential evidence is mainly system-generated and results derive mostly from application log files.

Experimental results obtained from digital forensic artifacts related to the four actions (scan, edit, save and print) are elaborated on in each of the subsections to follow.

3.2.1 Artifacts related to document scanning

Generally, when one attempts to create a fraudulent document, an original document has to be acquired to imitate or copy its identity. Scanning is a common option that results in the original document being available on computer for digital editing. The different models of scanners that are currently available use various software packages for executing scan commands. For the purposes of this research, the focus is therefore on commands generated from within the graphic design application and used for editing the scanned document.

Adobe Photoshop has the capability to scan a document using the ‘import WIA support’ document menu option. The document scanned is loaded into a destination folder as prompted. The application creates a folder, saves the scanned image and opens the scanned image in the application.

After a document is scanned, the application records the digital artifact (evidence for scanning) into one of its log files named *Adobe Photoshop CSX Prefs.psp* located in *C:\Users\<username>\AppData\Roaming\Adobe\Adobe Photoshop CSX\Adobe Photoshop CSX Settings*. The *X* in *CSX* represents the version of the graphic design application, which can be 3, 4 or 5. After the authors analysed this *psp* log file, they identified an **entry recorded of the location of the scanned file** at certain address offsets to be discussed in section 3.3 summary. Through examining this location, the authors were able to identify the copies of the original documents scanned for possible counterfeiting.

Adobe In-Design is not capable of scanning a document. In this case, if the application used cannot scan a document. Then the user could use the scanners own software, this means that the scanned document will be loaded into the application through the “place” function. As long as the application user has inserted the scanned document into the graphic design application, it is possible to trace the particular image inserted as shall be described in the sub section “artifacts related to document editing”. Even if not all actions are exercised(scan, edit, save and print), the traces obtained from any recognised actions are used to determine, for example what was inserted in the document and what the saved document created is. This would enable an investigator to visualise these aspects and determine if a counterfeit document was created.

After scanning, the regular process followed by a potential criminal is to edit the acquired document in a bid to falsify its content. This editing process is discussed in the next section.

3.2.2 Artifacts related to document editing

Document editing is one of the important stages of creating a counterfeit document as it allows one to insert objects of interest. For example, a human face, a bar code or a fingerprint can be inserted in the scanned document. A number of editing actions can be performed, including typing, colouring or drawing. Our focus is on editing by insertion of an image or object, as this can later be used to determine if the document created was counterfeit or not. Regarding inserted objects, experiments

were executed to establish what can be inferred from a system that indicates to the examiner what was inserted and from which location it was inserted. The terms ‘inserting’, ‘attaching’ or ‘placing’ an image are considered to refer to the same action, though called differently in various applications. In this paper, the term ‘inserting’ is used henceforth.

The same log file, *Adobe Photoshop CSX Prefs*, **records digital information with the name of the inserted file and the location from which it was inserted.**

Adobe In-Design can also perform the action of inserting an image into a document. In-Design log files consist of *FindChangeData*, *FontMaskCache*, *In_DesignDragDrop* and *idletask*. This application records digital artifacts for editing entries into one of its log files. The log file named *InDesign SavedData* (without a file extension), which is located at *C:\Users\ <username>\AppData\Local\Adobe\InDesign\Version 5.0\Cache*, contains the information that indicates the name of the location from which an image was inserted. Unlike Adobe Photoshop, Adobe In-Design only **records the folder location or the path of the inserted images**, and not the full name of the inserted image.

From these locations, the authors were able to obtain the actual images used during document editing, for example, images of a human face and fingerprint images. These images are essentially necessary for counterfeit investigations as they can be used for compare to the images within the suspect counterfeit document.

3.2.3 Artifacts related to document saving

Once a document has been edited, the user (or potential criminal) usually needs to save it either for later printing or further editing. In this section the authors examine what is found in the system relating to saved documents. This information is vital as it can point to an examiner the name of the potentially fraudulent saved file and where the file was saved to. If the file was deleted or moved, search commands can also be generated based on the names of the files saved. This is done by specifying the name of the file when searching, thereby extending the search filter or search domain during an investigation.

Adobe Photoshop log file records the digital artifacts that indicate saving entries. The same log file, *Adobe Photoshop CSX Prefs*, **contains information about the name, location and type of the saved file.**

The log file *InDesign SavedData* **contains information about the name and type of the file** that has been saved, as well as **the location to which the file was saved.**

In both cases, the names are arranged in order of the last saved file first. From this information the authors managed to obtain the documents created by the graphic design application and recognise the ones which are counterfeit documents.

Adobe Photoshop records both the name of the ‘saving folder’ location and the full name of the saved file. The name of the ‘saving folder’ is recorded in the beginning of the log file, while the entry with the names of the saved files appears towards the middle of the log file. It is noted that the log file records a maximum of 22 entries of saved files. As more files are saved, the log file overwrites the older entries with new entries. Adobe In-Design records an unlimited number of saved documents.

The digital artifacts for saved documents can be verified or compared to the registry entries. Values for the visited directories are acquired from the registry key *HKEY_CURRENT_USER\Software\Adobe\Photoshop<version #>\VisitedDirs*. Generally, saved files from any graphic design application can also be verified or checked by looking at recent documents available in folder *C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Recent*.

3.2.4 Artifacts related to document printing

Printing is one of the last stages of counterfeit document creation. A user might need to create a hard copy of the edited document so that it can be used in a physical environment. Unlike scanning actions, printing actions can be commanded from all the graphic design applications in question via the print menu command. The artifacts illustrated in this section are valid for any of the examined graphic design applications. To locate which printer(s) are used to print a document, one uses the registry entries below. The registry keys from which a list of printer connections can be established are the following:

- (1) *HKLM\soft\Adobe\Photoshop\11.0\Plugin path.*
- (2) *HKEY_CURRENT_CONFIG\System\CurrentControlSet\Control\Print\Printers*
- (3) *HKEY_USERS\<username>\Software\Microsoft\WindowsNT\CurrentVersion\PrinterPorts*
- (4) *HKEY_USERS\<username>\Software\Microsoft\Installer\Products\<productid>\SourceList*

After establishing the names of the printers from the above, the physical existence of the printers can be verified. This usually assists an investigator in cases where the actual printers have been removed. Physical printers are necessary in an investigation so as to match the digital evidence to the actual printer for supporting a case during court proceedings.

For each print job, two spool files are generated by the operating system located in *C:\Windows\System32\spool\PRINTERS*. The first is *XXXXX.shd* and the second is *XXXXX.spl*, where *XXXXX* represents the job number in decimal format. Analysing the binary data of these files indicates the name of the spooled document. Additionally, print jobs that were queued to print but have not actually been printed yet can also be found within print spools. Table 1 shows the recognised printing artifacts including examples.

Table 1 Address offsets for printed documents

Recognised printing artifact	Spool file containing artifact	Address offset for recognised artifact (in HEX)	Example
Name of printed document	spl	0X20	<i>Johnstone_passport_final_edit.psd</i>
Name of printer	shd	0X88	<i>HP Laserjet 2605_2605dnPCL</i>
Name of printer (repeat)	shd	0X3B0	<i>HP Laserjet 2605_2605dnPCL</i>
Name of the application that generated the print request	shd	0X2120	<i>Adobe Photoshop CS5</i>
Username and name of file	shd	0X2400	<i>Robert_graphics_editor. Johnstone_passport_final_edit.psd</i>

The column and row headings for Table 1 are briefly explained for the sake of clarity. *Recognised printing artifact* is the name of the digital artifact obtained from the stated print spool file (column *Spool file containing artifact*). *Address offset for recognised artifact* represents the address pointer in hexadecimal format for the digital artifact, pointing to the named artifact contained in the spool file. *Example* is an example of a digital artifact for the recognised printing artifact. *Name of printer* is the address offset where an entry of the name of the printer that generated the print job can be found, and

this entry is repeated at another place in the *shd* spool file as shown in the second column *Name of printer (repeat)*. The reason for this repetition is not known, however, as far as digital forensic evidence is concerned, the repetition merely confirms again that the printer that was indeed used. *Name of the application that generated the print request* is the offset of the name of the application that generated the print job. *Username and name of file* is the address offset of the name of the user that generated the print job and the name of the printed potential counterfeit document (evidence for printing).

3.3 Summary

A log file may consist of thousands of pages of binary data, of which only a few pages will contain the required digital forensic artifacts, which, in addition, may be scattered throughout these few pages. Figure 1 shows an example of an Adobe log file, indicating a path recognised for scanned documents.

One can use a hex editor to scroll, for example, approximately 60% down the log file consisting of thousands of pages to reveal the evidence that is required. This can result in wasting too much time and, ultimately, running the risk that critical evidence being omitted from the search.

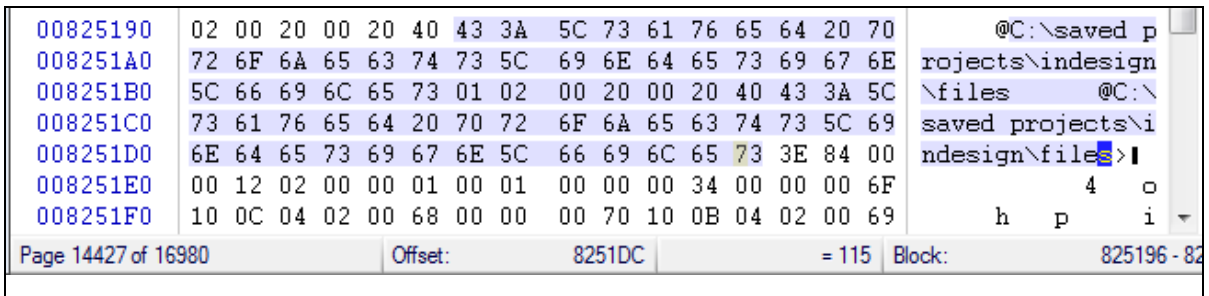


Figure 1 graphic design application log file containing 16980 pages

Another reason for recognising the locations of digital forensic information is that the digital forensic artifacts from the log files do not make use of evidence identifiers such as prefixes and tags. (Evidence identifiers are discussed in the previously mentioned paper by the authors [16]) In other words, the investigator does not know what to search for using keyword searching. The chart presented in this section guide the investigator to look for this evidence at a pre-determined location, for example, about six tenths (or three fifths) down the file. **It is therefore necessary to identify the location of this information by making use of radar chart** in order to pinpoint where the evidence can be found within the log file. Figure 2 illustrates the distribution of the digital forensic artifacts within the Photoshop *psp* log file.

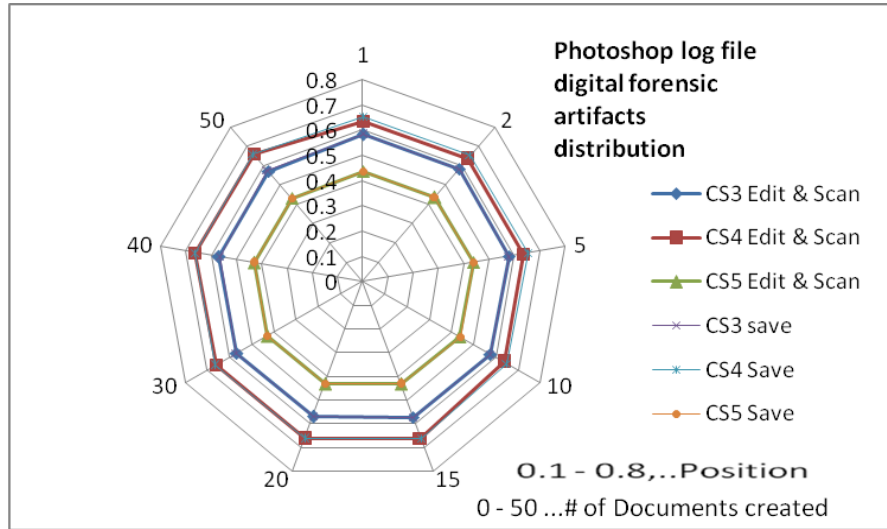


Figure 2 A graphic illustration of digital artifacts distribution in a Photoshop log file

The chart in Figure 2 shows that the digital forensic artifacts are located mostly in the middle of the log file for any action. In this chart, the centre represents the beginning of the log file represented by a 0 and the outer edges represent the end of the log file represented by a 1. The numbers one to fifty represent the number of counterfeit documents created. Such a chart helps the examiner to appreciate that they can access most of the information at the same location inside a log file. Figure 3 illustrates the distribution of digital forensic artifacts within the log file, *Indesign Save data*.

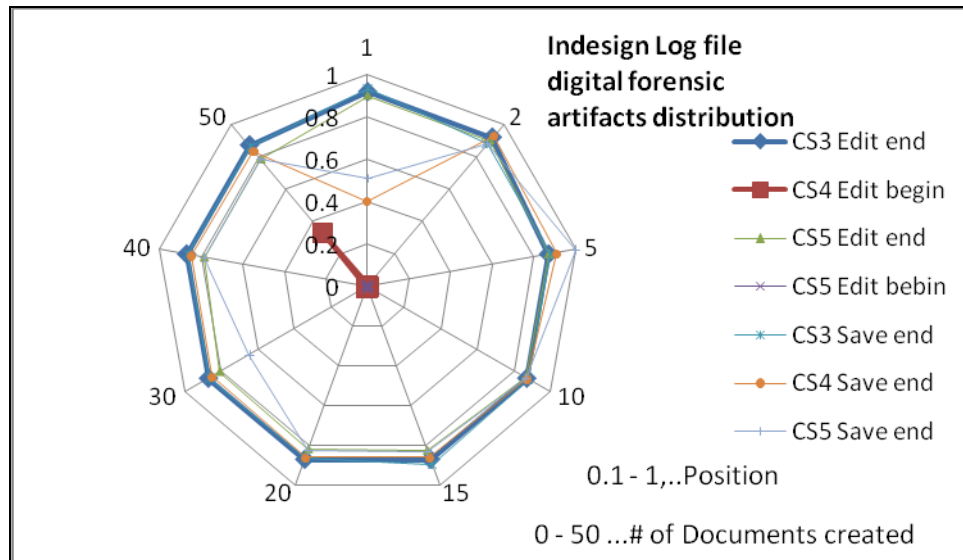


Figure 3 A graphic illustration of digital artifacts distribution in an Adobe In-Design log file

The radar chart (figure 3) shows that most digital forensic artifacts from the Adobe In-Design log file are located towards the end of the file. Some, however, are scattered all over the file from the beginning until the end. It can be recognised that the radar charts do not contain printing distribution; this is because the printing artifacts outlined in section 3.2.4 are fixed address offsets as displayed in Table 1.

Based on the experiments conducted in this study, the authors managed to establish the locations to which scanned documents were saved. In these locations one could discover several other counterfeit documents that were scanned. In respect of the action of *editing*, the authors established the names, file

types and file locations of inserted objects. By tracking the latter, the actual insertions were recognised by means of fingerprints and human face images inserted into the counterfeit documents. The *saving* action enabled the researchers to recognise potential digital evidence that reveal the location of the actual counterfeit documents created. The printing action exposed registry and spool files that revealed the names of the printers that had been used for document printing, as well as the names of those documents printed.

4. DISCUSSION

Given that a digital forensic investigation was initiated into a suspected counterfeit document creation crime, and given that the document was generated using a graphic design application, a digital forensic examiner can use the identified digital forensic artifacts to establish the route along which the document was created and corroborate the gathered evidence. For example, the digital forensic examiner is able to discover the human face, fingerprint, and/or bar code images that were used to create the counterfeit document. The inserted image can then be compared to match the image in the suspected counterfeit document. Such evidence can be presented in a court of law for prosecution. Presenting proof of the actions taken during the process of document tampering (scanning, editing, saving and printing) provides valuable support when a case of counterfeit document creation is brought before the court as evidence indicating how the document was created and what entities were used to create the document. In the end, determining if the system was used for counterfeiting purposes.

These results are essential for a digital forensic examiner to find and locate digital evidence related to the creation of counterfeit documents. This increases the transparency and reliability of the investigation process in cases where the crime tool was a graphic design application.

5. CONCLUSION

As mentioned before, that previously-published work, i.e., user-generated digital forensic evidence in graphic design applications [16], involves detecting a counterfeit document directly created by the user. That research lead to another question whether there exist system-generated evidence indirectly created by a system rather than directly created by a user, which then led to this paper, which identifies if a system was used for counterfeiting purposes.

The gathering of system-generated digital forensic evidence is effective in addressing cases where counterfeit document editing is largely associated with particular graphic design applications. Although this approach addresses only case studies involving Adobe products, the same can be done for other graphic design applications and for many other types of applications. A shortcoming of the approach is, however, that it does not tackle issues where the user only edits a hard copy, or scans and prints without using any pre-installed graphic design application. Another drawback of this approach is the fact that this exercise needs to be carried out on all new graphic design applications in order to detect where exactly potential evidence can be found within such a new graphic design application.

The techniques discussed in this paper can, however, be incorporated in commercial digital forensic tools like FTK or Encase, or it can possibly be used in the design of a new digital forensic investigation tool capable of specifically detecting counterfeit document creation. For example, a tool can be created similar to the 'porn detection stick' created by Paraben [15], which is a thumb drive device that scans and detects pornographic content on a computer.

Future research can include administering this process to other graphic design applications such as CorelDraw and also to other types of applications that could similarly be used to commit digital document fraud.

REFERENCES

- [1] Rawoot, I. (2011). Terrorists favour 'easy' fake SA passports, *Mail & Guardian*, 17 June 2011.
- [2] U.S. National Institute of Justice (2001). *Electronic Crime Scene Investigation Guide: A Guide for First Responders*.
- [3] DFRWS. (2001). A roadmap for Digital Forensic Research. *Digital Forensic Research Workshop*, p. 16.
- [4] Jones, A., and Valli, C. (2008). *Building a digital forensic laboratory*, Burlington, Elsevier pp 285.
- [5] Solomon, M.G. Barrett, D., and Broom, N. (2005). Computer Forensics Jumpstart. *Sybex*, London, p. 51.
- [6] Casey, E. (2000). *Digital evidence and computer crime*, London, Academic Press, p. 10.
- [7] Cohan, F. (2010). Towards a science of digital forensic investigation. *IFIP Advances Digital Forensics VI*, China, p. 17-35.
- [8] Altheide, C., and Carvey, H. (2011). *Digital Forensics with Open Source tools*. Elsevier, MA USA p. 2.
- [9] Zelkowitz, M. V. (2009). *Advances in computers information security*. Academic Press, Elsevier.
- [10] Bloomberg News. (2011). Stocks weaken after Fed Statements. *The New York Times*, 12 June 2011.
- [11] Tech Specs. (2013). Retrieved from <http://www.adobe.com> on 15 January 2013.
- [12] Gartner Research. (2013). Which operating system will be 2011's bestseller. Retrieved from <http://www.gartner.com/technology/research> on 15 January 2013.
- [13] Top Tech News. (2010). Windows 7, Office Drive Record Microsoft Revenue. Retrieved from http://www.toptechnews.com/story.xhtml?story_id=11300CM9DYVG on 15 January 2013.
- [14] Winhex. (2012). Retrieved from www.x-ways.net/forensics on 13 June 2012.
- [15] Porn detection stick. (2012). Retrieved from www.paraben-sticks.com/porn-detection-stick on 9 August 2012.
- [16] Authors, (2012).
- [17] Authors, (2012).
- [18] Lien, C. C. (2010). Fast forgery detection with the intrinsic resampling properties. *Journal of Information Security*, 1(1), 11-22.
- [19] Stamm, M. C. (2009). Forensic detection of image tampering using intrinsic statistical fingerprints in histograms. *APSIPA Annual summit and conference*, Japan, 563-572.
- [20] Cohen, K. (2007). Digital Still Camera Forensics. *Small Scale Digital Device Forensics Journal*, 1(1), June 2007, 2-8.
- [21] Farid, H. (2009). Image forgery detection. *IEEE Signal Processing Magn*, 16-25.
- [22] Bayram, S. Avcibas, I. Sankur, B and Memon, N. (2006). Image manipulation detection. *Journal of Electron Imaging*, 15(4), 41-52.
- [23] Wang, J. (2010). Image forensics based on manual blurred edge detection. *Multimedia information networking and security (MINES)*, 907-911.

- [24] Memon, N. (2012). Photo Forensics. *International workshop on information security*, New York University, p. 1-27.