FINDING DIGITAL FORENSIC EVIDENCE WHEN GRAPHIC DESIGN APPLICATIONS ARE USED FOR DOCUMENT COUNTERFEITING

by

Enos Kudakwashe Mabuto

Submitted in fulfilment of the requirements for the degree

Magister Scientiae (Computer Science)

in the

Faculty of Engineering, Built-Environment and Information Technology

at the

University of Pretoria

July 2013

FINDING DIGITAL FORENSIC EVIDENCE WHEN GRAPHIC DESIGN APPLICATIONS ARE USED FOR DOCUMENT COUNTERFEITING

by

Enos Kudakwashe Mabuto

Supervised by

Prof H.S Venter

Department of Computer Science

Magister Scientiae (Computer Science)

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 licenses or passports, among others. The products of graphic design applications, however, leave behind traces of digital information which can be used during a digital forensic investigation. Although current digital forensic tools are designed to scrutinise systems with the purpose of finding digital evidence, the tools are not designed to examine such systems specifically for the purpose of identifying counterfeit documents.

This dissertation reviews the digital evidence relating to the creation of counterfeit documents and gathered from graphic design applications. Digital evidence gathered in this way consists mainly of identifying and corroborating the counterfeiting events that occurred on a particular system. Firstly, such an analysis is accomplished by establishing linkages between the digital forensic information that has been gathered and the specific actions that were performed when the counterfeit documents were created. Such actions comprise scanning, editing, saving, and

printing. The researcher is able to compile a dossier of the digital forensic information that is generated by such actions by analysing the files that were generated by making use of a particular graphic design application for document creation. Secondly, the researcher extends the analysis to the actual files created by the application user. These files can be used as evidence to establish linkages between the content of the counterfeit documents that are being investigated and the document editing actions that are necessary for creating such documents. The researcher gathers digital forensic information of this kind by analysing the different file types that are associated with these applications. The researcher then gathers the associated timeline evidence separately by means of a third analysis that identifies timestamps from the application's system files and evidence files. The researcher is then able to draw a timeline from the timestamps to illustrate the sequence of events that occurred. From the digital evidence gathered in this way it is possible to propose a two-pronged counterfeiting investigation process. This proposed investigation process is application and platform independent. The researcher concludes the study by transforming the model into a working prototype by demonstrating how the prototype is capable of analysing and extracting digital forensic information from certain graphic design application file types and log files. Such a prototype is capable of identifying the system that was utilised for counterfeiting particular documents or identifying whether a specific document is counterfeited or not.

Keywords

Counterfeit documents, Digital evidence, Digital forensics, Digital forensic artifacts, Graphic design applications.

Acknowledgements

I would like to express my sincere thanks to the following people for their assistance during the production of this dissertation:

- My brothers, Ernest and Emmanuel Mabuto – Thank you for your loving financial and moral support throughout my academic studies.
- Hein Venter – Thank you for your guidance, support, continuous motivation, kindness, and patience.
- My parents, Enock and Elizabeth – Thank you for your support and patience when you most needed my assistance.
- My diaspora family – Ephraim, Edson, Emelda and Edward – Thank you for your spiritual and financial support throughout my studies.
- Isabel and Roger – Thank you for helping me to transform this text into standard academic English.
- Kamil and Alex – Thank you for our face-to-face discussions, and for assisting me with my research.
- Nickson and Emilio – Thank you for your laboratory reviews, particularly during the period in which I was writing my dissertation.
- Leandi, Kimberly and Angela – Thank you for your wonderful administrative support and encouragement, from your loving welcome at the beginning to the point where I was ready to submit my dissertation.
- ICSA – Thank you for your loving contribution towards my research, for your top-class research skills, and for assisting me throughout the course of this study.
- Above all, to God, for the above mentioned individuals.

Table of Contents

List of Figures

List of Tables

List of Equations

Part I: Introduction

CHAPTER ONE INTRODUCTION

1.1 Introduction

Most corporations rely on information technology systems (ITSs) to establish continuity in the conduct of their day-to-day business in the spheres of production, administration, marketing and other essential services. This kind of reliance has become indispensable over the years because of ITS's ability to store, process and exchange data much more quickly and reliably than human beings are able to do. The developers of software have over the years adjusted to this need for efficient and innovative technologies by designing complex systems that are exactly suited to the conduct of business in all kinds of enterprises and industries.

The same creativity, innovation, and rate of growth are evident in the ITSs that are used in the field of graphic design applications. Some of the most notable graphic design applications that have been developed to date are those that fulfil the need that operators have for a software that facilitates the creation, layout, composition, and editing of digital art. Digital art has become indispensable in so many different fields such as, for example, the creation of commercial artwork, flyers and banners, and for the expression and sharing of the vast array of artistic concepts that we encounter in contemporary life.

The exponential growth in the production of innovative designs and graphic illustrations has generated a number of complex software applications that are dedicated to the creation and manipulation of graphic art that cannot easily be created by any other medium. But graphic design applications are also used for organizing, creating, enhancing, and sharing of the graphic designs that have become an indispensable part of all profit-making enterprises that rely on pictorial, design and graphics products – whatever the medium in which they are cast. The industries in which such products are indispensable include those that are concerned with advertising, newspaper and magazine production, book production, architecture, fashion and design, project management, and many others that depend upon being able to manufacture graphic designs by making use of graphic design applications. Standard features in graphic design applications include enhancement tools such as paint brushing, vector drawing, digital pen work, and pencil drawing functions, among many others. These enhancement tools are used

to facilitate the creation of unique art for company logos, magazine advertising or computer-aided design, to mention only a few.

The widespread development and utilisation of graphic and computer-aided design applications has, however, resulted in an exponential increase in the production of counterfeit documents throughout the world wherever this technology is available. In a report written by Ilham Rawoot for the South African Mail and Guardian newspaper, the author states that terrorists demonstrate a distinct preference for faking South African passports because of the ease with which they can be counterfeited [1]. A story along similar lines by Jackie Bargas of the International Business Times pointed out that a Brazilian man had attempted to open a bank account by using fake identification documents [2]. In another report by Laura Blasey, students were identified to have counterfeited basketball tickets using Adobe Photoshop to gain access to basketball stadiums [112]. Similar reports also pointed out the use of graphic design application in counterfeiting various documents [113] [114]. These and numerous other examples from all over the world confirm that the counterfeiting of important documents has become crucial skill for supporting criminals in virtually every sphere of human endeavour in which human identities need to be validated for security and administrative purposes. Counterfeit documents often enable crimes of a serious and far-reaching nature such as, for example, terrorism, fraud, money laundering, and theft. All these crimes are made possible when the criminal concerned has acquired a counterfeit identity document that is sufficiently sophisticated and plausible to escape routine detection. Counterfeit documents can avail terrorists in particular because they enable them to plan, operate, and execute their activities without the prior knowledge of the authorities. It is therefore a matter of the utmost concern that those individuals and government agencies that are concerned with issues of safety and security possess the means for identifying counterfeit documents by acquiring an accurate understanding of how these documents are created in the first place.

The same graphic design applications that are used in industries today are eminently suitable for illegitimate purposes such as the creation and counterfeiting of vital documents. Graphic design applications have a platform that permits users to create documents from scratch, to edit original copies of documents, and to process and print the documents thus created. The ever-increasing extent of criminal activities throughout the world, such as the mentioned, give us

some idea of how important it is for security operatives to be able to undertake digital forensic investigations in any area where they suspect that counterfeiting may have been perpetrated.

Once a suspected computer-aided criminal event such as the creation of a counterfeit document has occurred, it becomes necessary to employ an expert to investigate, assess, and evaluate the situations, and the events that gave rise to suspicion in the first place. Digital forensic examiners are specifically responsible for evaluating crime scenes in which computers or any kind of computer-related technology has been used. It is vital to remember that digital forensic investigators are constantly confronted with innovations and novelties in each new criminal scenario because of the ever-changing range of ITS products including graphic design applications which are constantly becoming available on the market. Such changes occurring for both hardware and software (in operating systems and in application systems). It therefore follows that successful and competent digital forensic examiners have to be kept abreast with all the latest developments in state-of-the-art graphic design applications. But, in spite of developments, the products of graphic design applications routinely leave behind traces that are evident to any digital forensic investigator who examines the digital product in a systematic and logical way. It is with such investigations of this study is concerned.

The remainder of this chapter comprises the research motivation, the statement of the problem, the limitations of this research, the objectives of this study, the overall layout of the dissertation, and a conclusion to this chapter.

1.2 Motivation

This research was motivated and ultimately initiated for the reasons indicated in the bulleted items that follow.

- The unprecedented increase in the number of fake IDs and passports in circulation throughout the world

Natasha Domanski of the Queens Chronicle reported that the fake ID crisis is becoming ever more severe, and that the state senator for Queens had proposed a bill that would increase the penalties applied to the creators and distributors of counterfeit documents [3]. If this bill were to be implemented, those who were convicted would have to serve between five and fifteen years in

jail instead of the current two and a half years. This same report mentioned the present increase in number of reported cases of fake IDs and passports in circulation from all over the world. All of these documents are being created and used for illegitimate and criminal purposes in every country of the world. Several reports have identified counterfeiting exercised using graphic design applications [112] [113] [114]. It is therefore necessary to investigate the ways these applications are capable of creating counterfeit documents.

- Improvements in the capabilities and sophistication of graphic design software

John Kell of Dons Newswires reported that the Adobe Systems (ADBE) fiscal second quarter showed an extraordinary increase of 54% on the broad sales gains for the quarter that ended 3 June 2011 because of the enormous success of its flagship Creative Suite programs [4]. In the same way, an expansion in the international sales of graphic design software indicates an enormous and ever-increasing public interest in graphic design software. These graphic design applications can be used not only by the non-commercial public but also by various organisations and individuals who utilise it for legitimate purposes and, maybe in other instances, for illegitimate and criminal purposes of various kinds.

- Digital forensics research in graphic design applications.

Various organizations such as The Digital Forensic Research Workshop (DFRWS) have called for more research into the field of digital forensics on the part of researchers and operators in both academia and in commerce and industry [5]. Up to date, not much research has been done in the category of graphic design applications. 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 [6] [7]. Very little of the research carried out to date has specifically investigated the ways and means in which documents are counterfeited and the methods and procedures that can be used to detect such activities.

- The need to assist the justice system to detect and counteract crime

A “NewsTime” reporter reported that, “the fight against crime is in danger of being severely retarded because the public won’t work with the police” [8]. There is urgent need all over the world for experts and ordinary citizens to assist law enforcement officers and investigators in

their fight against crime. By addressing the problem for which this research is designed to identify solutions, the researcher will provide digital forensic investigators with a sure means for gathering the kind of evidence that is necessary for detecting forgeries, and for using such evidence to secure the conviction of counterfeiting criminals in court hearings.

1.3 Problem Statement

The main problem that this research addresses is set out in the following paragraph:

Graphic design applications can be used for creating counterfeit documents such as identity documents (IDs), driver's licenses, and passports. Moreover, there are no current digital forensic tools available for specifically examining a computer system and for identifying how it was used in the creation of counterfeit documents.

In order to find appropriate solutions to this problem, the researcher devised the following sub-questions that this study was designed to answer.

1. Where would the digital forensic information be located in a system, and from which files would this information be extracted? When an operator has completed the editing of counterfeit document, a digital forensic investigator needs to be able to compile potential evidence that a counterfeit document has indeed been created. Such an investigator therefore needs to know the locations from which such essential information and evidence can be extracted.

2. What kind of digital forensic information and evidence can be gathered from the traces left behind by those who utilise graphic design applications? A digital forensic investigator needs to know exactly what kind of information he or she can extract from graphic design application files. It is also essential for an investigator to have a precise understanding of the relevance to a particular investigation of such information and evidence.

3. What current tools and techniques are available for extracting digital forensic information from any document? It is absolutely necessary for an investigator to know which tools can be used for extracting certain digital forensic information. This facilitates any investigation in which multiple tools are available for trial testing.

The problem stated is addressed in part four of the dissertation which includes the evidence gathered from graphic design applications, from chapters seven to chapter ten.

1.4 Research Limitations

The potential limitations of this research are described in the two bulleted paragraphs that follow.

- The dissertation focuses on one specific graphic suite software package (the Adobe Creative Suite), which offers a total of three separate graphic design applications (Photoshop, In-Design, and Illustrator) in each package. Although a great deal of graphic design software has been developed, this dissertation focused only on that graphic design application package (the Adobe Creative Suite) that, according to a market share report from the Wall Street Journal [9], is known to be the best selling graphic design software.

- The research only investigated the following three versions of the Adobe graphic design suite of applications: Adobe CS3, Adobe CS4, and Adobe CS5. Because most graphic design application users prefer to use the latest versions of any software, the researcher used the latest Adobe version at the time of study (CS5) as the base in his experiments. It should be noted, however, that the researcher experimented with the two earlier versions of the Adobe suite, namely the CS4 and CS3 versions for comparative purposes. Although earlier versions might have revealed different kinds of digital forensic information, they were not examined or tested in this research as most of them either cannot be installed on current operating systems or they are no longer available on the market. The practical implication of this is that this research might not be relevant in those cases in which a perpetrator used a different or an earlier version of this software.

1.5 Objectives

The objectives of this research are as follows:

- To conduct a literature review of graphic design applications, counterfeiting, and digital forensics.

- To establish what kind of digital forensic information is left behind from counterfeiters who use graphic design applications, and to indicate how this information can be used as evidence in law to apprehend and convict criminals who engage in this kind of activity.
- To identify the files that contain evidence of criminal wrongdoing when documents and images are created or generated from graphic design applications, and to determine locations of the evidence within the identified files.
- To propose an efficient digital forensic process for investigating any system that is suspected of having been used for purposes of criminal counterfeiting, namely, the counterfeiting investigation process.
- To design a prototype for validating and collecting essential digital forensic data from graphic design applications used for criminal counterfeiting purposes.

1.6 Dissertation Layout

The dissertation is divided into five parts, and consists of a total of twelve chapters, each of which is described.

The first part consists only of one chapter, i.e. the current chapter, which is the introduction chapter. This chapter includes the research motivation, the problem statement, an account of the limitations of the research, the objectives of the research, and the layout of the dissertation.

Part two of the dissertation is a literature review which consists of three chapters: chapter two, chapter three and chapter four. Chapter two explains the context and background of digital forensics as an applied science. Chapter three contextualizes and explains the background to graphic design applications. Chapter four explains the context and circumstances in which counterfeit documents are created.

In the chapter on digital forensics, frequently used terms such as *digital forensics, digital evidence* and *digital forensic artifacts* are defined and explained. The chapter also offers an overview of digital forensic techniques, software tools and anti-forensic techniques.

Figure 1.1 graphically indicates the topics that this study dealt with over twelve chapters.

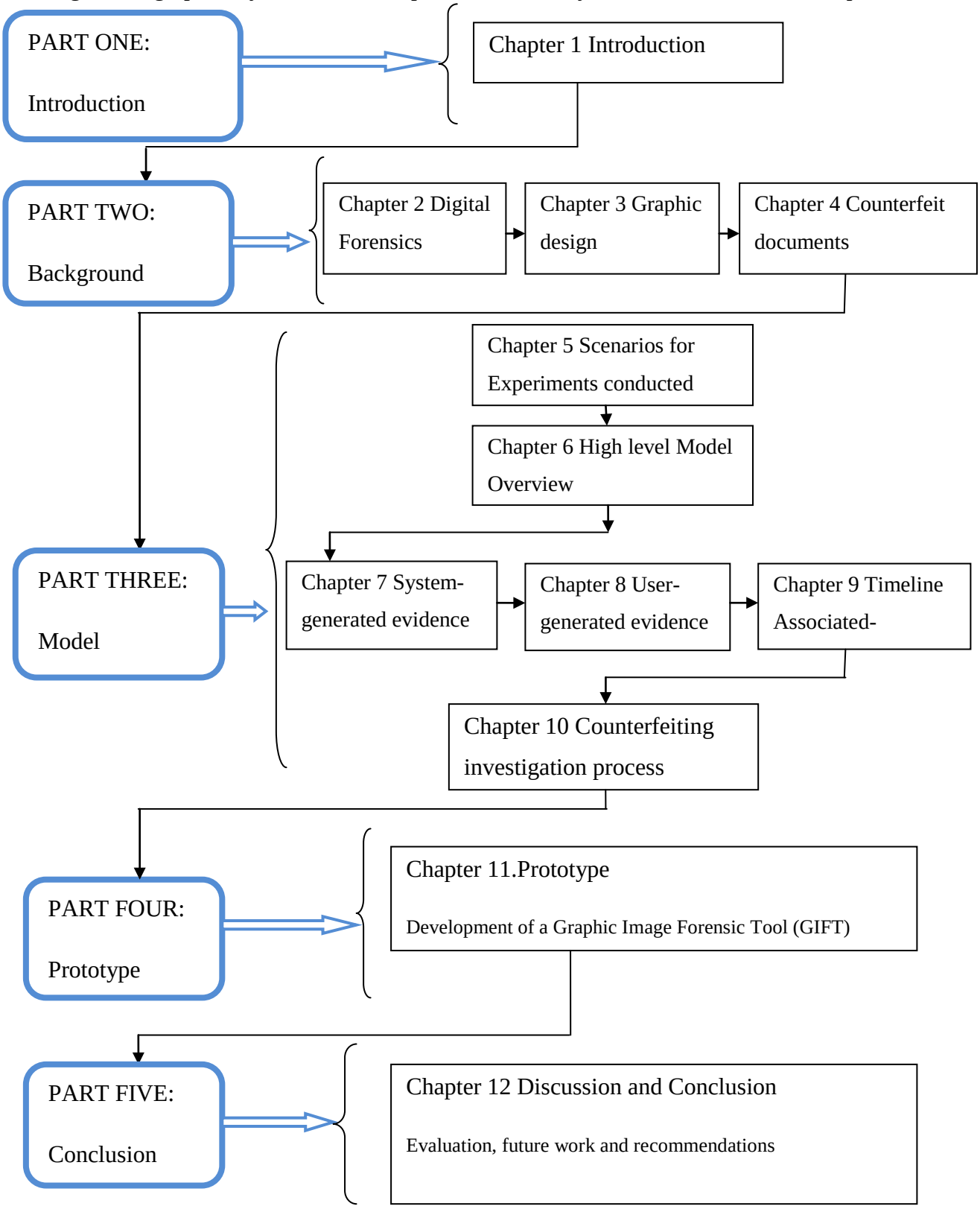

Figure 1.1: Dissertation layout

In the chapter on graphic design applications, the chapter provides an outline of the graphic design applications that are currently used in the industry. The researcher names the graphic design applications that were investigated for the purposes of this research, and explains the capabilities of each of these applications.

In the chapter on counterfeit documents, the researcher defines the terms *counterfeit, counterfeiting* and *counterfeiter*. Examples of counterfeiting are illustrated.

Part three of this dissertation discusses the contribution of this research, and it is divided into six chapters (chapters five to ten). Chapter five explains and illustrates the experiments that the researcher conducted for the purposes of this research. Chapter six provides a high-level overview of the model.

The third chapter of part three, chapter seven, explains how the evidence that is gathered from graphic design applications from the evidence that the systems themselves generate, can be evaluated. This evidence is that which the application itself generates without any intervention from a user.

The fourth chapter of part three, chapter eight, evaluates the evidence that can be gathered from user-generated evidence if such a user makes use of graphic design applications. This kind of evidence may be gathered from files that are intentionally generated by a user.

Chapter nine contains an evaluation of the evidence that determines the timeline of those activities that are undertaken to create counterfeit documents by making use of graphic design applications. The researcher indicates how evidence can be collected from timestamps to reveal when the application was installed and the last time that the application was used. Using the timestamps from the user-generated files, a timeline is drawn to illustrate the timeline of counterfeiting activities.

In the final contribution chapter, chapter ten, the researcher proposes a digital forensic process of investigating electronic counterfeiting. This proposed investigation process can be used to investigate a system that is suspected of being used for counterfeiting or a document that is suspected to be counterfeited.

Part four consist of one chapter, the prototype chapter. The chapter begins with the list of requirements that have to be met in order to gather digital evidence from graphic design applications. Also in this chapter, the prototype tool that the researcher created, is discussed and described. This tool gives an investigator the means to examine the files of graphic design applications and to extract evidence that can later be used for forensic purposes.

The final part of the dissertation, part five, consists only of the concluding chapter. In this chapter, chapter thirteen, the researcher summarises the dissertation, revisits the problem statement, and offers his final evaluation. This chapter concludes with suggestions for future research in this field and recommendations for particular research projects.

1.7 Conclusion

In this introductory chapter, the researcher discussed and described the motivation for the research, the statement of the problem, the limitations of the research, the research objectives, and the layout of the dissertation. The introduction also offered an overview of the dissertation. The following part of the dissertation consists of three background chapters: chapter two deals with digital forensics, chapter three deals with graphic design applications, and chapter four describes and discusses the problem of counterfeit documents in the contemporary world.

Part II: Background

CHAPTER TWO DIGITAL FORENSICS

2.1 Introduction

Digital forensics is an applied science that has grown extremely rapidly during the past few years as computer professionals and law enforcers are faced with the challenge of keeping up with the latest developments and advances in the field of computer technology and its many applications. The field of digital forensics has been developing in parallel to the advancement of technology over the past decade as more and more state-of-the-art tools such as tablets and smart phones have flooded the market. In the sections that follow, the researcher provides an overview of the current state of digital forensics and explains how this field of study is relevant to this research.

This chapter therefore offers a brief review of the current literature on digital forensics, the digital forensic process and includes explanations of what digital evidence and digital forensic artifacts are. After these terms have been defined, there is a discussion on image forensics, which includes discussions on related research work. Following image forensics is a brief discussion about the common digital forensics techniques in this kind of research. Thereafter follows examples and descriptions of common software tools and anti-forensic techniques applicable to this research. The chapter concludes with a summary of what has been discussed here. It should be recognised that none of the work outlined in background chapters two, three and four were developed by the researcher. The background consists of a literature review gathered from various sources.

2.2 Defining digital forensics

The Digital Forensics Research Workshop (DFRWS) of 2001 defined digital forensics 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 unauthorized actions shown to be disruptive to planned operations” [10]. The DFRWS is a non-profit-making organisation that sponsors conferences, technical working groups, and challenges in the field of digital forensics research. Numerous researchers and practitioners from across the globe from law enforcement agencies,

government, academia and industry have contributed in different ways to this organisation. The organisation was launched at a gathering in August 2001 in Utica, New York, and in 2005, F Adelstein, B Carrier, E Casey, G Richard and V Roussev were designated as the main organisers of the annual DFRWS workshop. Since that time, the organisers have undertaken the publication of a number of books that deal with important digital forensic topics, all of which are designed to disseminate important information and to assist all interested parties in academia and elsewhere to keep researchers and practitioners abreast of the latest developments in this field.

The terms *computer forensics* and *digital forensics* [10] are used interchangeably in various sources. While the term *computer fore*nsics has been in use for many years, the term "digital forensics" has become well established in the field and is now most commonly used among practitioners, educators and researchers. Conversely, the term *digital forensics* has only recently been adopted to accommodate the many new digital devices such as tablets and cell phones, which have appeared on the market and are now in wide use [10]. The term computer forensics is now reserved exclusively for investigations into computer systems that are suspected of having been used for fraudulent or criminal purposes [83]. The term digital forensics therefore covers all fields and applications from database management systems, networks, to mobile and other portable devices.

The field of digital forensics has been growing in tandem with the extremely rapid rate of technology advancement and innovation in the field of computer technology. This means in effect that as new technologies are developed and sold on the market, new digital forensic tools have to be designed and developed to accommodate these changes. Unless this is done, digital forensic investigators will not be in a position to investigate any new kind of criminal activity that is based on some other new technology in this field. Some of the common fields in digital forensics are databases, networks, multimedia, and malware forensics – among many others. The invention of some new device in one area of development for example, might well be accompanied by the release of a newer database management system that will in turn necessitate the development of a new digital forensic tool that will be capable of analysing the latest database management systems. This kind of knock-on activity exists in all fields of digital forensics.

The goal of any digital forensic investigation into a system is to find out what happened and who was responsible for a particular incident or crime [11]. By searching for evidence, facts have to be gathered. These facts are then interpreted logically and contextually to explain the incident and how it took place. Digital forensic investigations focus on finding digital evidence after a computer or network security incident has occurred or on locating data in systems that may be relevant to any subsequent litigation or legal proceedings, even if such data has been deleted. In all of these activities, it is the finding of relevant digital evidence that is the most important and salient task in any investigation. Once the necessary data has been gathered and analyzed in a transparent manner, it is the task of the digital forensic investigator to reconstruct what exactly happened on the system. This means that the methods used by an investigator need to be clear, concise, logical and scientific so that they will yield potential evidence of criminal activity. The means and methods that an investigator can use in a digital forensic process to achieve this end are discussed in the section that follows.

2.3 Digital forensic investigation process

If evidence is to be properly accumulated and processed, and if errors are to be avoided, it is necessary for an investigator to pursue an investigation in a structured and logical way that will yield the desired results. This structured method of approaching an investigation is known as *the digital forensic process* [12]. The digital forensic process can also be defined in terms of the number of steps that need to be taken from the time of the original incident alert through to the eventual reporting of findings [12].These steps may also be called the “sub-processes”. A digital forensic process can be categorized into the four phases of *acquisition, examination, analysis,* and *reporting by the National Investigation Justice* [13]. These phases are undertaken in the correct sequential order during digital investigations so that the desired result can be most efficiently achieved. This process is well established in different fields such as mobile and network forensics. Because this process has been so successfully used in numerous investigations, it is universally recognised by practitioners in this field of applied science [13] [14].

The digital forensic examiner should always know the digital forensic process and the appropriate toolsets used in a digital forensic investigation [15]. This knowledge enables an investigator to conduct an investigation that is competent, scientifically transparent and

professional. Generally speaking, the acquisition and reporting phases are the same for most investigations of this kind. The main focus of this research is therefore on the analysis and examination phases of such an investigation. An explanation of the four phases follows.

2.3.1 Acquisition Phase

The acquisition phase of an investigation describes how data is acquired from the various types of digital information sources. It is absolutely essential for data to be acquired in a manner that maintains its integrity and authenticity [11]. One of the necessities of data acquisition is that a copy of the evidence has to be made in such a way that the original evidence remains unchanged. The copy of such evidence may be found on a hard drive or in some other storage medium.

The acquisition of the data necessitates the physical material or hard drive to undergo forensic duplication or sector level duplication. Once the source of the data has been duplicated, the investigator makes certain that it is in fact an exact copy, and then uses the copy for the investigations rather than the original drive. A write blocker can be used in creating forensic duplicates of hard drives. A write blocker is a hardware tool that consists of cable connectors between the copying devices [15]. A write blocker ensures that nothing will be written to the original hard drive. Software imaging tools can also be used to create exact copies [16]. Software imaging tools are any forensic tools that are capable of imaging storage media [16]. With imaging either a physical image (bit-by-bit image) can be created of the entire physical device or a logical image can be created which comprises of active directories and files available to the operating system [11]. The latter is generally exercised during live forensics.

Data hashing is used as a way of verifying the integrity of acquired data. A digital hash conducts a mathematical algorithm of a device or file, and provides a fingerprint that is able to verify that the copied data has not been tampered with or altered. This distinctive fingerprint is then maintained throughout the duration of the case and in all its documentation. The case file is the file assigned to the case being investigated. In the following section, the analysis phase will be defined and discussed.

2.3.2 Analysis phase

The analysis phase of an investigation describes how the data is processed [13]. This phase involves searching for the required digital information and sorting the digital data that one finds into logical categories. During this phase, the investigator also becomes involved in the tracing, filtering and extracting of hidden data. Acquired links, or suspected links, can be followed to reveal whatever evidence may be hidden in the system. When large chunks of data are involved, the investigator has to reduce chunks down to essential data by means of filtering. Filtering may involve the removal or exclusion of those more common or trusted files such as, for example, the windows file system files. The investigator will then be in a position to evaluate the properties of hidden data. The investigator systematically pursues various methods of locating hidden data, and will check suspected hidden areas such as, for example, the swap space. Hidden data may also be found in places such as back up files, the registry, cache files, kernel statistics, modules, and in other external media. It is essential to scrutinise all these locations with great care because crucial evidence may be hidden in them.

In some instances it can be necessary to exclude unwanted or trusted data by using other tools. In order to do this, a hash analysis search that uses hashing tools such as SHA-1, MD5 or using CRC applications [17], may be undertaken. Cyclic Redundancy Check (CRC) is a checksum function implemented in hardware optimization [17]. These tools are able to conduct a mathematical analysis of a data storage device such as a hard drive. By comparing hash values, an investigator can exclude large numbers of files that are irrelevant to the case in hand. Enterprise forensic software such as FTK and Encase can be used to compare hash values, and the ensuing analysis is mainly concerned with locating digital media and assembling them before interpreting their contents. The following section contains a description of the examination phase of the analysed digital data.

2.3.3 Examination phase

The examination phase involves an in-depth assessment of the identified data as the content of various files are inspected to establish whether or not they contain any hidden information. This may involve identifying the properties of digital evidence and the metadata that is associated with the data itself. Each component of the recognised digital evidence needs to be interpreted

accordingly. Once the data has been carefully scrutinised and examined, it is assembled for the reporting phase, which is described in the following section.

2.3.4 Reporting phase

The report of an investigation needs to explain the route that the investigator followed in order to arrive at his or her conclusions. This document needs to explain the initial problem that prompted the investigation in the first place, the steps that were taken to solve the problem, and the conclusions that were drawn from the evidence that was obtained. Any such report needs to be carefully, concisely, and logically drafted and supported by clear reasoning so that all parties who are involved in the investigations and in any subsequent litigation or legal proceedings will be able to understand the process that was followed in order to arrive at the conclusions.

There are, however, other digital forensic processes that are involved in digital investigations [18] [19]. Other investigation processes may, for example, involve process steps such as scene detection. It should be noted that only one investigation process has been discussed in this dissertation in the context of the literature review.

Only this digital forensic process [13] has been discussed in the dissertation as it was developed by a major organisation, namely, the US Department of Justice unlike the ones developed by individual investigators and practitioners [107] [108] [109] [110] [111].

The main goal of most digital forensic investigations is to locate the evidence on which the solid legal case may be presented in court during litigation or criminal proceedings. It is necessary therefore to identify, consider, and collate any items or factors that may be of evidential value [20].

2.4 Digital evidence

Computer evidence or digital evidence is defined as any hardware, software or any data that can be used to prove one or more of the “who, what, when, where, why and how” of a security incident [15]. A security incident can be referred to as unauthorised access or attempts at unauthorised access to digital data or an electronic medium [84].In this definition, the “who” pertains to the person who committed the crime. The “what” of this definition describes an estimate or approximation of the cause of the incident. The “when” describes the date and time

of the incident. The “where” describes the location of the evidence or crime; the “why” infers the reason or purpose of the crime or incident; the “how” describes the process or method that the perpetrator followed to exercise or commit the incident. Computer evidence furthermore consists of the content in digital files of the perpetrator left behind after an incident. The data files may be of any type, and the content of the data files may indicate precisely what happened and who was responsible. Casey [12] defines 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 offender. Digital evidence consists entirely of sequences of binary values called bits [21]. Evidence in this binary format represents the evidence at a lower level of interpretation. It is essential to note that all evidence should be presented in a logical and comprehensible form in any court or disciplinary hearing so that participants can have an understanding of the evidence brought upon to court. This means that an investigator must always present the evidence in a clear, understandable and compelling form.

In the field of digital forensics, all evidence belongs to one of the three following categories: exculpatory evidence, inculpatory evidence, and evidence of tampering. Each of these types of evidence is explained in the three sections that follow.

2.4.1 Exculpatory evidence

Exculpatory evidence is evidence that refutes any allegation, theory, or hypothesis. It may be referred to as that evidence that shows, proves or demonstrates that someone is not guilty of an alleged crime [22]. This kind of evidence is usually crucial for establishing a criminal defendant’s innocence. In digital forensics, such evidence indicates that the alleged crime did not actually happen or that the defendant may not be the person who committed the crime that is being investigated.

2.4.2 Inculpatory evidence

Inculpatory evidence is evidence that supports a particular allegation, theory or hypothesis [23]. The kind of evidence that is being discussed in this dissertation is inculpatory evidence because it is evidence that supports the theory, accusation, or hypothesis that a counterfeit document was in fact created by the person who has been accused of the crime. It is always therefore a matter of

critical importance for an investigator to assemble sufficient amount of evidence to support the hypothesis of guilt.

2.4.3 Evidence of tampering

Evidence of tampering refers to that evidence which reveals that the system was tampered with or adjusted with the purpose of avoiding subsequent tracing and identification [24]. For example, perpetrators engaging in editing and adjusting certain features or properties of the system in order to obscure or obliterate any trail that might lead back to the responsible individual. Evidence of this kind demonstrates clearly that a system has actually been tampered with. Such evidence cannot be related to any theory or hypothesis as described for exculpatory and inculpatory evidence.

Traces that are left behind from the use of an application or from an operating system are referred to as *digital forensic artifacts*. Such traces clearly reveal what has happened at an earlier stage.

2.5 Digital forensic artifacts

An examiner reveals the truth of an event by discovering and exposing the remnants of all the relevant events that were left on the system after an incident occurred. In digital forensics, these remnants are known as *artifacts*, and they can also be referred to as *digital evidence* [25]. But because of the loaded legal connotations associated with the term “evidence”, the researcher prefers to use the term “artifacts” instead. *Artifacts* are traces that are left behind in a system as a result of that system's activities and events that took place on the system, whether or not they be innocuous [25]. Any attempt to remove such artifacts generates other artifacts in their place. Thus, for example, when someone tries to remove log files from a system, he or she has to use a removal tool, and this leaves particular traces that indicate that a log removal tool was used. The evidence scattered throughout a system indicates very clearly what happened at an earlier date on the system, and all these residues and traces provide crucial evidence during the conduct of a particular digital forensic investigation.

The application artifacts that are left by installed applications are an excellent source of potential evidence during an analysis. It should be noted, however, that an artifact does not

constitute evidence unless its ability to prove a fact has been incontrovertibly established [26]. It is therefore necessary to reconstruct all the events that occurred by gathering all the necessary digital information from the system that was used by the alleged perpetrators.

It is important to note that the “how” and “where” evidence involved in an investigation will differ from one investigation to another, depending on the particular crime that is being investigated, and the platform (operating systems) and the application that were used to commit the crime. The following section, discusses image forensics, a field related to the research conducted in this dissertation.

2.6 Image Forensics

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 [100] [102]. Wang proposed a method for detecting image forgery using intrinsic properties from a resampling scheme [100]. The scheme determines the portions of an image that have been tempered by running samples of photographic images and detecting periodic patterns using an image forgery detection algorithm. Another technique by Anthony et al [102] uses object distance and internal depth to aid detecting image forgery in images. An equation is derived and used for predicting distances of internal depth within an image which is used in detecting image forgery in stereoscopic (3D) Images [102]. Cien [95], proposed a method that uses a pre-calculated resampling weighting table to detect periodic properties for error distribution within an image. The errors in the distribution within an image are used to determine if the image has been forged. Stamn [96] proposes 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 [97] describes a method that determines characteristics associated within digital still cameras to determine the origin of the image. The characteristics are compared to the exact replicas and derivates of other statistical images to detect forgery.

These and other related image forensic techniques [98] [99] [101] focus exclusively on forensics within the image. Very little of the research carried out to date has specifically investigated the

ways and means in which documents are counterfeited and the methods and procedures that can be used to detect such activities.

Digital forensic investigators utilise different tools and methods to accomplish a similar task. Moreover, with the changing world of digital technology, these techniques are variable to change. It is therefore relevant to discuss the techniques related to this research.

2.7 Digital forensic techniques

According to the Oxford English Dictionary, a technique is defined as “a way of doing something, especially one for which you have to learn special skills” [27]. In the field of digital forensics, a digital forensic technique consists of a procedure that is followed and the tools that are necessary to conduct the investigation. A “tool” in such a context can either be a hardware tool or software tool. The digital forensic techniques utilised in this research include windows registry, log files and hex editors. Other techniques that can be used in an investigation include string searching, removing known files, recovering deleted files, recovering hidden files, alternate data streams, and live forensics. What follows in the next section is a discussion of the windows registry. The windows registry is discussed because it was utilised as a digital forensic evidence source in this research.

2.7.1 Windows registry

The Microsoft Windows registry stores quite a lot of digital forensic data, and such data may be crucially important for an investigation. According to the Microsoft Knowledge Base article “256986” [28], the registry is a central hierarchical database used in Microsoft Windows operating systems to store information that is necessary to configure the system for one or more users, applications and hardware devices. The registry files are usually called *User.dat* (in earlier windows versions) and *System.dat*. The registry is accessed from the operating system by means of *regedit* from the run command. The registry editor GUI then executes. The GUI consists of the registry hive keys, which is a group of keys, subkeys and values in the registry that are supported by a set of supporting files that contain data backups [28].The registry configuration files are located in the *%SystemRoot%System32\Config* folder, except for the hive CURRENT_USER which is in the *%SystemRoot%\Profiles\Username* folder.

The Microsoft Windows registry contains information such as installed programs, most recent documents, and most recent websites. The operating system continually references this information during operation. The registry files are found in the windows configuration directory.

Another method for analysing the windows registry involves using Forensic Tool Kit (FTK), which will be discussed in detail in section 2.7.1.The table to follow, Table 2.1shows the registry hives.

Registry Hive	Supporting files
HKEY_LOCAL_MACHINE\SAM	Sam, Sam.log, Sam.sav
HKEY_LOCAL_MACHINE\Security	Security, Security.log, Security.sav
HKEY_LOCAL_MACHINE\Software	Software, Software.log, Software.sav
HKEY_LOCAL_MACHINE\System	System, System.alt, System.log, System.sav
HKEY_CURRENT_CONFIG	System, System.alt, System.log, System.sav, Ntuser.dat, Ntuser.dat.log
HKEY_USERS\DEFAULT	Default, Default.log, Default.sav

Table 2.1: Registry hives

Examiners can view the registry files by using file-registry view. FTK can automatically locate every registry file in the evidence [29]. The following table, table 2.2 describes the hive keys. To explain how table 2.2 can be used. If an investigator for example needs to identify the user profiles on a system. An examination is conducted on the key titled HKEY_USERS (second row) which contains loaded user profiles. To identify the name of the actual file containing this information (as the registry viewer is a displayer only) table 2.1 is used to identify the system file that contains required information in this case, the system file that contains user profiles (Default.sav and Default.log). These system files can be individually examined to view their contents by using another application, for example, a hex editor. This is necessary if the investigator wishes to utilise another application which is not the system registry viewer.

Key	Abbreviation	Description of contents
HKEY_CURRENT_USER	HKCU	Configuration information for the currently logged on user such as users folder, screen colors and control panel settings
HKEY_USERS	HKU	The actively loaded user profiles on the computer
HKEY_LOCAL_MACHINE	HKLM	Configuration information particular to the computer that applies to any user
HKEY_CLASSES_ROOT	HKCR	Information that sets a particular program to open when a certain file has been opened
HKEY_CURRENT_CONFIG	-	Hardware profiles used by the computer during system start-up.

Table 2.2: Description of hive keys

In most cases a different application is used when an examination is conducted on a different operating system, for example, on a Linux system which does not have registry editor. However the resulting views from different applications are the same. The abbreviation in column two of table 2.2 (HKU) can be used as a short cut to find associated keys using the function “find key” in the registry viewer. Log files are utilised as an evidence source in this research and the particular log file examined are discussed in chapter seven.

2.7.2 Log files

A log file is a file to which a computer system inscribes a record of its activities [30]. There are two types of log files: system log files, and application log files. System log files are those log files that are generated by the operating system. Application log files are those log files that are generated by a particular application. System log files include Windows event logs (such as system, application, and security logs) and application logs (such IIS, FTP logs) [30]. The application log files are for applications in the operating system. The log files examined in this

research are application log files generated by the particular graphic design application. The researcher also utilised hex editor tools during the research.

An investigator could use, for example, the security logs to identify when a system was logged in and which user profile was accessed at particular times. The security logs therefore enable recognition of whom and when a system was accessed. These security logs can help identify a potential perpetrator of the system during an investigation.

2.7.3 Hex editors

A hex editor is a software tool that allows a user to edit the binary data that constitutes a data file [25]. A Hex editor is also referred to as a binary file editor or byte editor. The name hex is derived from the fact that it often uses hexadecimal numbers as its standard for displaying digital data. A user can edit the raw data contents of a file by using the hex editor. Other than hex editors, forensic applications can be used to edit raw data, such as, FTK [43] as they also contain a built in hex viewer. Hex editors are generally preferred during investigation because of their portability. For example, they are relatively available in small sized files than other forensic tools therefore easy to install, execute and in some cases without requiring a software license.

Examples of hex editors include Winhex [31], HxD [32], HHD [33], Hexedit[34]. During investigations, hex editors are preferable for analysing the binary data of files. The researcher used Winhex for analysing digital data during this research. Figure 2.1contains a representation of a hex editor.

So that the reader will be able to interpret the hex editor figures demonstrated in part three of this dissertation, the content of figure 2.1 is explained in what follows.

- The address pointer column is the hexadecimal representation of the address contents.
- The hexadecimal column is the hexadecimals for the metadata.
- Hexadecimal metadata is the actual metadata represented by a highlight.
- ASCII is the exact ASCII representation of the metadata.

- At the bottom titled offset, one finds the exact address that refers to the highlighted metadata.

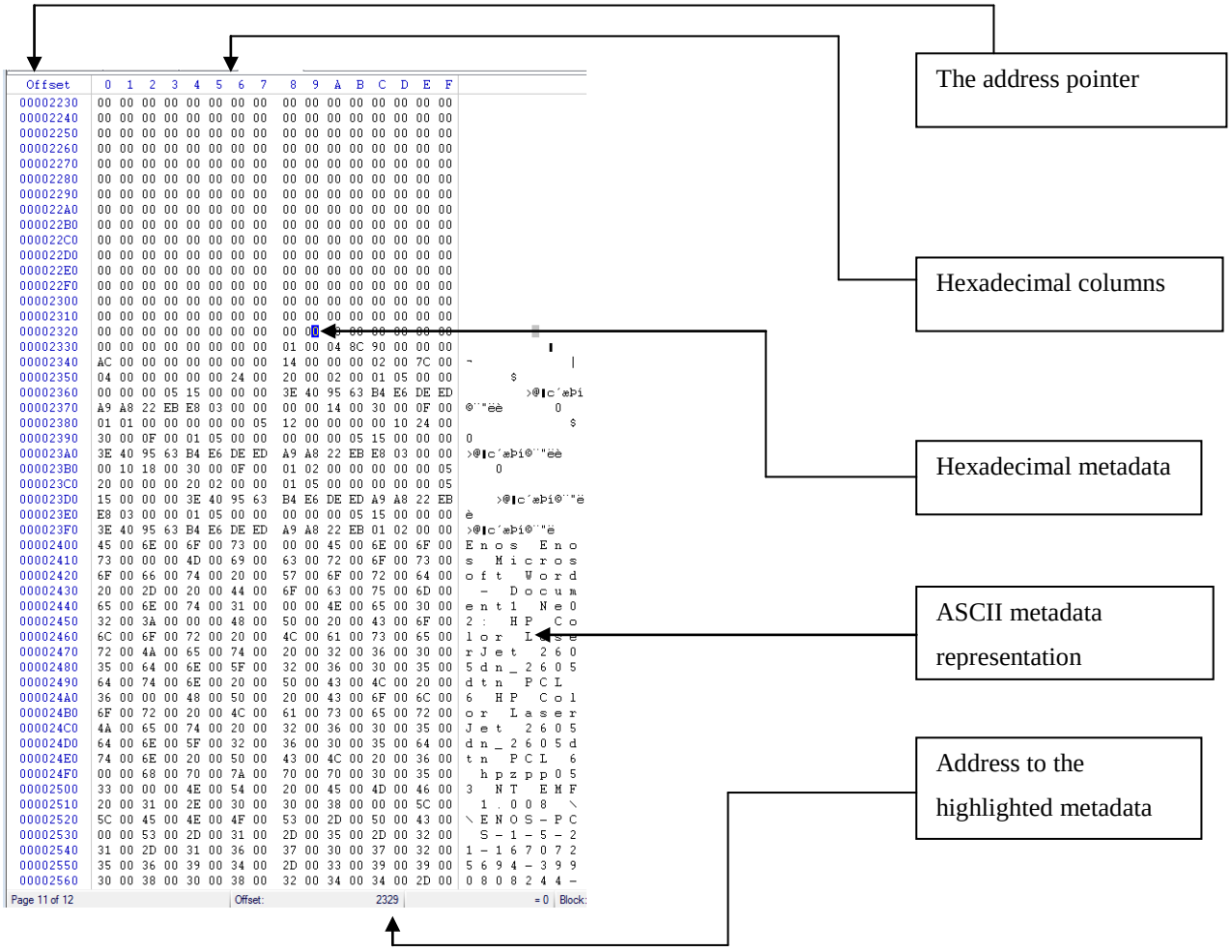

Figure 2.1: Example of a hex editor

The hex editor can therefore be used to examine any file. An investigator can use the various hex editor applications by inserting or dragging and dropping a file into the application depending on the commands available to the particular application. The investigator is then able to view the contents of the file by browsing through the displayed pages of data.

The digital forensic techniques to follow, 2.7.4 to 2.7.9 can also be utilised in this kind of research, related to the counterfeiting of essential documents.

2.7.4 String searching

During a digital forensic examination, an investigator needs to examine a large number of files and system folders. Because this is an extremely time-consuming process, it becomes necessary for an investigator to narrow the search grid. String searching is a technique that is employed by examiners when they know what they are looking for and they want to access only the necessary files as quickly as possible. To perform a string search, one extracts all the strings from the binary data of the evidence files [29]. The string command is exercised by specifying either the names of the required files or their file extensions. Thus, for example, the investigator might want to investigate only Microsoft and Acrobat files. He or she can then specify the names of the files or their extensions **.doc* and **.pdf* for the search grids. The investigator then uses the search command to search the strings for all keywords in the extracted text file, and then outputs them to a console. The result is a narrowed or filtered search that includes only those files that need to be examined.

2.7.5 Removing known files

A thorough investigation on a computer system consists of evidence scattered around the computer system inside or outside executable system files. In any investigation, it becomes necessary to exclude from the investigation all commonly known files such as operating system files so that time and money can be saved and so that the investigator can devote more attention to those files that are more likely to contain the required evidence. Jones and Bejtlich [29] suggest the following two methods for removing files that are irrelevant to the investigation.

The first method requires investigators to populate a directory with the data that they want to add to a hash set. The data might be a few trusted programs or a complete trusted operating system. Once they have obtained the trusted data files, they can compare them to a list of hashes that have been generated from evidence. Another command will produce all the files that do not match the trusted data set.

The second method involves the use of a commercial tool such as Encase. Encase has the capacity to build hash sets natively or to import them from an existing set. When a hash set is given to Encase, it populates the columns on the hash set and hash category. After comparing the

known hashes to the hashes of the file or the unknown files, the known files are filtered out. Encase can then build a hash set from the trusted files that are given to it.

2.7.6 Recovering deleted files

The recovery of deleted evidence files has been an important issue for the many years in the world of digital forensics. In those cases were one wishes to cover his tracks, the attacker can delete his files in such a way that no evidence will be found that will link the crime to him. The researcher describes two methods for the recovery of deleted files.

The first method involves taking a forensic duplicate and making it behave like a real hard drive device [29]. When the forensic duplicate is associated with a device, the examiner will be able to run tools just as though the original suspect drive were connected to the forensic workstation [29]. After executing tools on the device, the examiner will be able to see what the partition table looks like, and then mount the partition to recover the deleted files by using a forensic tool such as the Coroners Tool Kit by Dan Farmer and Wietse Venema, or by using TASK, which was later named The Sleuth Kit by Brian Carrier [29].

The second method involves the use of commercial forensic software. Deleted files are recovered automatically when an investigator uses Encase from Guidance software and the Forensic Tool Kit from Access Data, both of which recover files without any user intervention. The first step is to load evidence into such a tool. The investigator then creates a case to add the evidence, and then adds a raw image to the case. The graphical user interface displays the deleted files that have been recovered.

2.7.7 Recovering hidden files

The recovery of hidden files may be effected by the simple expedient of turning on the hidden property of folders to show the files that are hidden in a Windows environment. Hidden files may also involve the uncovering of a covert channel such as steganography or identifying slack spaces and performing cryptanalysis on encrypted files.

“Slack space” is a type of unused space not being utilized by the actual data from the cluster size and is used to hide data because it does not show up in a directory or file system [11]. “Steganography” is a method of message concealment that can be applied to pictures, audio or

videos [11]. Programs such as Steg detect [35] and Steg break [36] can be used to detect steganography. Cryptanalysis can also be used to recover hidden files. Cryptanalysis is a process that is used to translate encrypted data back into its readable form when the user does not have the encryption key [11]. Cryptanalysis does not always produce successful outcomes. Its success depends upon the strength of the encryption algorithm that the original encryptor used [37]. Frequency analysis that is used for performing cryptanalysis is another option that can be used for decrypting files. Such a higher-level frequency analysis involves examining the encrypted text for repeated character strings and then using the distance between the repeated string to calculate the key length [38]. Password cracking can also be used for finding passwords in protected files. The number of password cracking tools that are available on the market is legion for example, Ophcrack[39].

2.7.8 Alternate data streams

Alternate Data Streams (ADS) are features that were introduced into the Windows NTFS file system. Because its features are attached to a file, they are not generally visible to a user. [29] A file consists of different data streams, one which holds the security information (the access rights), and another stream holds the real data that one expects to find in a file. There may also be another stream that contains link information, and other alternate data streams that hold data in the same way that it is held by a standard file system. ADS are hidden from the standard file system. That means that a user can hide a lot of data in the alternate data streams, and nobody will even notice that from the standard file explorer.

NTFS's file stream can be detected with several command lines tool such as List Alternate Data Streams (LADS), Forensic Toolkit, Crucial ADS and Encase [40]. Prodiscover DFT can also examine alternate data streams and permit examinations without any altering of the original data [41].

2.7.9 Live forensics

Sometimes the imaging of an entire hard drive becomes complicated. It will take several days, for example, to make of a forensic copy of a hard drive with a size of tens of terabytes. In order to make a forensic copy of a drive, the drive needs to be taken offline. While this is happening, a company may suffer huge financial losses until the drive comes back online [11]. "Live

forensics" is a technique for leaving a system in an up-and-running state. That means that one needs to take a "snapshot" of the entire system, its memory and its drives while it is still running. Live forensics may, however, affect other files such as file timestamps, registry entries, swap keys, and memory that are frequently checked by a hash [29]. Given that a system is suspected of criminal activities, and the necessary authority granted, one can install monitoring programs such as Windows Forensic Toolchest (WFT) [42] prior to the occurrence of any incident. Another option involves running forensic software from a USB. From there it will be able to take "snapshots", and this will forestall the necessity of having to change a large part of the status of a system.

Several of the techniques discussed can be combined into software tools. What follows is a description of the available software tools that are able to perform the tasks described.

2.8 Software-assisted tools

"Software-assisted tools" is used in this dissertation to refer to application tools that are used in digital forensic investigations. In the subsections that follow, the researcher describes some of the digital forensic applications that are currently in use. In these descriptions, the researcher makes mention of the better-known software packages that are used in some digital forensic investigations. These include Forensic Tool Kit, Encase, Paraben, Email Detective, Data lifter and Drive Spy. In such cases, the reader will notice that some of these tools can be used to conduct some of the digital forensic techniques that have been discussed. Numerous software packages are available for digital forensic investigations including but not limited to, the ones discussed in this subsection. The focus is on the tools that are occasionally referenced [11] [12] [13] [15] [16] [18].

2.8.1 Forensic Tool Kit (FTK)

FTK is developed by AccessData. AccessData was established in 1987, and offers a variety of products, services, and training to digital forensic experts, government agencies, corporations, and legal firms [43]. They offer a broad spectrum of stand-alone and enterprise-class solutions for legal review, E-discovery, and compliance auditing. Their available software packages include Ultimate Toolkit, Forensic Toolkit, Distributed Network Attack (DNA), Password

Recovery Toolkit (PRTK), mobile phone examiner, Silent runner mobile, AD triage and FTK Pro.

AD triage is used for acquisition and pre-viewing of on scene live (turned on computer systems) and dead systems (shut down computer systems). Silent runner is like a network surveillance camera. It monitors, captures and analyses network traffic Password recovery is used to unlock files from known applications, for example *pdf*, *zip* and *rar*. MPE+ is used for visualisation and analysis of mobile data from mobile devices like tablets and cell phones.FTK Pro is a computer platform for visualisation and analysis of data from computer systems. Views data in multiple displays integrated with graphs and charts for analysis. One of the latest gadgets that are sold by AccessData is a FIELD TABLET. This tablet is designed for field investigations. Its handy size means that investigators do not need to carry evidence back to their laboratories for examination. It is used for both acquisition and analysis. One can simply connect this device to another source and then view mobile device data for analysis. This gadget supports iOS, Android, Windows Mobile and Blackberry, and includes a built-in reporting tool. Such mobile analyses can also be performed without having to acquire the data by means of device rooting.

2.8.2 Encase

Encase has been developed by Guidance Software. Guidance software is recognized globally as one a world leader in digital forensic investigations. The service that they provides include litigation support, incident response and training for cooperate law and government professionals [44]. Their products include the Encase suite of packages, which comprises Encase Forensics, Encase Portable, Encase Enterprise, Encase E-discovery, and Encase Cyber Security.

Encase Pro is used to search, collect, preserve and analyse data from servers and workstations. Encase E-discovery is used for managing electronic evidence from external internet domains including cloud reviewing and document reviews. Encase Cyber security is an end point incident response and data auditing software. It is capable of analysis of potentially infected systems. Encase forensics is used to acquire data from a wide variety of devices, analyse drives and report on findings. Encase portable is used to acquire data onto a USB device in a court proven manner, without having to physically carry the computer system for lab analysis, hence called “portable”.

2.8.3 Paraben

Paraben offers enterprise forensics, hand-held forensics, hard-drive forensics, and network forensics software [45]. They also offer training programs and certification for digital forensic investigators. In addition to these products and services, Paraben also offers open source software such as p2p shuttle, p2p explorer and link 2. Paraben Corporation has also developed a device seizure v4.6 that is similar to the field mobile examiner by Accessdata. The device seizure can acquire and analyse data from four thousand mobile devices. It can also acquire SMS capability, a phonebook, call history, calendars, email, etc.

Paraben's P2 eXplorer Pro allows one to mount almost any forensic image or hard drive and explore it as though it were a drive on a computer machine while preserving the forensic nature of the evidence. Also accesses the deleted, slack, and unallocated space on a drive. Email examiner is used for bookmarking and reporting within email servers including searching for attachments. Online charter is able to examine ICQ, Yahoo, MSN, Trillian, Skype, Hello, or Miranda chart messaging services. Paraben's Forensic Replicator is used to acquire a wide range of electronic media from a floppy drive to a hard drive. The replicator images can be compressed and segmented and read into other popular forensic analysis programs including FTK. Paraben's Decryption Collection is used for decrypting files BitLocker and TrueCrypt are used for recovering passwords for RAR on Google Chrome, QuickBooks and Quicken 2011 databases, MYOB 2008-2010 files and Peachtree 2008-2010 accounts. Deployable P2 Commander (DP2C) runs from USB and is used to boot into the USB to avoid contamination of data and to recover deleted files. The DS Box has an interlocking cable system for various cell phones. It is able to acquire data from a cell phones' SIM card, media card and internal memory. Link2 is used for visual representation of cell phone data. Deployable Device Seizure (DDS) is designed for use in the field with mobile phone. DDS can integrate into a PC tablet and supports devices with IOS 5 and blackberry devices.

2.8.4 Email Detective

Hot Pepper Technology is one of the leaders in custom applications. They offer specialized digital forensic products for email extraction, such as Email Detective (EMD) and echat locker [46]. The echat locker is used primarily for recording and sealing instant messages and chart

conversations along with any background information from an online conversation. The EMD allows investigators to extract email contents, including all American online database stores and the e-mail content of clients.

2.8.5 Datalifter

Since 1996, Stepanet Datalifter has been offering digital forensic investigating tools to the public and to professionals. Products developed by the corporation include Datalifter Forensic Solutions and Datalifter Training Programs [47]. They have also developed a file extractor tool which has data carving running on multiple threads. The tool supports source files that are bigger than 2GB. The extractor tool can be used for file recovery. It creates md5 hashes of each file and allows the optional deletion of duplicate files. It also has a digital companion that has been developed for use with a digital camera. It is also capable of recovering digital images from major camera brands and models of memory cards and USB thumb drives.

2.8.6 Drive Spy

Digital Intelligence Software offers software and hardware tools for identifying slack space, recovering deleted files, imaging, partition un-hiding, and write blocking, among other features. One particular product is Drive Spy [48].The drive spy is used to inspect slack space and deleted file metadata. The Ultra Bay is another hardware tool developed by Digital Intelligence. The Ultra Bay includes a touch screen display and a graphical user interface (GUI) for acquisition and process monitoring. It provides flexibility for write-blocked acquisitions from up to eight different types of storage media: IDE, SATA, SAS, USB 3.0/2.0/1.1 and FireWire 400/800. The Ultra Bay tool also contains a built-in network functionality that enables network image acquisition. The Ultra Bay tool is not the same as the write blocker described in section 2.3.1 but has the write blocking functionality as an added feature.

Various ways can be conducted by an offender in order to hide ones actions. In the next subsections, the researcher describes the various levels that can be exercised by a criminal within the context of this research.

2.9 Anti-forensic techniques

Anti-forensics can be defined as a way of avoiding detection of one's actions on a computer system [103]. In an investigation, one question would be, "what if the crime offender tries to hide all the evidence in different ways?" In this sub-section the researcher looks at different ways through which a criminal would hide his actions. The levels of anti-forensic counterfeiting will be illustrated by use of a pyramid as shown in Figure 2.2. The pyramid is drawn so as to demonstrate the different techniques to be discussed. The extent of anti-forensics is discussed within the context of this research ranging from simple file renaming to drive wiping.

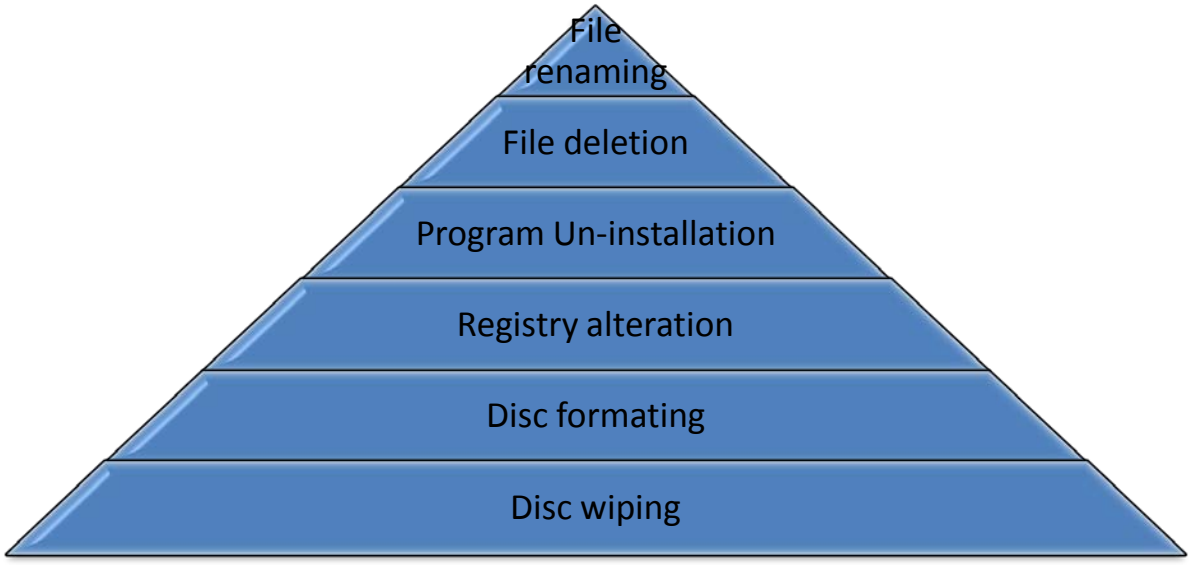

Figure 2.2: Anti-forensics in a pyramid illustration

Figure 2.2 represents the scope of anti-forensics to be discussed in this subsection. The pyramid displayed illustrates six layers of possible ways of hiding one's actions. The bottom layer which is "drive wiping' represents the lowest level. The lowest layer demonstrates the most effective anti-forensic technique, meaning that if one has drive wiped, then the entire block of layers become discarded as there is nothing to un-install or rename. The top layer represents the highest level, the least effective anti-forensic technique, which is "file renaming". After an offender has renamed a file there is still an option to continue hiding one's actions (down the pyramid) to the next level, for example, deleting the renamed file. Each of the layers is discussed in the next sub section.

2.9.1 File renaming

A criminal might decide to hide criminal actions by renaming a file. File renaming can involve renaming a document created from a graphic design application, for example, changing the name of a counterfeit document from *"Daniels passport edit"* to *"Daniels Graduation photo"*, so as to avoid suspicion. . File renaming on created documents is a relatively weak technique as it does not change the metadata within the renamed file. This means that the file can still be examined to identify traces of digital information indicating the files "original name" or "default name" when it was created. This "original name" can be determined from the files metadata using evidence identifiers. Discussions on the types of evidence identifiers are contained in chapter 8.

2.9.2 File deletion

When a file is deleted from a file system, most operating systems do not overwrite the blocks on the hard drive that the file is written onto [92]. Instead, they simply remove the file's reference from the containing directory. A criminal can decide to delete a document created from a graphic design application, log files or all application files within a computer system. File deletion is a well known anti-forensic technique. The technique can be addressed by use of file recovery tools by supplying specific file extensions, file signatures, file names or executing a full recovery on a system. The recovery is executed using traditional "undelete" tools, such as Paraben's P2 eXplorer Pro [42]. Files can also be recovered from multiple devices including internal memory of cell phones or external memory [93].

2.9.3 Program un-installation

Un-installation results in an installed software package being removed from a system. Un-installation results in the deletion of program files and registry keys associated with an installed application. Deletion of program files can be addressed through the use of recovery tools as discussed in the pre-ceding subsection. Registry keys can be recovered using special designated tools. The technique concerned with deleted registry keys is discussed in the following subsection.

2.9.4 Registry alteration

An offender can alter registry keys through access to the kernel memory [89].The offender can either change or delete registry entries. A technique proposed by Gavitt, [89] uses cached data to recognise changes made to the registry as most registry changes leave copies of digital information within the virtual memory of a system. Gavitt describes how to determine such changes within the registry. Morgan [104] uses an algorithm that recovers deleted registry keys within the registry structure. Both of these techniques enable an investigator to recover altered or deleted registry keys.

2.9.5 Drive formatting

Drive formatting is a lower level (more effective) anti-forensic technique that results in the entire drive being deleted of its contents. However given that a drive has been formatted, several tools have since been designed that are capable of recovering formatted hard drives. Examples of such tools are FTK [43] and Encase [44]. Formatted data can also be recovered from Solid state disks (SSD) while considering the manufacturer of the SSD [90].Another technique include using map reduce, which is a programming model for data retention. [91].

2.9.6 Drive wiping

Drive wiping is similar to drive formatting. Wiping is more severe to formatting as it is considered to involve over-writing on top of the formatted data. It is the lowest level considered for discussion in this section. Unlike formatting which can be exercised using the operating system, drive wiping often involves special drive wiping tools that erase the entire drive into a state that renders it difficult to recover the lost information, for example CCleaner [115]. Wright et al [106] discusses the controversy within which drive wiping is considered recoverable and in some cases irrecoverable. Some researchers however still believe wiped drives can still be recovered [105].

The pre-ceding section has been studied to a level of understanding necessary for the research conducted. However, the fight between forensics and anti-forensics is a growing battlefield.

2.10 Conclusion

This chapter set out the information that the researcher obtained from a literature review on the subject of digital forensics. In it explanations for terms such as *digital evidence* and *digital artifacts* are given. Discussions have been given for digital forensic techniques that an investigator and others frequently use in various stages of their investigations. Image forensics and anti-forensic techniques are also discussed. The following chapter, chapter three, is the second background section that prepares the reader for part two of this dissertation.

The main focus of the study is on methods of obtaining digital evidence from graphic design applications. The following chapter will therefore focus on graphic design applications and the contexts in which they are used.

CHAPTER THREE GRAPHIC DESIGN APPLICATIONS

3.1 Introduction

Graphic design applications are the tools that are nearly always used for the creation of digital art. The capability and platform availability of these tools are always being expanded, and their latest tools support three-dimensional (3D) [49] digital editing and Smartphone graphic designing. Graphic design applications are used to carry out the case studies that are involved in the creation of counterfeit documents in this research. It is therefore important to establish a record that describes their role and function in the modern world. The information contained in this chapter was assembled from a literature survey that the researcher undertook on the topic of graphic design applications. The chapter begins with an introduction to examples of those graphic design applications that are currently available in the industry. This is then followed by a description of the applications that the researcher used for a case study in this research. The researcher then compares the different versions of these applications before concluding the chapter.

3.2 Introducing graphic design applications

There are many graphic design applications that are currently available in the industry. While the list of these applications is extensive, the field remains an area that is constantly expanding and growing as more and more applications are added to the list. What follows in Table 3.1is a list of random examples of the graphic design applications that are currently in use as there are no sources recordings or rankings for all graphic design applications.

All the mentioned graphic design applications in Table 3.1 are able to perform similar tasks although they perform them in different ways. It is therefore important for digital forensic investigators to remain aware that different applications leave different trails and residues of information behind, and that all of these need to be interpreted accordingly.

Adobe Systems Incorporated, however, remains the largest software manufacturer of graphic design software [65]. Because it is the largest software producer with the largest customer database, it can be assumed that most software users are using Adobe graphic design applications.

	Application Name
1	Adobe [50]
2	CorelDraw Graphic [51]
3	AutoCAD [52]
4	Free DWG [53]
5	Primo [54]
6	Sweet Home 3d [55]
7	Google SketchUP [56]
8	Ulead [57]
9	Edraw Max [58]
10	DAZ Studio [59]
11	ChemDraw Ultra [60]
12	Photo to Cartoon [61]
13	Easy Flyer [62]
14	PCB artist [63]
15	Sothink [64]

Table 3.1: Selected Examples of Graphic design applications

The researcher therefore undertook his case study by using Adobe graphic design applications because they are more frequently used than any other graphic design application. When one focuses on a method for finding digital evidence from applications that are more widely used than any other, such a method will be relevant to most investigations that deal with graphic design applications.

3.3 Adobe Systems Incorporated

Adobe Systems Incorporated has created software technologies that are used in online transactions, business applications and in social websites [50]. A TopTen review indicated that the Adobe Creative Suite 5 (CS5) is currently the best in the category of graphic design applications [66]. The creative suite is a package that consists of sixteen different applications for various purposes such as audio editing, video editing, file creation, video players, web players, photo editing, web designing, and application linking tools. The applications that are capable of graphic designing include Acrobat, Photoshop, In-Design, and Illustrator. Adobe Acrobat is a portable document file editing, protecting, filing, and delivering tool. The application can only create and save documents in *pdf* format. Since quite a number of articles have already been published about *pdf* files [67] [68], the researcher has not examined this aspect of Adobe Acrobat in this study. There are various ways in which one can create a counterfeit document using Adobe graphic design applications.

To create a counterfeit document using Adobe applications, one can use the following procedure. To acquire a document to counterfeit one can utilise the scan option through the *import WIA support* command which links any attached scanners. This immediately activates the scanning on the scanner application. The scanned application is loaded into the application and is ready for editing. An image might already be available on the system, for example, the image might have been received through email. In this case the criminal would load the image into the application using open command which opens supported file types and loads in the application ready for editing. To edit the document, an offender can change images of human faces, fingerprints or any other image by using the place command that browses and inserts supported files. The user then shifts the inserted image to the required position through dragging and resizing. Other application tools can be used to alter the image. These include type characters, cropping, spot healing, erasers, blurs and others to enhance the image to look similar to an original copy. To create a copy of the document the user makes use of the *save as* or *export* command both of which create a file using different file formats. To print the document the user utilises the menu file *print*. The described procedures apply to all the Adobe applications except the scan command which is available to Photoshop only. This means counterfeiters using In-

Design or Illustrator would have to use the place command to obtained images pre-loaded on the system.

What follows is a description of the three graphic design applications that have already been mentioned and that are used in this research.

3.3.1 Adobe Photoshop

Adobe Photoshop is a professional industry-standard application for digital image editing and creation. Adobe Photoshop provides an interactive platform for changing a picture's format, and joining and splitting pictures, and in changing the colour and appearance of pictures – among numerous other features [69].

The application is also an excellent tool for manipulating photographs and for applying textures and blurs and many other special effects. The application also possesses many photograph enhancement tools. This also means that one can alter or enhance the photographs in identity documents with the greatest ease. Photoshop is also used for print web and motion graphics. Photoshop is a pixel-based application. This means that its output is more pixel formatted than vector formatted. It also possesses a colour correction facility, and can bring vector paths as smart objects. It is not, however, the best software for logo creation because its output is more pixel-based, and pixel data cannot be enlarged without distortion. But vector format logos retain their quality even they are scaled to different sizes. For example, a logo created with 345 X 250 pixels can still look similar when resized to another size of 600 x 434 pixels. This is a function that cannot be so well performed in Photoshop. The type tool in Photoshop is generally of a poor quality for printing. All in all, however, Photoshop produces the best results in design with photo associated documents such as the human faces that need to be inserted into identity documents. Another advantage of Photoshop is that when one creates a document, one can scan the original document, edit the copy, save it, and then print it in the same application.

Photoshop it is a better tool for creating and inserting a logo into an ID document in the form of a photograph than it is at creating it individually because such individual creations might be less than perfect. In a real world example, Photoshop is more preferred if a counterfeiter is interested in changing a human face or any photographic image. This is because of its versatile

enhancing tools like brush and blur. These can clear the edges created so that it becomes difficult to visually notice the counterfeiters' traits in the image.

3.3.2 Adobe In-Design

Adobe In-Design is a professional layout and design application that delivers complex and high-quality graphics and typography. Adobe In-Design is frequently used in the design and editing of magazines, for printing page layouts, and for facilitating digital distribution by using creative built-in typography tools [70]. Adobe In-Design is almost identical to Photoshop in terms of its design capabilities. In-Design is, however, mostly used for editorial design, book design, magazine design, and annual reports. The application can also be used by an offender when one is counterfeiting an ID book that consists of a number of pages. While In-Design is also able to draw lines, it does have not have filter tools. It is therefore difficult to use In-Design to create logos. It is capable, however, of creating three-column layouts more efficiently than the other two applications. This application can also define page numbers: this means that one can create files with multiple pages. In-Design can also import vector art-type files such as *.*ai* file types. If the same file types are loaded or imported into Photoshop, they are converted to paths and thus lose print quality. With this, its shows that it is better to create a logo from another application then import it into In-Design.

If an offender is using the application for counterfeiting by making use of In-Design, one can edit a pre-loaded document, save it, and then print it. But because Adobe In-Design has no function for scanning a document, users have to scan documents by making use of pre-installed printer software packages.

In a real world example, In-design is preferred in creating passports and identity books. This is because the application can create page by page documents. A counterfeiter can utilise the "document set up" option to increase the number of pages in the document. For example a counterfeiter can create a passport with the identification page. Add pages for visas as required. The application enables the ability to bind the pages into one document distributable and printable as required by the counterfeiter.

3.3.3 Adobe Illustrator

Adobe Illustrator is an application that is used for vector artwork in planning projects [71]. It has drawing tools and brushes that can be used in graphic designs that require rigid shapes and various types of line drawings, to mention but a few of its applications. Its scalability functions also work very well. Adobe Illustrator is very similar to drawing programs such as Corel Draw [51] and has similar tools. It is an application that is better at designing logos and the print output for Illustrator is superior to those in the other two applications. Other features that are offered by this application include monogram features or an insignia designing function. It is also capable of designing web and motion graphics. Vector paintings are a feature that is greatly valued by users who are interested in designing logos. But Illustrator cannot create animations. Unlike In-Design, Illustrator does not have a master utility for page creation. This means that a forger cannot use it to create an ID book type or a passport because Illustrator does not have a paging utility.

When criminals are using Illustrator for counterfeiting purposes, they can edit a pre-loaded document, save it, and then print it. But since Adobe Illustrator does not include a function for scanning documents, users have to resort to a pre-installed printer software package if they want to scan anything. All in all, Illustrator is the preferred application for creating a document that includes a professional-looking logo. Such are the three graphic design applications with which the researcher has been involved in this study.

In a real world example, the counterfeiter can utilise the application when intending to design an emblem rather than inserting it. The application has functions and tools like “warp” and “lasso” which can be used to draw a national emblem, a court of arms or a company logo. This enables a counterfeiter the ability to create a logo similar to one on an original document.

In the following section, the researcher describes the three different versions of the Adobe suite investigated during the course of this research.

3.3.4 Adobe versions CS5, CS4 and CS3

Because most graphic design application users prefer to use the latest versions of any software, the researcher used the latest Adobe version at the time of study as the base in his experiments. It

should be noted, however, that when the researcher experimented with the two earlier versions of the Adobe suite, namely the CS4 and CS3 versions, similar results were obtained. While the latest CS5 version of the Adobe suite has been given superior design tools than the two previous versions, all of these versions possess the same design functions.

3.4 Conclusion

In this chapter, the researcher discussed the three graphic design applications, namely Adobe Photoshop, Adobe In-Design and Adobe Illustrator, used during experiments for this study. Also discussed are the different properties, capabilities, and features each of these three applications possesses. The three applications used were Adobe Photoshop, Adobe In-design and Adobe Illustrator. Although any one of these applications can be used for editing and creating documents, the versions that the researcher used were the CS5, CS4 and CS3 versions of the Adobe suite. The researcher deliberately chose the Adobe suite because, as has already been mentioned, Adobe is used by more people than any other suite of this kind in the world today. The researcher therefore felt that it would be helpful to investigate the latest version of the suite, at the time of the study. However, several experiments were performed with the earlier versions of this Adobe suite (namely versions CS3and CS4) for comparative purposes, and they produced similar results, some of which will be discussed later in this dissertation.

Because the problem statement of this study states that graphic design applications can be used to create convincing counterfeit documents, it is necessary to review what literature on the subject has to say about counterfeit documents. In the following chapter, the researcher will discuss counterfeit documents and their implications in modern society.

CHAPTER FOUR COUNTERFEIT DOCUMENTS

4.1 Introduction

Ordinary citizens encounter numerous counterfeit documents in their day-to-day lives without even being aware of their existence. Since this research is devoted to the various issues that surround the creation of counterfeit documents, the researcher has devoted this chapter to the discussion of what counterfeit documents are and how they function in modern society. The researcher first offers definitions of the terms counterfeit, counterfeiter and counterfeiting and then illustrates the ways in which a counterfeit document can be made by available graphic design software that utilises a whole range of creative techniques. After describing these techniques, the researcher briefly explains how counterfeiters focus on specific elements of design and verisimilitude. The purpose of using the counterfeit documents is also described in brief before concluding the chapter.

4.2 Defining “counterfeit”

A counterfeit document is any document that has been illegally created for fraudulent purposes [27].A counterfeit document is a fake that has been forged with *mala fide* (bad faith or intention). In this dissertation, the researcher refers to the person who intentionally creates forged documents for criminal purposes as the “counterfeiter”, and refers to the process and actions involved in creating these fraudulent documents as “counterfeiting”.

The discussion that follows describes why it is important to have a clear understanding of the various ways in which counterfeit documents are created.

4.3 Techniques for creating counterfeit documents

There are various standard ways of creating counterfeit documents, and the skills with which they are created depend entirely on the expertise of the counterfeiter concerned. In the subsections that follow the researcher describes some well-known techniques for counterfeiting documents. These include photocopying, laminating, manual picture replacement, and digital editing.

4.3.1 Photocopying

Because state-of-the-art colour photocopying machines are freely available to the public, it is possible for anyone with some degree of skill, to create a false identity. In order to do this, the counterfeiter needs first to acquire an original identity document and photocopy all the required sides or pages in preparation for the creation of a false ID document. Sometimes the counterfeiter is not even required to make any changes to the counterfeited document because the person who uses the false identification papers may prefer to use an accurate copy of the original. And the owner of a counterfeited document may wish to use the counterfeited documents to represent oneself for various purposes that will be described later.

Photocopying uses light sensitive material to reproduce written or graphic material [88]. The light sensitive material enables production of copies with minimal by noticeable distortions to original materials.

4.3.2 Laminating

Some identity documents are laminated for protective purposes. Because superior laminating machines are freely available on the market, it is possible for a counterfeiter to laminate a counterfeited document so that they look exactly like the original document. A laminating machine is an electronic machine that is designed to bond a sheet of plastic protective covering of a selected thickness to any document for the purposes of protecting it. Since these laminating machines are highly portable, they can be used in any environment for effecting both the small-scale and large-scale production of laminated documents.

The plastic like enhancement properties laminated by laminating machines improves the appearance, durability and resistance against mechanical and chemical damage [88]. This improves the barrier properties of the basic substance against light, grease and vapour.

4.3.3 Manual picture replacing

It is a relatively simple matter for a counterfeiter to physically remove a photograph of the head and shoulders of a person from its original document by making skilful use of some sharp object such as a razor blade. Once the original picture has been removed, a similar size but different human face can be placed in the space provided for a photograph. This kind of manipulation is

usually only used in the forgery of identity documents of a fairly poor quality. The manual replacement of photographs can fairly easily be detected by an alert person with sharp eyes because the replacement process often generates visible defects. Some counterfeiters would regard this technique as old-fashioned and no longer acceptable for the production of a satisfactory product.

4.3.4 Digital editing

Digital editing on a personal computer requires the installation of applications that are capable of editing digital art. These tools have been exhaustively discussed in the previous chapter. The sophisticated tools and techniques that such applications provide make it relatively easy for counterfeiters to produce highly sophisticated and plausible counterfeited documents. Such tools are capable of making plausible alterations to most of the features of an original document, including colours, fonts, textures, human faces, finger prints and bar-codes.

A counterfeiter who uses such applications is able to make a high-quality alteration to the several features of an identity document during the process of production. In the sub- sections that follow, the researcher discusses some of the elements that can be altered digitally by a counterfeiter in order to falsify a document.

4.3.4.1 Barcodes

Barcodes can be acquired through inserting an image received from an external source, for example, email or scanned. The bar code can also be acquired through the use of bar code generator applications [52]. These applications can generate a random bar code or the user can specify the bar code number and the bar code is automatically generated. The barcode can then be saved in a favourable format. This is then inserted into the application for document alteration or editing.

4.3.4.2 Fingerprints

Fingerprints can be acquired from external sources as an attachment or through the use of biometric systems. Portable biometric systems can be purchased that can be attached to a computer and the user scans the fingerprint and it is saved on the system using a native format.

The fingerprint image can then be inserted into the document within a graphic design application to counterfeit an existing fingerprint

4.3.4.3 Signatures

Electronic signature captures are available from vendors. The signature capture machines have a pad used to capture a signature. The signature is transferred to the computer and becomes ready for counterfeiting the signature already present on the counterfeit document.

4.3.4.4 Human faces

Images for human faces can be acquired through external sources or can be acquired using cameras or other digital devices like a cell phone, tablet or webcam among many. The image becomes available on the system and can be transferred to the document for counterfeiting using Bluetooth, USB or any available connectivity to a computer system.

In the section that follows, the researcher discusses some of the changes that can be made during the process of counterfeiting an identity document.

4.4 Counterfeiters target specific elements

Depending on the nature of the identity document that needs to be counterfeited, it is possible to replicate different object or elements of the original document in the counterfeit copy. In this subsection the researcher provides examples of how certain elements or objects can be altered in a counterfeited document. Figure 4.1 depicts a South African driver's license with a fingerprint that can easily be replaced. Figure 4.2 represents a South African ID book with a bar code and human face that can also be easily replaced.

Apart from the elements mentioned, textual data such as identity numbers and names are also usually changed. The nature of the items that need to be replaced varies in accordance with the type of the identity document and the purposes of the counterfeiter. This happens because counterfeit documents are used for the whole variety of purposes. In the following section, some examples of the uses of counterfeit documents are discussed.

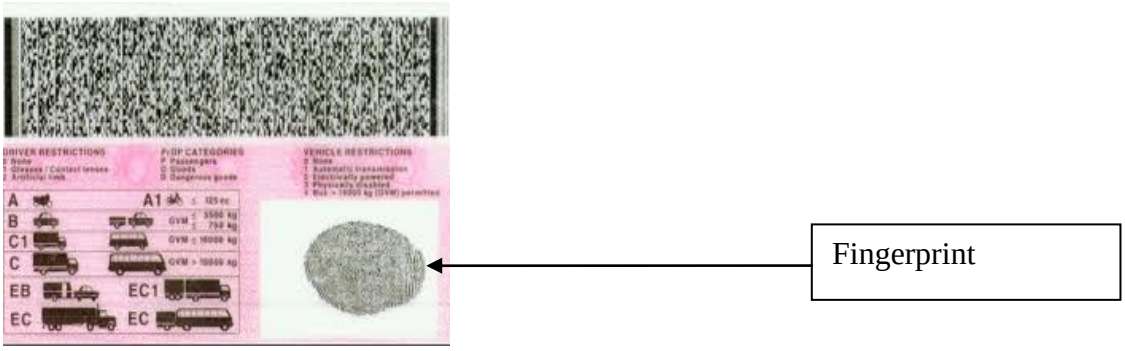

Figure 4.1 South African drivers license (back view)

Figure 4.2 South African ID identity page

4.5 Uses of counterfeit documents

A counterfeit document can be used for a variety of criminal and fraudulent purposes. Counterfeit documents are often difficult to identify with the naked eye. Apart from this, many corporations and government agencies lack the necessary machines for detecting whether a document is counterfeited or not. In what follows in the next sub-sections, the researcher provides some examples of how counterfeited documents are used. Such uses include under-age alcohol purchases, fraudulent banking transactions, protecting the identity of terrorists and criminals from detection, and illegal driving.

4.5.1 Under-age alcohol purchases

It is illegal in most countries for individuals under the age of eighteen to purchase alcohol or cigarettes. It is for this reason that some under-age individuals make use of counterfeit documents such as false identity documents to purchase alcohol.

4.5.2 Fraudulent banking transactions

It is a standard procedure in banks for banking officials to ask customers to produce proper identification before certain banking transactions can take place [2]. Illegal banking transactions are often facilitated by criminals who produce counterfeited identity documents.

4.5.3 Terrorism

Terrorists make use of counterfeited documents, and passports in particular, for example, to escape detection at customs barriers [1]. In order to conceal their identities and their intentions, terrorists invariably make use of counterfeited passports for travelling between different countries.

4.5.4 Unlicensed driving

Under-age individuals or in-experienced drivers often make use of counterfeited driving licences.

These documents vary according to the country or state in which the under-age individual operates and the severity of penalties for conviction for this crime if they are caught.

4.6 Conclusion

This chapter examined various aspects of the production of counterfeit documents. It offered definitions of the terms counterfeit and counterfeiter. It also examined the techniques, uses and replaceable elements of counterfeiting. Counterfeiting has been discussed so as to explain the degree to which this research is focused on. However, the research conducted in this study focuses only on the production of counterfeit documents and not on the uses, categories and consequences of counterfeiting. This chapter concludes the background part of the dissertation, which is part two.

Part two of this dissertation contained three background literature chapters about digital forensics, graphic design applications, and counterfeiting respectively. The literature gave the researcher the capacity to conduct a digital forensic investigation on graphic design applications utilised for counterfeiting purposes. The following chapter describes the experiments that the researcher conducted during this research.

Part III: Model

CHAPTER FIVE SCENARIOS FOR THE EXPERIMENTS CONDUCTED

5.1 Introduction

The previous three chapters explained what the researcher had gathered from a review of the literature on graphic design applications, counterfeit documents, and digital forensics. The information gathered from this literature review put the researcher in a position to explain how a digital forensic investigation illuminates the methods that counterfeiter utilise when they make use of graphic design applications for counterfeiting. In this chapter, the researcher explains the research method used for this study. The research method involves the creation of a series of experiments for demonstrating how counterfeit documents are created and how they may be detected by an investigator who accumulates the necessary digital forensic evidence. The experiments are explained in the following two sub-sections, which are entitled Experiment counterfeiter and Experiment investigator respectively. The chapter concludes with an explanation of the experiments.

5.2 Experiments

The researcher conducted two experiments for this research. In the first experiment, the researcher simulated the activities of a counterfeiter. This involved simulating the various methods that a counterfeiter would use to create a counterfeit document. In the second experiment, the researcher followed the various steps that an investigator would use to trace the activities of the counterfeiter.

The experiments are explained in the following two sub-sections. The first section explains the creation of the counterfeit documents and the second section explains how an investigator gathers incriminating evidence.

5.2.1 Experiment counterfeiter: 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. The researcher created these

documents by editing the following four ID document insertions: the barcode, fingerprints, signatures, and photographs of human faces. This required a combination of twenty four options (4! (Factorial)= 24) on eleven examined file types, all of which will be discussed in chapter eight. The combination for all file types equalled two hundred and sixty four (24 X 11), and included a few extra repetitions for clarity, yielding almost three hundred documents. These documents were created for the different versions of each application (CS3, CS4 and CS5). The researcher performed the same operations with each version so as to be in a position to notice the differences or the changes to the digital forensic artifacts as more documents were created. The different versions of the application usually resulted in more application capabilities, and the improved digital tools created changes in the digital forensic artifacts. The researcher will explain these changes in later chapters.

What follows is a description of the hardware and software tools that the researcher utilised for the Experiment counterfeiter.

5.2.1.1 Software tools

Since most graphic design application users prefer to use the latest versions of software if they can, the researcher use the most recent version of Adobe, CS5 (at the time of the study) for his base experiments. The researcher also conducted further experiments with two previous versions – CS4 and CS3 – for comparative purposes. The differences between the products of the application's versions will be discussed in later chapters.

5.2.1.2 Hardware tools

The researcher used three different computers, each with a different Adobe version installed on it. The researcher used a HP 4500 Office jet 4 in 1 printer to scan the original documents and to print the counterfeited documents. The researcher created the counterfeit documents by replicating the actions that would have been performed by a counterfeiter to obtain the necessary results. These included the counterfeiting actions of scanning, editing, saving and printing of counterfeit documents. During the reputation of the counterfeiting process, the researcher used plain paper (standard white printer paper, 80g/m^2) as output material for printing, as opposed to the high quality paper that is used in real counterfeiting actions. The output material and type of printer did not, however, have any negative effect on the digital forensic evidence gathered. This

was because the necessary evidence was obtained from the system itself and not from the output hard copies.

5.2.1.3 Platform used

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 [73, 74]. Because of this, the researcher analysed the replicating digital forensic artifacts by using a Windows 7 platform.

After concluding experimental counterfeiter, the researcher performed the experiment investigator.

5.2.2 Experiment investigator: Searching for the evidence

Once the counterfeit documents had been created, the researcher carried out various experiments to search for the traces of evidence that had been left behind from the use of the graphic design applications that were used in Experiment counterfeiter. What follows next is a description of the software tools and methods that were used in experiment investigator.

5.2.2.1 Software tools

The researcher used the operating systems’ registry editor tool, “regedit” to search for associated registry entries, and then used a hex editor, Winhex [31], for analysing the binary data of the log files. The Winhex editor was chosen mainly because one can open files with unlimited sizes much easier. In addition, Winhex is an open-source application. It is, however, also possible to use industry-standard tools such as FTK to view binary files. However, just like the Winhex editor, one needs to make one's own interpretation of the data from the tools.

5.2.2.2 Method

To respond to the problem stated earlier, that “Graphic design applications can be used for creating counterfeit documents such as identity documents (IDs), driver’s licenses, and passports. Moreover, there are no current digital forensic tools available for specifically examining a system and for identifying how it was used in the creation of counterfeit documents.”

Three methods were used to gather digital forensic evidence. The first method was that the researcher gathered digital evidence from graphic design application’s log files. The use of this

method necessitates four actions to create an experimental document from which digital forensic information can be accumulated. These four actions are document scanning, editing, saving, and printing. The researcher will undertake an analysis to find the digital forensic information that indicates that these actions had indeed taken place. By tracking the actions performed, an investigator is able to conduct a systematic investigation that is designed not only to identify the files that were used in the creation of the document, but also to identify the actual documents that were created. For example, if the document was scanned, then the next step would probably be that it was edited. If it was never scanned, then it was probably only edited. At the conclusion of this process, the investigator would be able to trace the possible actions with relative accuracy that a counterfeiter would have used to create a counterfeit document..If none of these four actions were performed, then there is need to investigate the suspected documents, explained in the second method.

The second method allows one to understand how actual counterfeit documents are created by a counterfeiter. This involves analysing the binary contents of the counterfeit documents so that one can accumulate the digital evidence that indicates how they were created. In this case, the file types that were generated from the graphic design applications were examined for digital evidence that indicated whether or not they were indeed counterfeit documents.

The third method involves identifying the time stamps that are linked to the creation of the counterfeit documents and the graphic design applications. In the end the researcher was able to support the evidence gathered from the previous two methods.

5.3 Conclusion

The results that were obtained by using these three methods (evidence from application log files, suspect counterfeit documents, and time stamps) are explained in chapter seven, eight and nine respectively. The three chapters, chapter seven, eight and nine correspond to each of these three methods, and are titled system-generated evidence, user-generated evidence, and the timeline associated evidence, respectively.

It may be observed that any one of the three applications mentioned in the background chapter (Adobe In-Design, Photoshop and Illustrator) can be used for document editing. The researcher did not, however, obtain any digital forensic information from using the Adobe Illustrator’s log

files. It is therefore relevant to note that the system-generated chapter examines the two graphic design applications, Adobe Photoshop and Adobe In-Design, because they record digital forensic information in their log files. The researcher illustrated digital evidence from Adobe Illustrator in the user-generated evidence chapter, which examines all three applications. It is important to note that the terms, investigator and examiner are used interchangeably to refer to the same digital forensic expert throughout this research. Before the researcher describes the detailed results in the evidence chapters, the following chapter offers a high-level overview of the model.

CHAPTER SIX OVERVIEW OF A HIGH LEVEL MODEL FOR FINDING DIGITAL EVIDENCE FROM GRAPHIC DESIGN APPLICATIONS

6.1 Introduction

The previous chapter described the research method that the researcher utilised in this study. The problem being addressed in this research is that graphic design applications can be used to create counterfeit documents and current digital forensic tools cannot examine a system specifically for the creation of counterfeit documents. What follows now is a high-level overview of the dissertation contribution model that shows how digital evidence can be collected from graphic design applications. This overview is intended to offer some insights into the contribution model.

6.2 High-level Overview

The researcher refers to chapter ten as the investigation process, not a model.(Adapted from US Department of Justice [13]). And the model refers to the dissertation contribution chapters. The model involves four chapters. Three chapters describe how evidence is gathered from research, and the fourth chapter describes a counterfeiting investigation process. In order to demonstrate a high-level overview of the model, the researcher uses three diagrams. Figure 6.1 illustrates the counterfeiting investigation process as a two pronged investigation process. The two pronged counterfeiting investigation process is employed for the purpose of illustrating two alternative processes of administering the model. The model can be utilised in two different ways as is indicated by the green and blue coloured process routes. These colour codes are applied in all the diagrams so as to emphasise their essential uniformity. This two-pronged approach therefore indicates two possible investigation routes that can be followed by an investigator, depending on the resources that are available. The green route represents an investigation when the suspect document is actually available. The blue route represents the investigation route that must be followed when a suspect document is not available. Figures 6.2 and 6.3 represent each of these investigation routes of the high-level model in more detail. The colour coding in figures 6.2 and 6.3 are used in the same way as they are in figure 6.1. The detailed high-level model overview

also demonstrates an order that signifies the way in which the remaining model chapters are related to one another.

The phases of each of the process routes are elaborated in detail in chapter ten. For any route that may be taken, whether the green route or the blue route, the three chapters on the gathered evidence are demonstrated in a high-level way in this chapter. The detailed analysis of the gathered evidence is demonstrated in chapters eight, nine and ten.

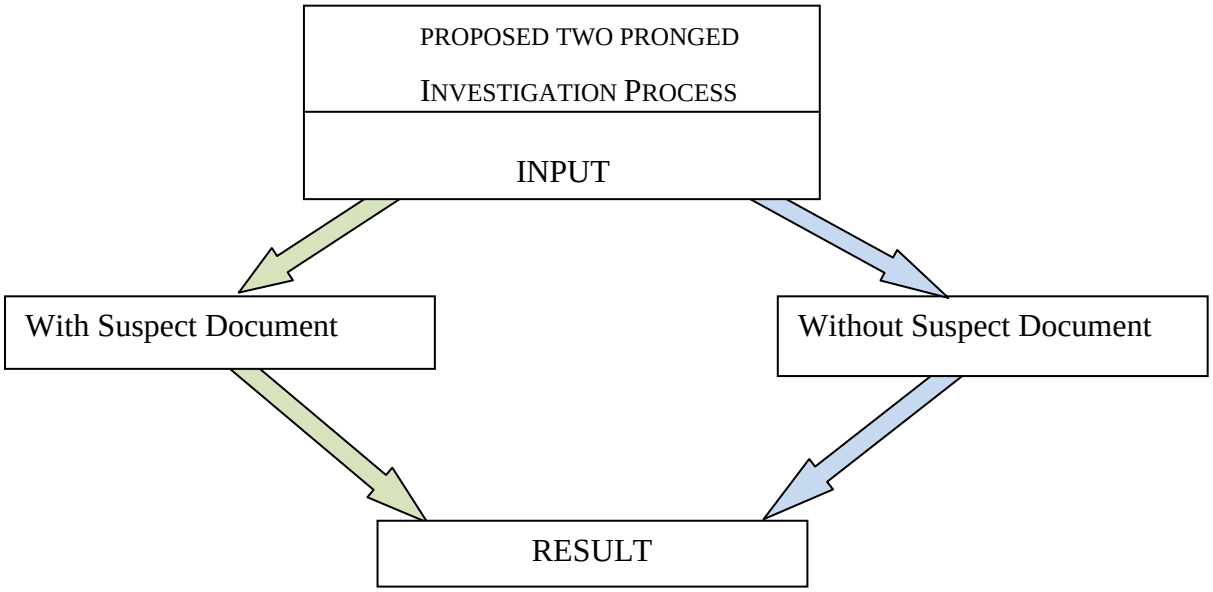

Figure 6.1: High-level overview of the two pronged investigation process

In figure 6.1, “input” represents the entity that is consumed in the process in order to achieve an output, represented as “result” in the diagram. “With suspect document” represent an investigation that is based on an investigation that is initiated with a suspect document. “Without suspect document” represents an investigation without a suspect document being available for examination. For a more detailed representation of each of the processes, on a still higher level, two diagrams highlight the process of the investigation by using the model. The first demonstration illustrates the green route and the later illustrates the blue route. The key on the right bottom of the figures defines the input terms used in figures 6.2 and 6.3. It is important to keep in mind that the explanations in this chapter are high-level descriptions. More detailed descriptions will be given in the relevant chapters.

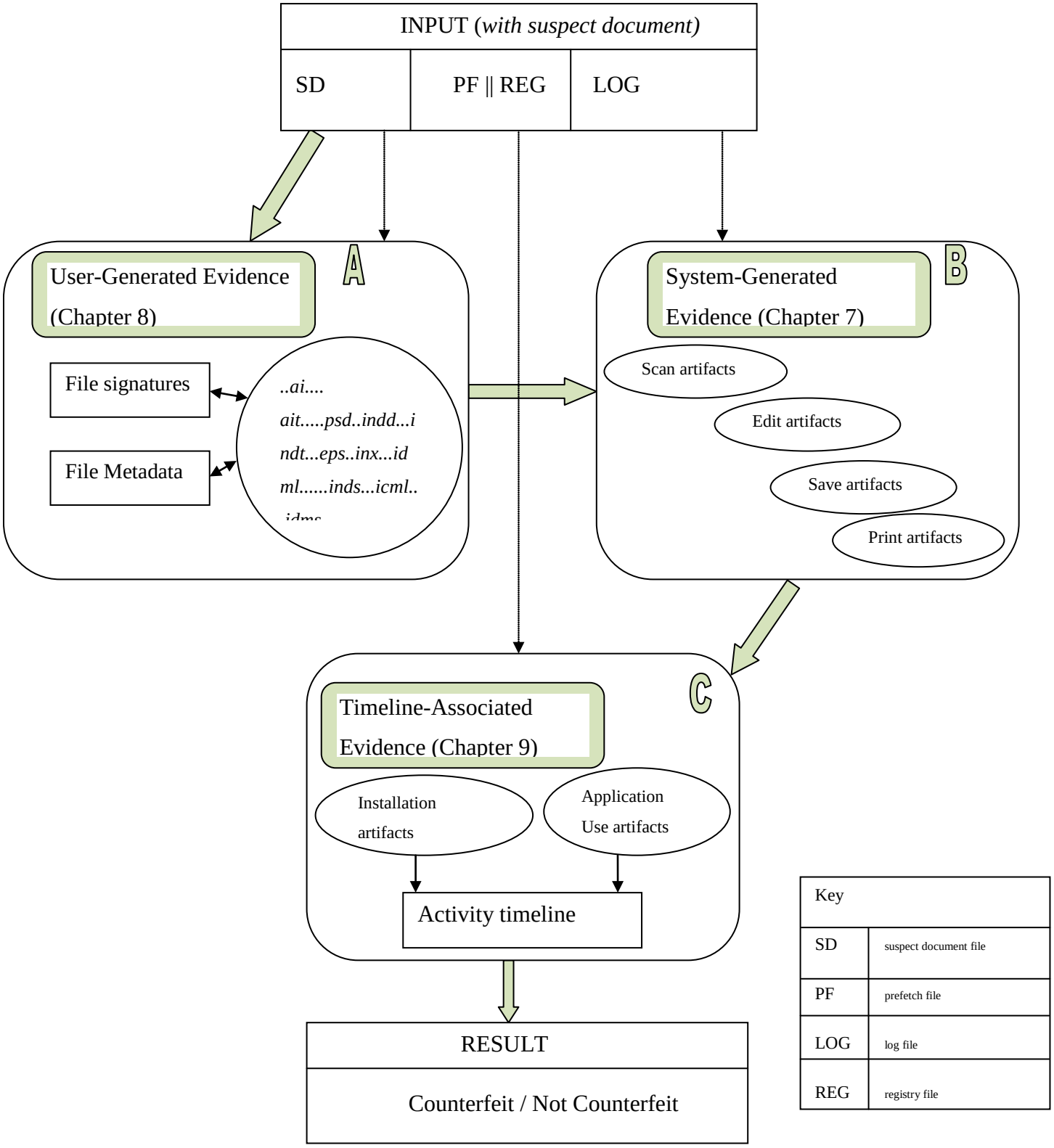

Figure 6.2: High-level model overview for gathering digital evidence from graphic design applications with a suspect document (Green route)

It is relevant to point out that the diagrams are mere guidelines to the dissertation model for the reader to anticipate what is to follow. The diagrams are intended to give an insight to the reader to indicate how the evidence chapters are related. The chapter number is given in brackets in the respective block. In chapter ten, the researcher proposes a digital forensic process of investigating electronic counterfeiting. The proposed investigation process can be used to investigate a system that is suspected of being used for counterfeiting or investigating a document that is suspected to be counterfeited. An overview of this process is given in this chapter as flow guidance to highlight the relationship between contribution chapters.

6.3 Investigation with suspect document

An investigation with suspect document begins with an input of the suspect document itself (SD). SD is examined in the user-generated chapter, which is still to follow, and is represented here by block A. The coloured lines represent the flow of the model and the dotted lines represent the input types, i.e. SD is input source for User-generated evidence, PF and REG are input sources for Timeline-associated evidence, while LOG is input source for system-generated evidence. The coloured arrow represents the flow from input source SD to user-generated evidence block A, and then to system-generated, right through to the result.

Sub-sections 6.3.1 to 6.3.4 are high-level descriptions of the model, which are explained in more detail in the chapters that follow.

6.3.1 User-generated evidence

“User-generated digital forensic evidence” refers to evidence that is intentionally produced by the application user. The researcher discusses eleven generated file types as user-generated evidence in chapter nine. Each one is discussed in terms of the file signatures and the metadata that is gathered from it. File signatures’ true identity can be identified by examining file headers of the files. While for metadata, an examination is conducted on the contents of each file. These are files created by a user through the save command after the creation or editing a document. The researcher offers an evaluation of the evidence identifiers in relation to the particular that is gathered evidence for each file type. The results from the experiments on user-generated evidence are shown in the form of tables. From block A, the process follows to block B, representing the log file examination as input for system generated evidence.

6.3.2 System-generated evidence

"System-generated digital forensic evidence" refers to that evidence produced by the application without user intervention. Log files are examined exclusively for system-generated evidence, and this is indicated by block B. Four possible actions taken during the creation of a document were used as a hypothesis to gather digital forensic evidence from graphic design applications. The researcher explains the sources of the digital evidence and illustrates the four actions, namely scan, edit, save and print. The results are illustrated in the form of tables. Clarifications and comparisons are given with aid of graphs and pie charts. A timeline associated with evidence analysis immediately follows with an examination on prefetch files and registry entries, which are represented by block C.

6.3.3 Timeline-associated evidence

Timeline-associated evidence refers to evidence that reveals the time stamps from the graphic design applications. Mainly two types of time-related evidence are recognised. Time-related evidence is generated from application installation and application usage. These two types of time-related evidence are explained in detail in later chapters. In addition to this, the researcher drew a timeline to represent time-related evidence in a chronological order that signifies the occurrence of counterfeiting events.

6.3.4 Result

From the three evidence chapters, the output is represented as the result of the process of investigation, by following the route where the suspect document is available as indicated in figure 6.2. The output in this case is recognising if a system was employed for counterfeiting or not, and identifying whether the examined document is counterfeit or not.

6.4 Investigation without suspect document

Figure 6.3 represents an investigation without a suspect document. The diagram is similar to the earlier diagram with the suspect document.

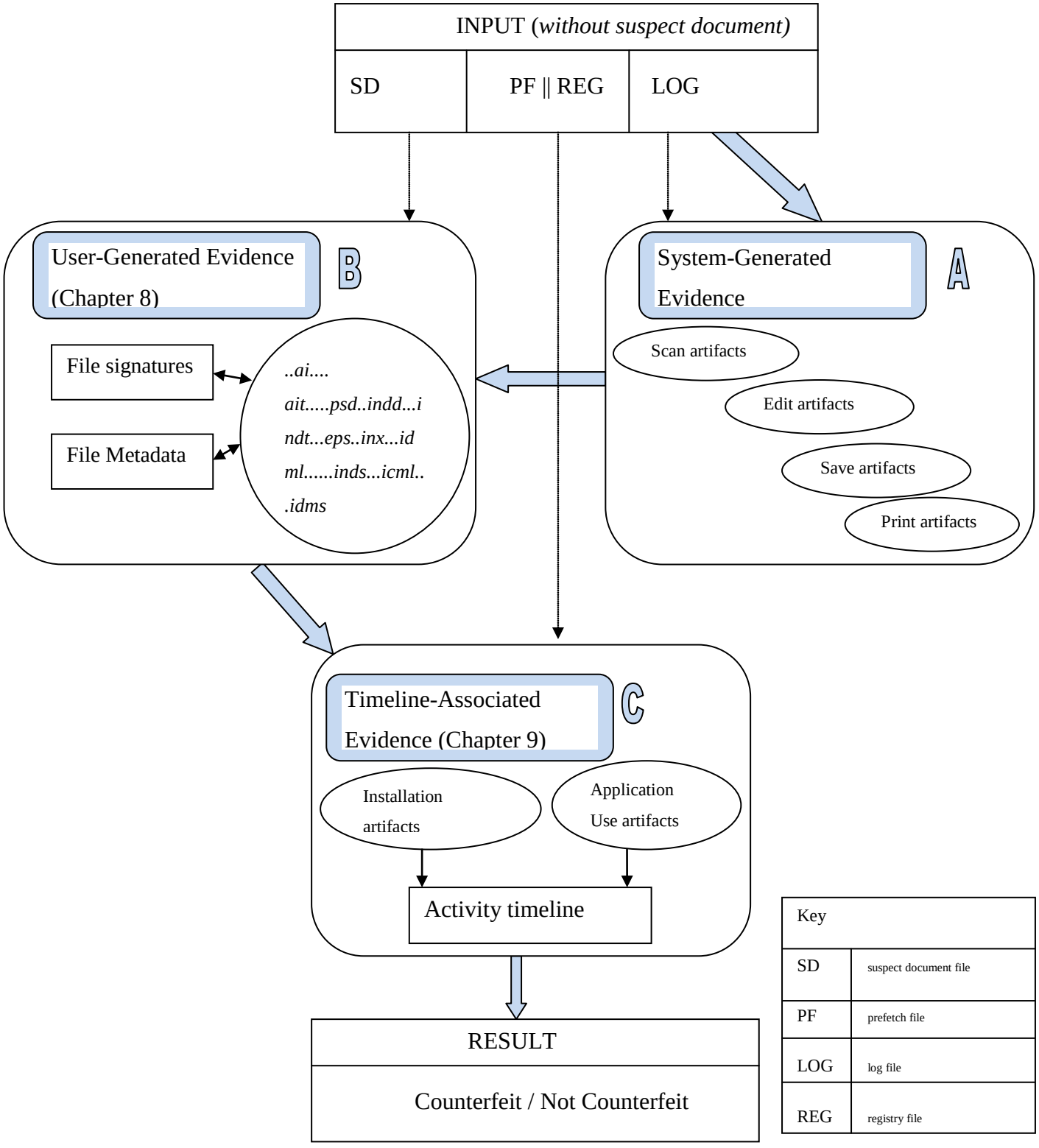

Figure 6.3: High-level overview model for gathering digital evidence from graphic design applications without suspect document (Blue route.)

The difference between the diagrams is mainly in the initial phase, in the first block (Block A) on the blue route which involves examining log files for system-generated evidence, as indicated on in figure 6.3. This is in contrast to the green route, which involves examining suspect files for user-generated evidence as in the initial phase, as indicated on figure 6.2.

It should be noted that the blocks A and B are not inter-dependent though an investigation can be conducted in the described fashion. This means that not obtaining evidence from block A does not render the investigation process to abort. And the same applies if no evidence is obtained for block B. To elaborate, if there is no evidence for system generated evidence, the process can start on user-generated evidence. And if there is no suspect document available for investigation, file extensions described in the user-generated chapter are used to find graphic design application files or files that have been deleted. These files would then be examined as user-generated evidence. The routes described as blue and green route offer alternatives for investigation and the detail is given in chapter ten, investigation process.

The diagrams (Figure 6.2 and 6.3) have been shown mainly to demonstrate the contents of the dissertation contribution model, the flow and relation of the chapters of part three of this dissertation.

Generally the difference is a swap between block A and block B. After block B, the process is similar. The diagrams were deliberately kept separate so that the differences could be emphasised and to avoid a complex combined diagram of figures 6.2 and 6.3 with clustered details. Because of the similarity in block details, the descriptions of each block are similar to those of the previous subsection. The same descriptions will therefore not be repeated here.

6.5 Conclusion

This chapter illustrated an overview of a high-level model for finding digital evidence from graphic design applications. The synopsis gives an insight of what might be anticipated in the model chapters. The block names represent the names of the respective chapters from the model that is still to follow. The model comprises four chapters: a chapter on the counterfeiting investigation process, and three chapters on the accumulated evidence. These four chapters have more or less been previewed in this chapter so that a holistic picture can be obtained of the model

before the details of the model are discussed. Each of these chapters will be discussed in more detail in the chapters that follow.

CHAPTER SEVEN SYSTEM-GENERATED EVIDENCE

7.1 Introduction

The previous chapter presented an overview of the model that showed how digital evidence may be obtained from graphic design applications. This is the first chapter to deal with the detailed model that shows how results may be gathered from the system-generated evidence as indicated in figure 7.1.

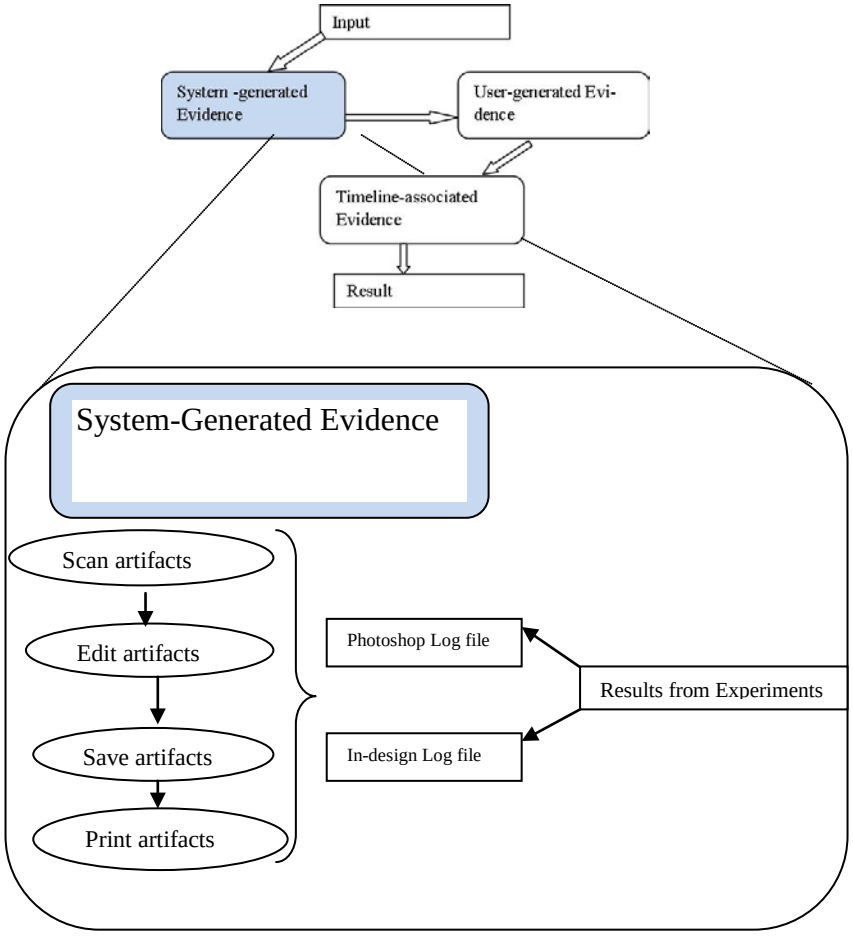

Figure 7.1: A representation of the flow of the model for system-generated evidence

The evidence to be described (chapter 8 to t10) represents the results acquired from the graphic design applications studied in the research. These results are equivalent to the digital forensic evidence that remains behind when certain actions are performed during the creation and production of counterfeit documents. These digital forensic artifacts show that a document has

been scanned, edited, saved, and printed. As has already been mentioned in the chapter about digital forensics, the term “artifacts” is regularly used instead of “evidence” to refer to the remnants that are left behind when these actions are performed. Such artifacts constitute the evidence in any digital forensic investigation.

While “system-generated digital forensic evidence” refers to that evidence that is automatically produced by the application without any specific user intervention, the term “user-generated digital forensic evidence” refers to that evidence that is intentionally produced by the graphic design application user. Evidence of this latter kind is discussed in the following chapter (chapter eight) of this dissertation.

As has already been explained in chapter five, the researcher used the four possible actions that are performed during the creation of a document as a hypothesis to gather digital forensic information from graphic design applications. The researcher undertakes an analysis in order to find the digital forensic information that indicates that these actions were actually taken.

It should be recognised that in some cases not all traces of the actions can be obtained based on the counterfeiters actions. Scenarios of these cases are given. In the first case, if traces of scanning only are obtained, it could mean that the acquired document has not been altered or the counterfeiter did not intend to alter the document. Another scenario, if traces of editing only are obtained, it could possibly mean that the document edited was acquired through other means like email or fax. In another scenario, if traces of saving only are obtained. This could possibly mean that the document saved was edited on another system. In another scenario, if traces of printing only are obtained, the printed document could have been created on another system.

From the different scenarios given, any combination of the traces actions can be obtained. It should also be noticed that evidence of the actions does not necessarily qualify possible counterfeiting. Each recognised trace should be examined to obtain the actual scanned, edited, saved or printed document to be elaborated in the sub-sections of this chapter. If none of the actions are obtained, the task would be to examine suspect files as explained in chapter eight.

The analysis begins by identifying the actions that the counterfeiter performed. By means of this process, any digital forensic investigator is able to begin an investigation in a uniform and

orderly manner. An investigation begun in this way, allows the investigator to acquire the actual files that were used to create the document and the actual files that were created.

What follows in the next section are the results of the experiments described earlier, which are referred to in this text as the accumulated digital forensic artifacts. The results of these experiments are subdivided into four subsections that are labelled respectively as artifacts that are related to document scanning, artifacts that are related to editing, artifacts that are related to saving, and artifacts that are related to printing. Afterwards, a summary is presented to elaborate the concept of the accumulated digital evidence. The chapter ends with a conclusion. Figure 7.1 shows how this chapter is divided.

7.2 Results from the experiments: accumulated digital forensic artifacts

The discussion that follows centres on the digital forensic artifacts that are found in graphic design applications when the source of the potential evidence is mainly system-generated and the results are derived mainly from application log files. Although application system files usually consist of a number of log files, it is nevertheless necessary for forensic purposes to identify the log file that contains the vital information.

The researcher tabulates the experimental results for each of the graphic design applications in terms of their versions (these results in six tables for each action performed). The researcher uses a line graph to illustrate the differences in address offsets for the digital artifacts (this generates two graphs for each action performed). The researcher uses a radar chart to illustrate the distribution of digital forensic artifacts within each of the log files (this generates a total of two charts). Because the radar chart is derived from the tables and graphs, only the two radar charts are presented in this chapter. The tables and graphs are included in Appendix A. The experimental results that are obtained from the digital forensic artifacts that are produced by each of the four actions (scanning, editing, saving, and printing) are further described in each of the subsections to follow.

7.2.1 Artifacts generated by document scanning

Generally, when one attempts to create a fraudulent document, it is necessary first to acquire an original document so that one can use it as the basis for creating a new and fraudulent identity.

When the counterfeiter follows this procedure, the first action that becomes necessary is to scan the original document so as to make it available for digital editing on a computer. There are a number of different models of scanners that are currently available on the market, and all of these utilise different software packages for executing scan commands. For the purposes of this research, the researcher has focused on the digital artifacts that are created from executing the scan commands within the graphic design application. These scan commands need to be executed in the same graphic design application that subsequently executes the editing of the scanned document.

Of the two graphic design applications in question, only Adobe Photoshop is capable of scanning a document when the user selects the “import WIA support” document menu option. “Import WIA support” is an Adobe Photoshop function that operators make use of to connect to available printers or scanners. By making use of this function, an operator is able to connect the scanner directly to the graphic design application. And so the document that is being scanned is loaded into a destination folder when the operator is prompted. The application then creates a folder, saves the scanned image, and opens the scanned image in the application so that it is ready for subsequent editing.

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 gathered by document editing” (section 7.2.2). Even if not all actions are exercised, 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. The more the artifacts recognised, the stronger the evidence.

During the course of this study, the researchers scanned twenty documents for each application version by using the command mentioned (import WIA ...”). When a document has been scanned, the application automatically records the digital artifact (the forensic evidence that scanning has taken place) 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* indicates the Adobe version, which could be version 3, 4, or 5. Adobe Photoshop consists of several log files, namely, *Actions Palette*, *LaunchEndFlag*, *WorkSpace Prefs*, *RepousePresets*, *PluginCache*, *NewDoc Sizes*, *Materials*, and *Favourites*. However, only the *Adobe Photoshop CSX Prefs.psp* log file will contain the digital forensic information that is generated by the actions that are performed during document creation. The other log files generally consist of artifacts for colour, fonts, styles, and whatever other settings may be used during document creation. Moreover, because any system comprises thousands of files, the path to the place where the log file is located, needs to be identified.

After the researcher analysed this log file's binary data, an **entry recorded of the location of the scanned file** in the log file was obtained at certain address offsets. The address was obtained from noticing the path to be similar to the path used during experiment counterfeiter.

Adobe Photoshop thus records a single entry for scanned documents. Any further examination at this location will reveal the actual copies of the original scanned documents. Examination of the scanned documents enables one to determine if the scanned document was later altered or not.

After scanning has taken place, the counterfeiter may inevitably follow this up by editing the acquired document in order to falsify some of its content. The production of editing artifacts is discussed in the following section.

7.2.2 Artifacts related to document editing

Document editing is one of the most important stages in the creation of a counterfeit document because it allows the counterfeiter to insert objects of interest into the scanned document, which may include the image of a human face, a bar code, or a fingerprint. There are number of editing actions that can be performed. These include typing, colouring or drawing. The focus in the study is on the kind of editing that results in the insertion of an image or object because these can later be used by an investigator to determine whether the document that was created was counterfeit or not. While analysing inserted objects, the researcher conducted experiments to establish what could be inferred from a system that would indicate to a digital forensic examiner what had been inserted and the location from which it was inserted. All of the three graphic design applications upon which this study has been based have the capacity to edit a document

by means of attaching or placing an image. The terms “inserting”, “attaching” or “placing” an image are all considered to refer to the same action, although they are referred to differently in various applications. Copy and pasting or drag and drop are shortcuts to inserting an image, therefore recognised similar. In this dissertation, the term “inserting” will be used henceforth to refer to all three of these terms. Inserting an image is one of the main functions found in most graphic design applications.

It is possible to insert various images of different file types into the Adobe Photoshop application by using the place command. Such a placed image can then be moved to any new position. When a counterfeit document is being created, the inserted image will usually be inserted and moved to a position that *covers* or *hides* the original image (thereby enacting a counterfeiting process).

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**. On the basis of this of information, the researcher was able to recognise the **names and location of inserted objects**. After that, the researcher was able to obtain the image of the human face or the fingerprint image by examining the stated location.

In all the examined versions, the digital artifact entry with the name of the inserted object is usually placed after the entry with the name of the path into which the file was inserted (i.e. the path name first, and then the name of inserted object). This means that it is possible to relate the entry with the inserted object to the particular document into which it was being inserted. The next section discusses how digital forensic artifacts indicate insertions from Adobe In-Design.

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. The entry for the path of inserted images is recorded twice within the log file: in the beginning of the log file and towards

the end of the log file. The reason for this double recording is not known. The researcher was unable to reach anyone in Adobe who was able to explain why this should be so. The research also noted that, as more documents were created, and different objects were inserted, the software log file maintained the same entry address. Because the application records only the same entry, the last accessed folder location from which an image was inserted, is recorded in the log file.

In the following section, the researcher examines the action of saving documents and how these create digital forensic artifacts.

7.2.3 Artifacts that are generated by document saving

Once a document has been edited, the user (the criminal) usually needs to save it either for later printing or for further editing. In this section, the researcher discusses the indicative residues that are left in the system when documents are saved. This information is vital because it can point an examiner to the name of the created file (potential evidence) and to where the file was last saved. If the file was deleted or moved, the examiner can also generate search commands on the basis of the names of the files that were saved. The examiner can do this by specifying the name of the file while searching, thereby extending the search filter or search domain during an investigation. All three of the applications that were considered in this research have the capability of saving edited documents in various file types. The various file types will be illustrated in the following chapter. The digital forensic artifacts that are accumulated by document saving are explained for both Adobe Photoshop and Adobe In-Design.

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 names are arranged in order of the last-saved file first. Adobe Photoshop records the location to which the saved files are located at the beginning of the file size. Entries with the actual names of the saved documents are located at about six tenths of the log file. This entry consists of the full file name and it includes the location and the file extension in which the document was saved.

The digital artifacts for saved locations 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\11.0\VisitedDirs*.

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 at the beginning of the log file, while the entry with the names of the saved files appears towards the middle of the log file. The researcher noted that the log file records a maximum of 22 entries of created files. When more files than this are saved, the log file overwrites the older entries with new entries. This procedure applies to all the examined application versions.

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. It is therefore necessary to identify the location of this information by making use of radar chart. Another reason for doing this is that the digital forensic artifacts from the log files do not make use of evidence identifiers such as prefixes and tags. (Evidence identifiers will be discussed in the following chapter.) This tells the investigator to look for this evidence at a pre-determined location, for example, about six tenths (or three fifths) down the file.

Figure 7.2 illustrates the distribution of the digital forensic artifacts within the log file.

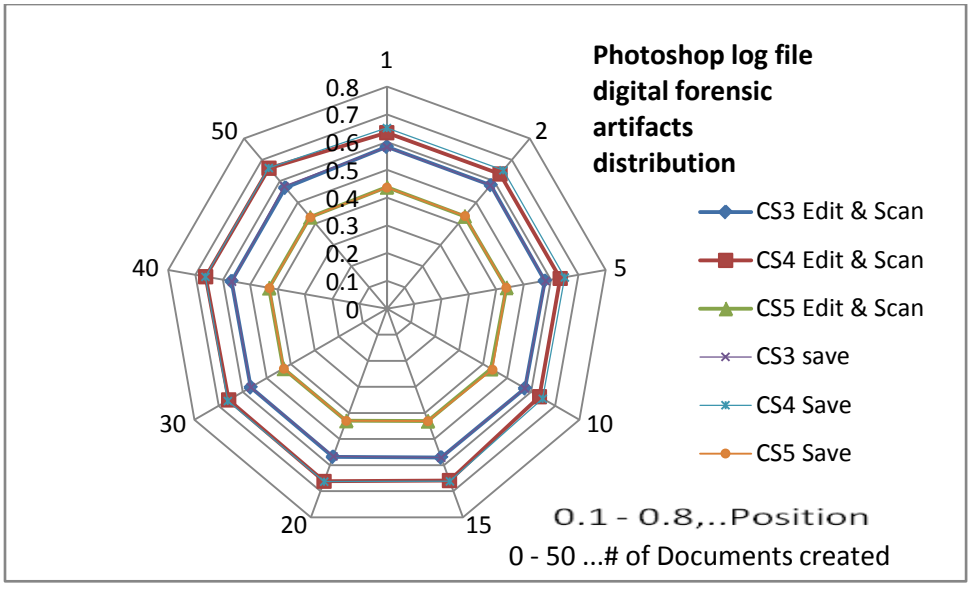

Figure 7.2: A graphic illustration of digital artifacts distribution in a Photoshop log file

The chart in Figure 7.2 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.

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**. This information is located at various locations within the log. The digital artifacts for saved entries are recorded consecutively in the log file with the latest saved document appearing first. For example, if the document "*zzz.indt*" is the first on the list and document "*yyy.indd*" is next, this means that document "*zzz.indt*" was the last to be saved or the last to be created. These digital artifacts are also grouped in terms of the number of documents that were created before the user exited the application. For example, if, at a certain address offset, three digital artifacts for saving entries are present, and at another address offset, six digital artifacts for saving entries are present, this means that the user first created six documents, and then closed the application. When the application was run again later, three documents were created before the application was exited again.

Adobe In-Design log file records the name of the "saving folder" at the beginning of the file, and the full name of the saved file both at the start and towards the end of the log file. Adobe In-Design records entries for saved entries automatically throughout the log file. This means that several data entries are located at different locations. The log file also indicates the name of the folder location for saved documents at the beginning of the file. An entry with the list of documents is located twice in the log file. Tables for address offsets have been included in appendix A.

Generally speaking, one may also verify saved files from any graphic design application by looking at the recent documents available at *C:\Users\<username>\ AppData\Roaming\Microsoft\ Windows\Recent.* This location is useful in those cases where the

user did not delete any recently accessed documents. Figure 7.3 illustrates the distribution of digital forensic artifacts within the log file *Indesign Save data*.

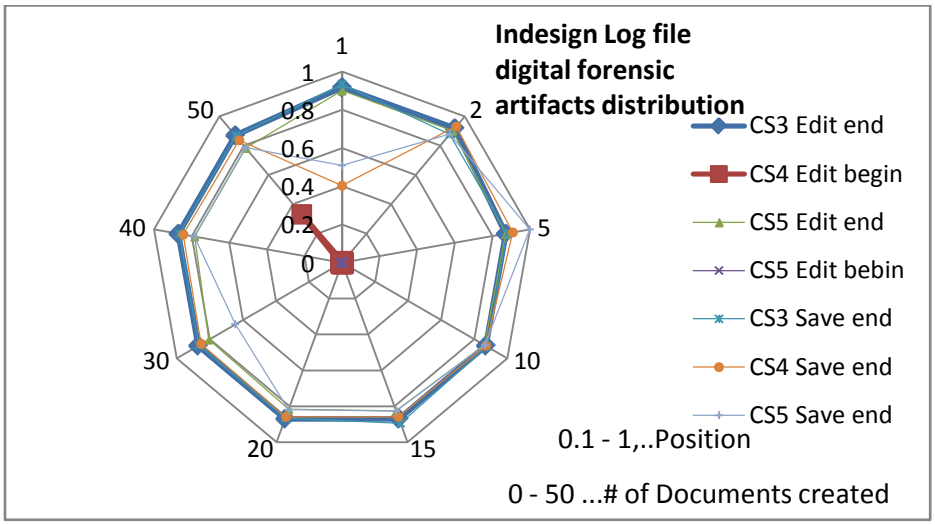

Figure 7.3: An illustration of how digital artifacts are distributed in an Adobe In-Design log file

The radar chart (figure 7.3) 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. After counterfeit documents have been saved, the counterfeiter moves on to the process of printing. This is discussed in the section that follows.

7.2.4 Artifacts indicative of 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 what happens in scanning actions, printing actions can be commanded from all the graphic design applications under consideration via the print menu command option. The artifacts that are illustrated in this section are from any of the graphic design applications that have been previously examined, and this applies equally for all versions.

In order to locate the place from which printer(s) were used to print a document, one makes use of the registry entries set out below. The registry keys, from which a list of printers' connections can be established, are as follows:

(1)*HKLM\soft\Adobe\Photoshop\11.0\Plugin* path.
(2)HKEY_CURRENT_CONFIG\System\CurrentControlSet\Control\Print\Printers

(3)HKEY_USERS\<user id>\Software\Microsoft\Windows NT\CurrentVersion\PrinterPorts
(4)HKEY_USERS\<userid>\Software\Microsoft\Installer\Products
41E0A130314079C4792762937B284FF6\ SourceList

After establishing the names of the printers by using the method described, the actual physical existence of the printers can be verified. This can be a great help to an investigator in those cases where actual printers have been removed. Physical printers are necessary in an investigation because they are needed to match the digital evidence to the actual printer so that the case against the counterfeiter can be supported in court proceedings.

Two spool files are generated for each print job by the operating system that is located at *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. By analysing the binary data of these files, one may locate the name of the spooled document. In addition, those print jobs that were queued to print but have not actually yet been printed can also be found within print spools. Table 7.1 sets out the data entries from the *spl (*spool) file.

Job number	Size of spool file (Mb)	Name of printed job address offset
13	2.19	20
14	1.87	20
15	1.41	20
16	1.82	20
17	3.75	20
18	0.79	20
19	0.29	20
20	0.07	20
23	1.30	20
27	4.53	20

Table 7.1: Address offsets for printed documentsfrom s*pl* log file

The column headings in Table 7.1 are briefly explained here for the sake of clarity. The “job number” heading indicates the system-generated number for a print job. The “size of spool file”

heading represents the size of the spool file at the time of the examination. The “name of the printed job address offset” heading represents the address pointer in hexadecimal format for the digital artifact, pointing to the name of the document in the spool file.

For each print job, the name of the printed file is located at a fixed address offset of 20(hexadecimal), irrespective of the file size of the spool file or the size of the printed file. For this reason, no line graph or radar chart is used in this section to illustrate the distribution of the digital forensic artifacts.

Table 7.2 shows the address offsets from the *shd* spool file, and shows the name of the application that generated the print request, the login name of the user who initiated the print request, and the name of the printed file. It is not necessary to rationalise address offsets from a printer spool because all the entries are recorded at the same address offset (i.e. 0X20), irrespective of the file size. If any other document is printed after the counterfeit document, the evidence is still available. The evidence for printing is mainly used to determine if a document recognised from editing or saving artifacts has been printed. Therefore the investigator uses the names recognised from previous artifacts to verify if the same document was printed amongst the list of all printed documents.

The column headings for Table 7.2 are briefly explained here for the sake of clarity. The “name of printer” is the address offset, with an entry recording the name of the printer that generated the print job, and this entry is repeated, as is shown in the second column “(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. The “application generated” column indicates the name of the application that generated the print job such as, for example, Photoshop. The application name will be the absolute path to the executable graphic design application. The “username and name of file” column shows the address offset of the name of the user who generated the print job and the name of the printed potential counterfeit document.

Name of printer	Name of printer (repeat)	Application generated	Username and name of file
88	3BO	2120	2400
88	3BO	2120	2400
88	3BO	2120	2400
88	3BO	2120	2400
88	3BO	2120	2400
88	3BO	2120	2400

Table 7.2: Address offsets for printed documentsfrom *shd* log file

Towards the end of the **.shd* file, the name of the printed file appears. This indicates the location from which it was printed and the name of the printer that was used to print the document. The timestamp of the **.spl* and **.shd* file indicates the date and time at which the document was created. This information is vital for establishing which counterfeit documents were actually printed.

7.3. Summary

On the basis of the experiments that the researcher conducted for this study, locations to which the scanned documents were saved were obtained. In these locations, the researcher also identified several other documents that were also scanned. With regard to the action of *editing*, the researcher was able to establish the names, file types and file locations of all the inserted objects. The latter are typically represented by fingerprints, digital signatures, bar codes and human face images that are inserted into the original documents. By tracking all the editing and saving actions, the researcher was able to establish the file names, the file types, and the file locations of all the saved documents. The *saving* action enabled the researcher to identify the digital evidence that confirmed that documents had been created, as a list of created documents. The printing action generated registry and spool files that included the names of the printers that had been used for document printing, as well as the names of those documents that had been printed in all the graphic design applications that were investigated. The artifacts gathered point to the digital evidence as defined in chapter two, which are digital images used to counterfeit documents. These can be actual fingerprints, signatures, scan files or actual suspect counterfeit documents.

7.4 Conclusion

This chapter described and discussed the artifacts that could be accumulated from Photoshop and In-Design to indicate the actions of scanning, editing, saving and printing. This chapter also described how digital forensic artifacts are generated in the system, as well as the applications that generated particular artifacts. The researcher described how evidence that indicated scanning was able to identify the digital copy of the original document. Evidence from the editing of the documents enabled an investigator to identify a variety of inserted images including bar codes, signatures and images of human faces. The research also showed how evidence generated by saving actions could be used to identify actual counterfeited documents for the purposes of creating a case against the counterfeiters. In the same way, evidence of printing actions enabled an investigator to establish that documents had actually been printed. It is also important to note that, in the case of printing artifacts, spool log files are generated by the operating system and not the graphic design application. This means that documents can be printed from any application which is not a graphic design application. By pursuing these four actions during an examination, an investigator is able to show that accumulated digital forensic artifacts accurately indicate the chain of events that a particular user used to create a counterfeit document.

Evidence of all actions does not necessarily indicate counterfeiting but document alteration. Therefore an examiner should focus particularly on the elements of document creation like the images of the scans and images of the photographs or fingerprints inserted to determine if a document was counterfeited. Further investigation should also be conducted on non system files explained in chapter 8. It should be recognised that in some cases not all traces of the actions can be obtained. If there is no trace of printing, but the traces of other actions it could indicate that the document was not printed or printing evidence was erased. If also the traces of editing are available then it would mean that the counterfeiter edited a document received through other means like fax and did not save the copy or erased a saved copy. If only the traces of saving are available it could mean the document was edited on a different system or the evidence also could have been erased. In all the cases, if an assumption has aroused, data carving could be conducted using the signatures as described in the next chapter. In the following chapter, the researcher discusses how one evaluates user-generated digital forensic evidence.

CHAPTER EIGHT USER-GENERATED EVIDENCE

8.1 Introduction

In the previous chapter, the researcher examined and discussed system-generated evidence. In this second chapter that deals with the detailed model, the researcher discusses the kind of digital evidence that is intentionally generated by the user, as is indicated in figure 8.1. The chapter deals with the file types (file formats) that are created by the graphic design applications that the researcher selected for the purposes of this study.

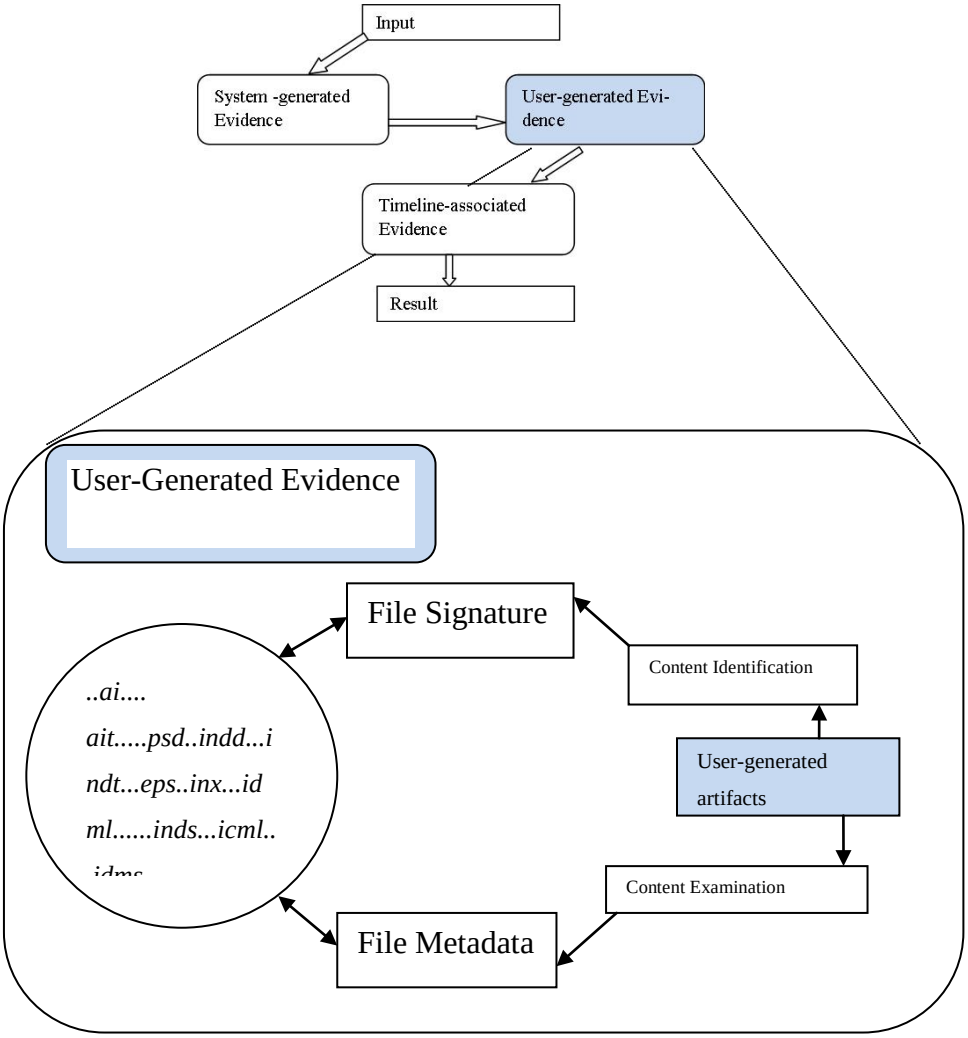

Figure 8.1: A representation of the elements of the model for obtaining user-generated evidence

In order to conduct a comprehensive investigation into any crime that has been committed with the use of a graphic design application, the digital forensic examiner must first acquire a thorough understanding of the nature of the files that are generated from the particular genre of applications that are being used by the criminal. In this case, such applications are graphic design applications. An effective digital forensic investigation is designed to enable an investigator to make full use of the digital forensic artifacts in the affected files [25].

At this point it is worth recalling that *user-generated digital forensic evidence* refers to evidence that has been intentionally and deliberately created by an application user, for example, the perpetrator. Such files are created when the perpetrator saves a document or image, and they are dependent on their type or extension. The identification of user-generated evidence can be divided into the two distinct categories of *content identification* and *content examination*.

When examining counterfeit documents, the digital forensic examiner first examines all changes that have been made inside files in a careful, systematic and thorough way. The investigator will thus, for example, make a careful study of all the fingerprints, barcodes and human faces that are embedded inside the graphic design application file types. It is therefore essential that the investigator examine each file type individually. The three graphic design applications that have been described and utilised in this study are associated with more than thirty nine file types. In this study, however, the researcher has only focused on those file types that are specific to the three graphic design applications that have been used in this research, and has excluded other well-known and well-researched file types such as *jpeg*, *bitmap*, *tag*, *tiff* and *tga*. It should be noted, however, that Gary Kesler and Martin Reddy have created separate lists of these common file signatures online, and that is a valuable database that remains a work in progress, which is constantly updated with new information [75] [76]. There is also a free online metadata extractor tool that can also be utilized for these common file types [77]. This metadata extraction tool was not designed to extract metadata associated with counterfeited objects such as, for example, the names of fingerprints or the bar codes inserted into a counterfeit document. They are more generally used to extract metadata such as, for example, the kind of camera model that was originally used to capture a graphic image.

Before an investigator examines a file intensively, first of all, there is a need to establish the files true identity. What follows in the next section is an extensive description of the two categories that are recommended by the researcher as an integral part of a forensic digital investigation, namely, a “content identification” and a “content examination”. After that is being done, the researcher will explain how digital forensics artifacts are gathered from each of the eleven file types under consideration. The researcher provides an explanation of both the signatures and the metadata and includes examples of the kind of artifacts that are obtained from user-generated evidence. The chapter ends with a brief summary and conclusion. Figure 8.1 sets out the various elements of this chapter.

8.2 Content identification

Content identification is the process that an investigator uses for determining or verifying particular types of specific files. In other words, content identification involves the means that an investigator uses to verify the identity of a file extension. Counterfeiting criminals have the capacity to alter the file extension of a particular file so as to confuse potential investigators and conceal the trail that might lead to their conviction. A person engaged in such criminal activities might, for example, change the file signature of a particular graphic image to a *dll* extension with the hope that the file concerned will be overlooked during a forensic digital investigation. Since system files, including dynamic link library files (*dll),*are usually trusted files, they might well be left unexamined during any forensic file examination processes (how this could be done was discussed in chapter two, in section 2.6.5, under the heading, “Removing known files”).It is therefore essential to confirm the integrity of files by conducting a file signature analysis. In this particular case, it is necessary for a digital forensic examiner to be able to recognize a file type. Proof of the *real* file identity resides within the content of the file, and is usually known as *the file signature*. It is normal practice to identify a file signature by examining its first bytes [78]. The file signature normally resides in specific offsets which are usually located at the beginning of a file. This kind of signature is uniform for all files with an identical file extension.

The researcher noted that although known digital forensic tools such as FTK are able to detect a range of file types, they are not able to produce the same information in the graphic design applications that are discussed in this paper. Digital forensic tools can therefore verify file types

such as tga, bmp, gif, tif and png, among others, but they are not able to verify the graphic design applications that are being examined in this dissertation.

8.3 Content examination

Content examination refers to the retrieval of any embedded metadata that may be present in any given file. Content examination necessitates the identification of the metadata of files, which, in this case, are graphic design application file types. Metadata means *data about data* [25]. In Windows systems this will include the time stamps that have been assigned to the modification, access, and creation of documents, to mention only a few. For the purposes of this study, the researcher focused more particularly on metadata that indicated that counterfeiting had occurred within a file. Metadata is an indispensable component of any forensic digital investigation because it contains evidential information about what might be extracted from a particular file. Such information may include the name of the tool that was used for criminal purposes or, indeed, or the name of the perpetrator who used the application.

For each type of file discussed in the section about accumulated digital artifacts to follow in the next section, the researcher has provided a table that displays the metadata extracted from the file and the address offset respectively. The *offset* refers to the address pointer of the described metadata. In other words, if an investigator is using a hex editor to search for a specific offset, the tool will immediately skip to the particular metadata that will point to the place where the required digital artifact is situated.

However, several experiments reveal that the offset can slightly differ by plus or minus 780 bytes per metadata, which is usually in the same page view depending on the size of the file and size of metadata present in the file. But this is usually in the same page view, depending on the size of the file and size of metadata that is present in the file. In this case, because of the variances in address offset, the digital forensic investigator needs to focus on how the digital artifact is embedded or tagged. If a digital artifact is recognized, for example, as “*For:tG5$u*y6w#”*, an investigator would infer from the prefix “*For”* thatthis prefix refers to the author of the document. In the same way an investigator is able to identify each digital artifact by the tag or prefix that indicates its metadata. A tag may be defined as an identifying keyword or term that is assigned to a piece of information [27]. Because the tag has been

interpreted correctly, the researcher will not draw a data distribution chart, as done in the previous chapter. The same system of tagging was not utilised for log files in the previous chapter because the log files do not contain neither a tags nor a pre-fix, referred to in this dissertation as evidence identifiers. For each file that is examined, the version of the file and the number of images inserted into the file are indicated in the table.

The tabulated values contained in the following section can still be used for graphic design files of any size. It is important to remember, in this context, that Adobe software uses the conventional metadata scheme Extensible Metadata Platform (XMP) to indicate embedding metadata [79]. Because this is an open and extensible scheme, it can be used in various file types, as the researcher will illustrate in the accumulated digital forensic artifacts that follow.

8.4 User-generated digital forensic artifacts

The following section outlines the content identification and content examination digital forensic artifacts that the researcher accumulated for each of the file types examined. The results from his examination are shown in the form of a table. Because of the essential similarity between the tables, only two tables will be shown for each of the first three file types. The first table shows the file signatures of the metadata, and the second table shows the address offsets of the metadata. The file signatures of the remainder of the file types are set out in the summary of section 8.5 (file signatures), and in the metadata tables in the appropriate appendix, appendix B. As was mentioned earlier, the study focuses only those file types that are unique to the three graphic design applications that the researcher examined. What follows are discussions about eleven of the file types that the researcher examined. These are identified with the extensions: *ai, ait, eps, psd, inx, idms, inds, indd, indt, icml* and *idml*.

8.4.1 Illustrator (ai)

Adobe Illustrator artwork 15.0 file type with the extension **.ai* is the default file type for documents that are created with Adobe illustrator. It is regarded as the Illustrator format. The file type adopts the extension from the abbreviation of its creator tool Adobe Illustrator (*ai*).

The metadata in this file type is stored with the extensible metadata platform (XMP) standard, upon which others such as Adobe Bridge [80], In-Design and Photoshop are built. XMP is the

xml-based platform created by Adobe. The files that are saved with the *ai* extension range in size from about one hundred kilobytes to a few mega bytes, depending upon the amount of embedded information in the file. The file type records a certain amount of metadata that includes the author's name and the names of the inserted objects. One can, however, specify a particular version of the format when saving when one is prompted by the dialogue box. This only affects the quality of a document concerned, and not the digital artifacts. Illustrator files are much richer in metadata in comparison to other file types from the other graphic design applications that are examined in this research. The file contains metadata that includes the time stamps, the creation date, the metadata date and modification date. It also includes the author's name and important information such as the name and path of the inserted documents (whether they be fingerprints or bar codes), that were inserted into the document. The name of the author is identical to the log-in name that was used on the computer. This metadata is recorded at certain offsets that will be elucidated in the metadata table.

The metadata tables indicate the metadata that was extracted as well as their respective offsets. For the Adobe Illustrator version CS3, the metadata is embedded with *xap* tags such as, for example, *<xap:Create Date>2012-04-16T14:21:48+02:00</xap:CreateDate>*. Version CS4 and CS5 metadata are, on the other hand, embedded with *xmp* tags. Adobe Illustrator also records the metadata that indicates the names of the inserted images and the respective locations from which they were inserted. This metadata is embedded with *<stRef: file Path>* tags or with the prefix *%%Document file* concatenated with a *%%+* symbol. The earlier consists of a single entry of an inserted image and the later will be all the inserted images in that particular examined file. For three entries, this might conceivably consist of a fingerprint, the image of a human face, and a bar code or digital signature. The name of the author is also recorded in the file with the prefix *%%For:*. The default name for the file is the original identity name that was assigned to the file when it was first created. This name is embedded with-in the tag *<rdf:li xml:lang="x-default">*. The default name in the metadata does not change even when the file is renamed. Other non-essential information includes link forms, object, color types, and other non-ASCII data.

Adobe Illustrator file signatures follow the *pdf* signature convention of starting with pdf-1.5, but it differs in follow-up characters. Table 8.1 shows the file signature of the *ai* file.

File signatures are generally expressed in hexadecimal values. Each signature table in this chapter contains the file signatures that are identified and specific to each of the graphic design applications in this study. The file type in each signature table represents the named form of the particular graphic design file. The file extension is merely a suffix that indicates the encoding of a file's content, and it usually consists of three or four characters that are separated by a dot from the file name itself. The file extension should, however, never be trusted because it can be easily be altered and renamed. An investigator should rather focus on the file signature in order to determine the correct file type.

In the table columns of Table 8.1, the file type represents the name of the file as it is described in the graphic design application. The file extension represents the encoding of the file. The ASCII column represents the entry in its text-readable format. The file signature columns represent the entry in a hexadecimal format. Both these entries appear exactly as they are shown in a hex editor.

File type	File extension	ASCI II	File signature
Illustrator file	ai	%PDF-1.5%âãÏÓ 1 0 obj	255044462D312E350D25E2E3CFD3 0D0A312030206F626A

Table 8.1: Hexadecimal signature for Illustrator *ai* file type

Table 8.2 shows the metadata from Illustrator *ai* files. In that table, the **size of file** column contains the size of the file at the time of the examination. The **Version of App** column contains the version of the application that generated the file. The **Number of images inserted** column contains the number of image objects that were inserted into the document when it was being editing. The **Offsets for time stamps** column shows the address offsets for the time stamps. The **Inserted image** column indicates the address offsets for inserted object images. The **Default name of document** column indicates the address offsets for the original name of the document. The Author name column represents the name of the file creator. The **Creator tool** column shows the name of the graphic design application that the perpetrator used to create the file. This name is necessary for the gathering additional digital forensic information from the graphic design application (the system-generated digital forensic artifacts) that created this file.

Size of file(kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted image	Default name of document	Author name	Creator tool
924	CS3	1	3F1	C6422	E6C30	C6349	3B0
1005	CS3	2	3F7	D9580	2A1	D94A0	3B6
1060	CS3	3	3F7	E6E49	2B3	E6D64	3B6
2024	CS4	1	3EA	16F7B1	290	16F6CC	473
2160	CS4	2	430	19182D	290	19173E	475
2113	CS4	3	430	19E4FB	290	19E410	475

Table 8.2: Address offsets for the metadata that was gathered from Illustrator *ai* files

8.4.2 Illustrator template (ait)

The Adobe Illustrator template with the extension *ait* represents another Illustrator file. This file type is a compacted replica of the previous *ai* file type. The metadata is similar in this case to that of the default *ai* file. This metadata is, however, located at different offset addresses. The file itself still observes the pattern of embedding metadata with *xap* tags for the CS3 version, and with *xmp* tags in later versions. These files are generally smaller by a margin of about one hundred kilobytes because the template is like a model or prototype for all the ai file types. As far as visual quality goes, the template *ait* does not produce as good a visual quality as the *ai* files because they have a lower resolution. Although the file contains information that can be stored in a default *ai* file type, it can only be stored in compact layout. Table 8.3 shows the signature for the Illustrator template *ait.*

File type	File extension	ASCI II	File signature
Illustrator template	ait	%PDF-1.5%âãÏÓ 1 0 obj	255044462D312E350D25E2E3CFD3 0D0A312030206F626A

Table 8.3: The hexadecimal signature for the Illustrator *ait* file template

This file contains metadata that is similar to that contained in *ai* files, and it includes time stamps, the name of the creator tool, and the names of the inserted images. It is important to

recognize that the name of inserted images is recorded with a *stRef:filePath* tag. Towards the end of the file is the default name of the saved file, although it is tagged differently with *Title(xxxxxxxxxx)>>* tag or *Rdf:li* tags for the name of the file. Table 8.4 shows the metadata from the Illustrator template file *ait*.

Size of file (kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted images	Name of document	Author name	Creator tool
293	CS3	1	409	276E0	2B3	27607	3C2
295	CS3	2	409	27BFE	2B3	27B25	3C8
296	CS3	3	409	27D9A	2B3	27CBB	3C8
171	CS4	1	3FC	110C7	267	10FDF	441
172	CS4	2	3FE	1163C	267	11552	443
176	CS4	3	3FE	11EFD	267	11E13	443

Table 8.4: Address offsets for the metadata that is gathered from Illustrator *ait* files

8.4.3 Illustrator (eps)

The Encapsulated Post Script (EPS) file is based on postscript language. This file type contains both vector and bitmap graphics. It also supports RGB-based, CMYK-based, and CIE-based colour models for both vector and bitmap images. When it has been created with an Illustrator application, the file will contain similar metadata to the metadata of the other two Illustrator file types *ai* and *ait*. But *eps* does not record the inserted images in a bundle: it records them in single entries. The time stamps that are contained in the file are recorded only with their creation and modification time but not with their modification time. The creator tool is prefixed with a *%%Creator* prefix and the time stamp for date of creation is prefixed with *%%CreationDate.* The author name is prefixed with *%%For*: The default name of the created document is prefixed with *%%Title.* The metadata for inserted images are still also in *<stRef: file Path>*. This means that inserted images are recorded as single entries. This is different from what happens with the default file type *ai* which records both single entries and a compound entry with all the inserted images at one address location. Table 8.5 shows the hexadecimal signatures for the *eps* file type.

File type	File extension	ASCI II	File signature
Encapsulated post script	eps	ÅÐÓÆ	C5 D0 D3 C6

Table 8.5: Hexadecimal signature for Illustrator *eps* file type

The difference between the Illustrator *eps* and the Photoshop *eps* is that the Photoshop *eps* does not record all the essential data. This file type does not include the inserted images and the author's name. Table 8.6 displays the metadata that can be obtained from the Illustrator *eps* file.

Size of file (kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted images	Name of document	Author name	Creator tool
2751	CS3	1	32F9B	3B027	64	A4	32F5A
2994	CS3	2	32FA2	3B389	64	A5	32F61
3566	CS3	3	32FE3	3B3A9	64	A5	32F62
4960	CS4	2	BA	346EB	271C2B	271C14	7F
5538	CS4	3	BA	34916	2F0C9C	2F0C85	7F

Table 8.6: Address offsets for the metadata gathered from the Illustrator *eps* files

8.4.4 Photoshop (psd)

The Adobe Photoshop Image file with the extension *psd* is the default and only save type for Photoshop. The other files types are export file types. *Psd* is one of the most popular file types in graphic design applications. The file size of *psd* files ranges from a few hundred kilobytes to a few mega bytes. The file is known as the Adobe Photoshop image file. These Photoshop files are different from Illustrator files because they consist mostly of metadata for document resolution, pixel data, color spacing, and information about document pixel dimensions. The files follow the Adobe CS3 pattern of using *xap* tags for embedding the metadata and *xmp* for indicating later versions. The file type has the least amount of essential metadata in comparison to the other file types that the researcher has examined in this study. The file signature of the Adobe Photoshop *psd* file and the file types that follow are summarised in section 8.5.

One might notice that the Photoshop *psd* file maintains the position of its metadata in comparison to other file types, in which some degree of variation occurs. The table showing the *psd* metadata address offsets is contained in Appendix B1, and the tables for the remaining file types are also appended in appendix B.

8.4.5 In-Design (indd)

In-Design *indd* is the default file type from Adobe In-Design graphic design application. The file size ranges from three hundred kilobytes to a few megabytes depending on the amount of editing done within the file. Saving documents in In-Design *indd* results in metadata that contains information about layout and references to source files. In-Design metadata also includes non-essential digital forensic information such as the size of fonts and color swathes. The most important metadata for an investigator consists of the *lasturl* file metadata, which may be a path that the user visited either to acquire an image or to save a file. It also usually records the creator tool type at two different address offsets within the file. The tags used for In-Design are different from those that are used in Photoshop and Illustrator. For inserted images, the In-Design tags are ++@ or just a @ sign or the preferred *<stRef:lastURL>file:* tag.

The time stamps are embedded in *xap* tags for CS3 and *xmp* for later versions. In-Design file type also contains all three time stamps creation, modify and metadata time. The file type does not record the author of the document or the default name of the document, the metadata that it offers is more extensive than that offered by Photoshop. The inserted images are listed in sequential metadata entries. The tag is *file:* or *<stRef:lastURL>file*. The file also contains metadata for the previous file type in between the metadata for inserted images. The table that shows the *indd* address offsets is contained in Appendix B2.

If one compares In-Design file types to the other file types, one notices that *indd* includes a lot of white spaces in the metadata. Other file types consist of unreadable text data.

8.4.6 In-Design template (indt)

The Adobe In-Design template with the extension *indt* represents another In-Design file. This file type is basically a compacted replica of the previous *indd* file type. Although the metadata in this case is similar to that for the default *indd* file, the metadata is located at a different offset

address. Templates are generally used for creating standard documents that can be used as a model to represent a document with minimal features. It is usually used to create a template that can be used at some later stage for importing new content. File sizes range from about three hundred kilobytes to a few megabytes or more depending on the number of alterations made. The file type all use tags that are similar to the *indd* file type and that contain metadata that is similar to *indd.*

The number in the bracket after the inserted image indicates the number of times that a particular image was inserted. Other non-essential digital forensic information includes proximity settings, language settings for application, dimension data for the created doc image, types of fonts used, and other formatting information. The inserted images follow closely upon one another in the metadata entry. The file embeds metadata at different address offsets. The address offsets shown in the metadata table are for the first entries that appear in the file. The table that shows the *indt* address offsets is contained in Appendix B3.

8.4.7 In-Design interchange (inx)

The Adobe In-Design interchange is another type of In-Design file that is known as an In-Design XML Interchange document. The file type is created in smaller sizes of about less than a hundred kilobytes, and it maintains formatting tags that are similar to those of other In-Design file types: they show last file *url* tags and bracketed access numbers. It also retains the name of the document as it was saved with the different prefix *AsMt hDPT="rf_* (1 sided). All three types of time stamps are recorded in the file. The default file name is recorded towards the end of the file, usually in the last ten lines of ASCII code in the binary file. The *xap* and *xmp* tagging style is maintained for each version. The table that shows the *inx* address offsets is contained in the Appendix B4.

8.4.8 In-Design markup (idml)

Adobe In-Design markup with extension *idml* is an XML Interchange document. It is saved in smaller sizes of about less than a hundred kilobytes (compare this, for example, with a counterfeited passport that will occupy a size of roughly thirty kilobytes). The file type does not contain any forensic information that will indicate the creation of counterfeit documents. Most of

its metadata is set in a non-printable format. This means that an investigator as to access the digital forensic information from the log files.

8.4.9 In-Design Snippet (inds)

The In-Design Snippet file type with the extension *inds* is another type of In-Design file. This file does not record information apart from the file default name and the file author. Inserted images are recorded by means of a different tag from other file types. The tag for an inserted image of this type is *<clnk lURL="k_" letg="rk_" laID="k_" lstk="re_lnsk" LnkI="x_c_c.…* The file records only a single inserted image. Extra images are not recorded. The file type maintains a bracketing pattern to indicate the number of times that the image was inserted.

All three of the time stamps are recorded in the metadata. It also maintains the address offset, which undergoes no changes regardless of any amount of editing that is made to this file. These files are generally smaller size and are similar to the *idml* file type. The table that shows the *inds* address offsets is contained in Appendix B5.

8.4.10 In-Design markup Snippet (idms)

The In-Design markup Snippet with the extension *idms* is another type of file from In-Design. It does not contain any metadata showing the name of the author or the default name of the document. The metadata of the inserted images is embedded in the tag *<Link Self="ucf" AssetURL="$ID/" AssetID="$ID/" LinkResourceURI="file*: The file's metadata contains all the inserted images, unlike other *inds*. It also has metadata that shows the string of events for the file. The tagging for string events is *<stEvt:...>*. The table that shows the *idms* address offsets is contained in Appendix B6.

8.4.11 In-copy mark up document Snippet (icml)

The In-copy markup document snippet *icml* is another file in the In-Design graphic design application. The file does not contain any metadata that shows the name of the author or the default name of the file. Inserted images are tagged with a different kind of tagging, namely *Link Self="ucf" AssetURL="$ID/" AssetID="$ID/" LinkResourceURI="*file: The file only records a single entry for any inserted image, as is the case with *inds*. The table that shows the *icml* metadata address offsets is contained in Appendix B7.

A summary of the file signatures for the eleven file types that have been discussed in this chapter is set out in the section 8.5.

8.5 Summary of all the file signatures discussed in this chapter

Table 8.7 shows that template files take the signatures of their default file types. Thus, for example, file *indd* and *indt* have the same file signature. File *ait* also has the same signature as the default file type *ai*. It is also noticeable that markup file types all have the same signature: *icml*, *inx*, *inds* and *idms* files, for example, all have a similar signature. A digital forensic investigator can use the information in this signature table to identify the particular file for the graphic design application in question.

8.6 Summary of content examination

Table 8.8 shows metadata in combination with examples to illustrate how digital artifacts may be recognised from a tag or a prefix. The contents in this table consists of a random selection that the research made from the various files types for the purposes of illustrating these points.

In table 8.8, the first line represents an In-Design document file with an extension *indd.* The file contains a digital artifact that contains the name of an inserted image file. This inserted image is prefixed with a *file:* identifier. The digital artifact shows the full name of the file as well as the extension of the insertion, which is *JPG.*

The second line represents two string events of the saving history. The event actions are named *created and saved* respectively. While the event action is tagged in a <*stEvt:Action*>tag, the time of the action is tagged with a <*stEvt:When*>tag.

File type	File extension	ASCI II	File signature
In-Design	indd	íõØFå½1ïçþt·DOCUMENTp	06 06 ED F5 D8 4D 46 E5 BD 31 EF E7 FE 74 B7 1D 44 4F 43 55 4D 45 4E 54 01 70 0F
In-copy markup document	icml	<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	3C3F786D6C2076657273696F6E3D22312E302220656E636F64696E673D2255544462D3822207374616E64616C6F6E653D22796573223F3E
In-Design XML Interchange document	inx	<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	3C3F786D6C2076657273696F6E3D22312E302220656E636F64696E673D2255544462D3822207374616E64616C6F6E653D22796573223F3E
In-Design markup	idml	PK.....	50 4B 03 04 14 00 00 00
In-Design markup snippet	idms	<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	3C3F786D6C2076657273696F6E3D22312E302220656E636F64696E673D2255544462D3822207374616E64616C6F6E653D22796573223F3E
In-Design Snippet	inds	<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	3C3F786D6C2076657273696F6E3D22312E302220656E636F64696E673D2255544462D3822207374616E64616C6F6E653D22796573223F3E
In-Design template	indt	íõØFå½1ïçþt·DOCUMENTp	06 06 ED F5 D8 4D 46 E5 BD 31 EF E7 FE 74 B7 1D 44 4F 43 55 4D 45 4E 54 01 70 0F
Photoshop	psd	8BPS	38 42 50 53 00 01
Illustrator file	ai	%PDF-1.5%âãÏÓ 1 0 obj	255044462D312E350D25E2E3CFD30D0A312030206F626A
Illustrator template	ait	%PDF-1.5%âãÏÓ 1 0 obj	255044462D312E350D25E2E3CFD30D0A312030206F626A
Encapsulated post script	eps	ÅÐÓÆ	C5 D0 D3 C6

Table 8.7: Hexadecimal signatures for the examined graphic design application file types

File type	File extension	Description of Metadata	Example of the Metadata (As presented in a hex editor)
In-Design document	indd	Name of inserted image	file:C:/Users/<username>\/Pictures/dvd%20 picture%20sleeves/Capture_005%20%282%29.JPG
		String events of saving history	<stEvt:action>created</stEvt:action><stEvt:when>2011-05-04T15:13:25+02:00</stEvt:when>stEvt:action>saved</stEvt:action><stEvt:when>2011-05-04T15:15:43+02:00</stEvt:when>
Illustrator Postscript file	eps	Name of application that created the file	%%Creator: Adobe Illustrator(R) 14.0
		Date file was created	%CreationDate: 9/17/2011
		Author name	%%For: <username>\ %
Illustrator file	ai	Metadata Date	<xmp:MetadataDate>2011-05-04T15:51:17+02:00</xmp:MetadataDate>
		Date file was modified	<xmp:ModifyDate>2011-05-04T15:51:17+02:00</xmp:ModifyDate>
		Name of application that created the file	<xmp:CreatorTool>Adobe Illustrator CSX</xmp:CreatorTool>
Photoshop file	psd	Date file was created	<xmp:CreateDate>2011-05-04T14:39:08+02:00</xmp:CreateDate>
		String events of saving history	<stEvt:instanceID>xmp.iid:DE0657134D76E011B00EFDC555D228CB</stEvt:instanceID><stEvt:when>2011-05-04T14:50:23+02:00</stEvt:when>
Illustrator template	ait	Name of user that created the file	%%For: (Pinchers) ()
		Imported images	%%DocumentFiles:C:\Users\<username>\Pictures\Sizzla-Soul Deep-Front.jpg %%+C:\Users\<username>\\Pictures\Tulips.jpg
In-Design interexchange file	incx	Last file path used	%%DocumentFiles:C:\Users\<username>\\Pictures\Sizzla-Soul Deep-Front.jpg %%+C:\Users\<username>\\Pictures\Tulips.jpg

Table 8.8: Examples of metadata from random files

8.7 Summary of tags and prefixes

The following table, Table 8.9 displays the tags and prefixes described in this chapter. The table is set out by describing the information contained in the evidence identifier as contained in the column titled “full description”. The column “associated tag/prefix” represents the tags and prefixes grouped according to description of contents. The last column “associated extensions” represents the extensions that contain metadata containing at least one of the evidence identifiers in the second column.

Full Description	Associated Tag / Prefix	Associated extensions
Events of document creation, can be saving, editing or changing	stEvt <stEvt:when> <stEvt:Action>	Ai, ait, indd, indt, idms
The timestamp for the last date of modification for the document	</xap:ModifyDate>	Ai, ait, eps, psd
The timestamp for the date the document was first created	%%CreationDate </xap:CreateDate>	Ai, ait, eps, psd
The timestamp for the date the metadata was last changed	</xap:MetadataDate>	Ai, ait, psd
The name of the graphic design application used to create the document	Creatortool %%Creator	Eps, psd
The name of the author or the log in name when the document was created	%For	Ai, ait, eps
Is the default name in which a document was named when it was initially created	%%Title x-default Title Rdf:li Link Self AsMt	Incl, ai, ait, eps, inx, icml
The name of an image that was inserted into the document during editing	<stRef:file, %%Document file stRef:lastURL <clnk	Ai, ait, eps, indd, indt, inds,

Table 8.9: Summary of tags and prefixes

8.8 Conclusion

Chapter eight described and discussed in detail the kind of accumulated digital forensic evidence that can be obtained from files that are intentionally generated by a user. In this chapter, the researcher discussed the various kinds of file types that are generated by graphic design applications. In the process the researcher examined and critically analysed eleven different types of files. For each file type, the researcher identified the content by analysing the hexadecimal signatures of the files. After the researcher had examined the signature, the content was identified by examining the metadata contained in each file type. The researcher pointed out that metadata was indicated by an identifier, which was either a tag or a prefix. Because the metadata within the content of file consists of scattered data and of digital forensic information within that scattered data, identifiers are indispensable for pursuing rational digital forensic investigations. The identifiers make it possible to define the type of metadata that is embedded within the identifier. This means an ability to recognise that an identifier such as "*<Link Self*" may indicate the presence of evidence in the form of an inserted bar code in a suspect counterfeit document with the extension *idms*.

The research also pointed out that these identifiers vary from one file type to another, and that the most valuable forensic evidence can be obtained from the names of insertions, from author names, and from the names of counterfeited documents. The researcher pointed out that insertions could be used to identify what had been replaced within any particular document, for example, a fingerprint by utilising the evidence identifier for insertions. For example, *<stRef:file,* contains the full name of an inserted object within a document. Insertions also showed how further examination could help an investigator to understand how an inserted image had been acquired. The researcher demonstrated, for example, how it is possible to identify the inserted image properties such as a signature card or a barcode generator or the kind of camera that had been used to capture the suspect images. The researcher then showed how the names of the authors could be used to identify the name of the individual who had created the file, and how document titles could be used to identify the names of the suspect counterfeited documents.

Since the various kinds of file types have been discussed and described in detail, it is necessary to consider the role of time stamps within a forensic digital investigation. In the following chapter, chapter nine, the researcher conducts a timeline analysis to and demonstrates how the

timeline of activities for creating counterfeit documents is relevant to a successful forensic digital investigation.

CHAPTER NINE TIMELINE-ASSOCIATED EVIDENCE

9.1 Introduction

In the previous two chapters, the researcher explained how an investigator can accumulate sufficient amount of system-generated and user-generated evidence to make a compelling case against perpetrators of counterfeit documents. These processes are represented graphically in figure 9.1In this chapter, the researcher explains how evidence can be accumulated to reveal the timeline of the counterfeiting activities. The "timeline of the activities" refers to that kind of evidence that is based on the interpretation of the time stamps that are automatically generated in graphic design applications. Time stamps are a vital and indispensable part of any forensic digital investigation because they provide potential evidence of when alleged criminal activities occurred. It is therefore essential for all forensic digital investigators to have a clear understanding of what time stamps are and what they are able to disclose about what people have done by making use of graphics software.

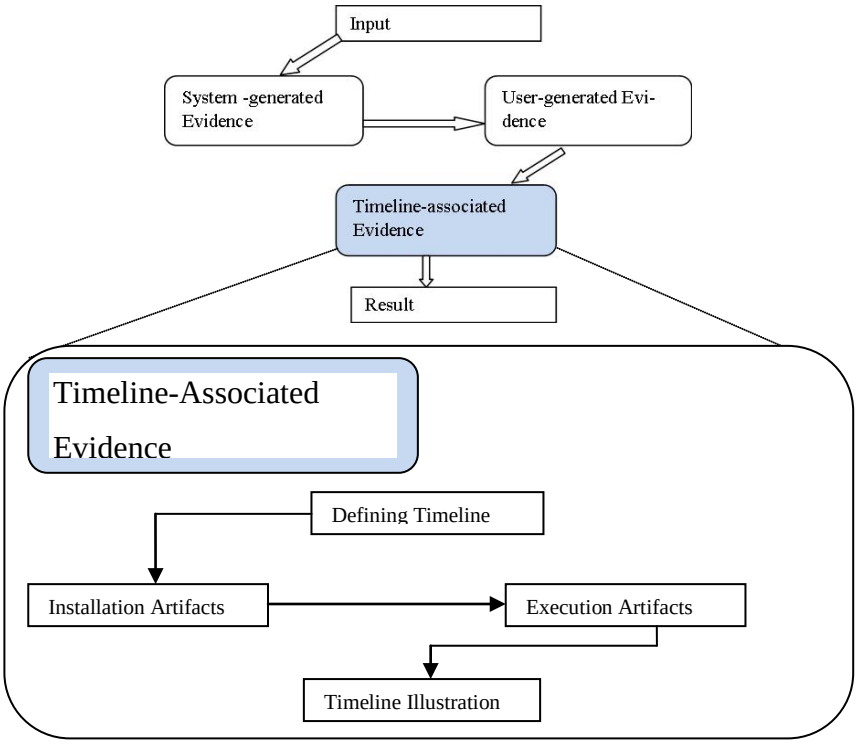

Figure 9.1: A representation of the flow of the model that indicates how timeline-associated evidence is accumulated

It is important to remember that the evidence that remains on the system is both system-generated and user-generated. This is particularly true with regard to time stamping. The time stamps that remain in the system are indicative of a timeline of activities, some of which will be able to explain the activities that took place during the course of counterfeiting documents. The kind of time analysis that is discussed in this chapter provides important supplementary evidence to support the kind of evidence that was discussed in the two previous chapters.

The discussion that follows contains a definition of the system's timeline. This is followed by a description of the artifacts that are generated by installation and the artifacts that are automatically created as a result program execution. The author will then discuss the concept of a timeline in a more general way, and will summarise what was said in this chapter by way of a conclusion. Figure 9.1 details the component elements of this chapter streams.

9.2 Defining a timeline

The timeline can be used to display certain sequential events in chronological order. A timeline may be defined as a horizontal line that is used to represent events that occurred during a specified period, with past events recorded on the left-hand side, and future events described on the right-hand side [27]. Such a timeline may be a schedule of events or even just a simple representation of events that occurred within a delimited period of time. One of the uses of such a timeline is that it can be used to compare and contrast important events. What is more relevant to this study is that timeline is exactly what is needed for locating and pinpointing specific events that occurred within the system at particular moments in the past. A timeline may also be used to demonstrate the root causes of actions or to justify events that occurred within a specific period of time.

Within the context of this dissertation, a timeline is defined as a description of all the events and the specific times at which each of those events occurred on a graphic design application from the moment of its installation until the moment when the application was last used. It is between these two points that the events occurred that enabled a perpetrator to create the counterfeit documents that were used for criminal purposes.

In any digital forensic investigation, it is absolutely necessary to establish a timeline of activities so that the chain of criminal actions can be linked together and explained in such a way that they are comprehensible to a magistrate, to court officials, and to anyone connected with the case who is not familiar with the technicalities of a digital forensic investigation. This kind of explanation is indispensable for obtaining a successful outcome in a criminal case. The suspect will sometimes, for example, deny, during court proceedings, that a particular application was installed and used for creating counterfeit documents. Under such circumstances, it becomes necessary to prove that a particular application was installed and that it was actually used for criminal purposes. This is done so by an interpretation of the time stamps those are associated with the installation and the subsequent uses to which the application was put.

It also becomes necessary for an investigator to identify the individual who was responsible for the installation and for performing the actions that resulted in the creation of counterfeit documents. In order to establish the origins and circumstances of the actions, a digital forensic examiner needs to have interpreted the time stamps and have understood how they indicate particular actions and events on the system. In this chapter the researcher explains why time stamps are necessary for this process, and how the investigator is able to pinpoint the time and place of particular events. In other words, the digital forensic investigator needs to prove the exact times at which a counterfeit document was manipulated and assembled by means of the graphic design application.

During an investigation into a crime in which a particular application was used, the first question would typically be whether that particular application was installed, whether that particular application was used, and, finally, whether any evidence exists to prove potential linkages between whatever actions were undertaken by utilising the application, the computer crime itself, and the incident being investigated.

In what follows, the researcher demonstrates how a timeline reveals the nature of the application installation that was used and the succession of events that led to the creation of counterfeit documents.

9.3 Artifacts relating to program installation

A successful installation is necessary before any application can be used. During an installation, folder and file additions are generated and registry settings are updated. Because artifacts vary in dependence on the application being examined, it is important for an investigator to be able to identify the digital artifacts that are generated by the graphic design applications in this study. In order to connect the digital forensic artifacts to the installation of a graphic design application, it is required first to ascertain the identity of the package that was installed. It is important to know whether it was a creative suite package or a single application that was installed. If the application was installed as a suite package such as the Adobe Master Collection CSX package, then the installation is identified from the acrobat registry key. If it was a single application that was installed, then the installation is identified by locating the registry key of that particular application. In the first case, the values for the application settings, the installation time, the installation date and the installation path are obtained from the registry key, *HKEY_CURRENT_USER\ Software\Adobe\ Adobe Acrobat\9.0\Installer* (as is highlighted in the smaller circle in Figure 9.2). Figure 9.2 illustrates a view from the registry editor, *regedit*.

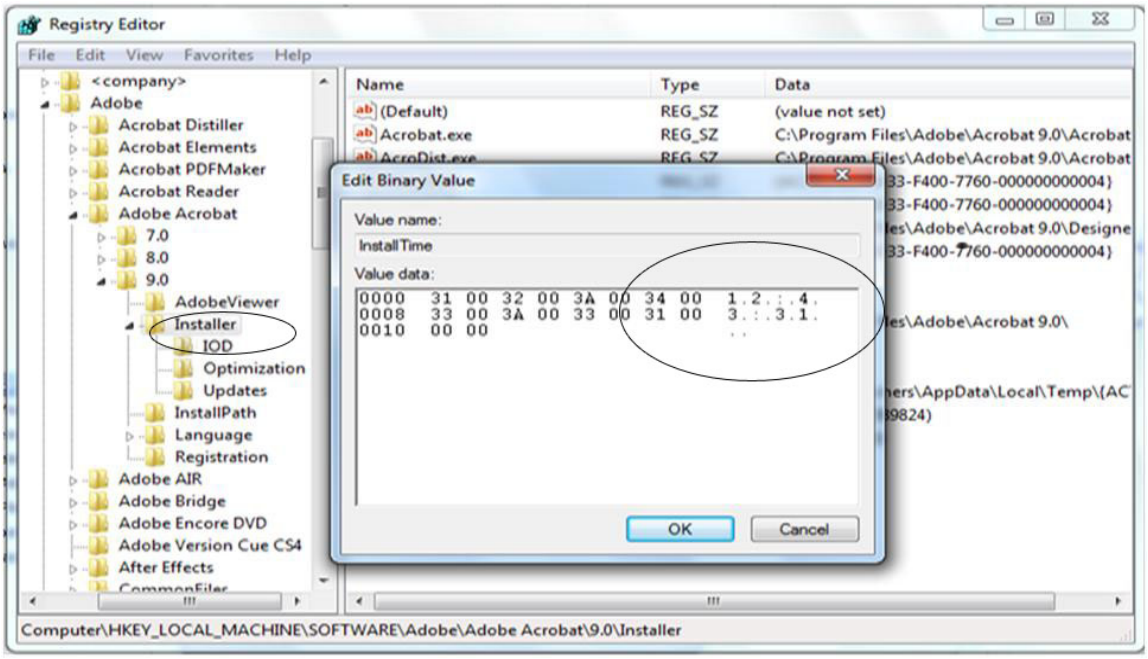

Figure 9.2: Registry view of Acrobat installation time

It is in this place that one finds the installation date with value *InstallDate* and installation time with value *InstallTime* (as indicated by the larger circle in figure 9.2 reflecting the time 12:43:31to a second value). When a single application such as Adobe Photoshop has been installed, the timestamp digital forensic artefact can be identified from the key *HKEY_LOCAL_MACHINE\ SOFTWARE\Adobe\Photoshop\X.0\ ApplicationPath.* – where*X* representsthe version of the application. If these keys are found in the registry, they indicate that the application was indeed installed.

After an installation analysis, an investigator needs to determine whether the installed application was actually used or not.

9.4 Artifacts relating to program Execution

In order to determine whether an installed application was run on a particular system, an investigator needs to conduct a prefetch file analysis. A prefetch analysis involves analysing the prefetch files from the system.

Prefetching was developed by software manufacturers in order improve the systems performance [78]. The purpose of prefetching is to allow regularly used applications to load more quickly by prestaging segments of the loaded code in a specific location so that, instead of having to search for it every time (an operation that results in drive faults), the operating system knows exactly where it is. When a digital forensic examiner finds a prefetch file for a particular application, this indicates that the particular application that is being investigated was indeed run on the system. The creation date of the file will indicate the date on which the application was first run, if one makes the assumption that a previous prefetch file was not deleted and that a new one was created in its place. It is possible to make such deletions because prefetch files are actually temporary files that can be deleted or overwritten by the operating system at any time. The prefetch file contains a 64 bit time stamp that indicates when it was last run, as well as a count of how many times it was run. On Windows 7, the last run of the 64 bit time stamp is located at offset 0x80 (128 bytes) within the binary contents of the prefetch file, and the run count 4 bytes is located at offset 0x98 (156 bytes), as is shown in Figure 9.3 .

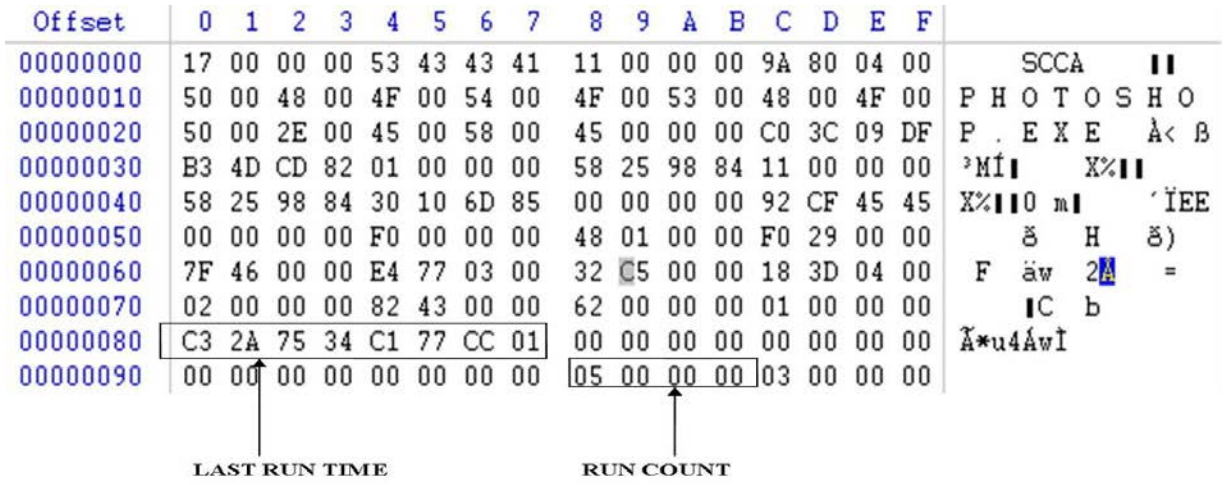

Figure 9.3: Hex editor extract of an Adobe Photoshop prefetch file.

Once the data has been processed, it is written to a **.pf* file in the system's prefetch directory. The **.pf* file will be referenced at a later stage when the program is run again. The system creates a file name by using the application's name, followed by a dash, and then by a hexadecimal representation of the hash of the path of the application, for example, *ACROBAT_SL.EXE DC4293F2.pf*. That means the same program that is run from different locations will create different *.pf* files. On the next occasion when an application is launched, the prefetch directory will be checked for a prefetch file. If it exists, the code within the **.pf* file will be used to launch the application. If, however, the prefetch file is not present (probably it was deleted) the application will still be launched but will load slowly. The prefetch files are located in the folder: *%sytemroot%prefetch*. It should also be noted that an investigator would need administrative privileges to be able to access the prefetch folder. Values that correspond to the number of times the application was launched and a value indicating the last time the application was launched are all contained within the prefetch file. The investigator obtains this information from an analysis of the prefetch file with a hex editor, as illustrated in Figure 9.3.

The prefetch files are therefore able to establish the last run time and run count of an application. Because an operating system generates several different prefetch files, it is necessary for a digital forensic investigator to understand how all prefetch files are generated – all the more so because, in some cases, the name of the prefetch file will not be similar to the name of the application itself. Table 9.1 sets out the Adobe prefetch files that are obtained from

%systemroot%prefetch. It is also necessary to know the names of the prefetch files because these are unique to each particular application and because, in most cases, the name of the prefetch file does not reflect the name of the application. And because these files can be deleted, the tabulated values in Table 9.1 should be used to recover them using recovery tools as and when necessary.

Application Name	File Name
Adobe Acrobat	*ACROBAT_SL.EXE DC4293F2.pf*
Adobe Distributor	*ACRODIST.EXE1C2D8F2D.pf*
Adobe Reader	*ACRORD32.EXE DE3ACC1.pf*
Adobe Collaboration	*ADOBECOLLABSYNC.EXE621E7FA.pf*
Adobe Updater	*ADOBEUPDATER.EXE9AAD898.pf*
Adobe Service manager	*CSXSERVICEMANAGER.EXE B80CD935.pf*
Adobe In-Design	*INDESIGN.EXE C8D4FD6C.pf*
Adobe Tray	*VERSIONCUECS4TRAY.EXE D4DE4E1A.pf*
Adobe Photoshop	*PHOTOSHOP.EXE 4545CF92.pf*

Table 9.1: Adobe prefetch files

Any deleted log files or prefetch files can be recovered by using any of the widely used forensic tools such as FTK and Encase. After the digital forensic investigator has confirmed the application installation and the events that took place in the system, the following step involves the creation and interpretation of the timeline.

9.5 The creation and interpretation of the timeline

The timeline indicates the sequence of a series of events between the installation and the execution of an application. At this point the digital forensic examiner will know when the application was installed and when it was last run. The next task for the investigator will be to establish whether the files that were created were actually created within the time frame defined by the timeline. The investigator will then be able to make use of the actual files, the time stamps, and the modifying dates obtained from user-generated evidence to establish that these files were created between the installation time and the last date of execution of the graphic design application. Figure 9.4 illustrates a timeline analysis in a graphic format. It is the responsibility of the digital forensic investigator to establish whether or not the counterfeit document was created between the time when the application was installed and the last time that the application was run.

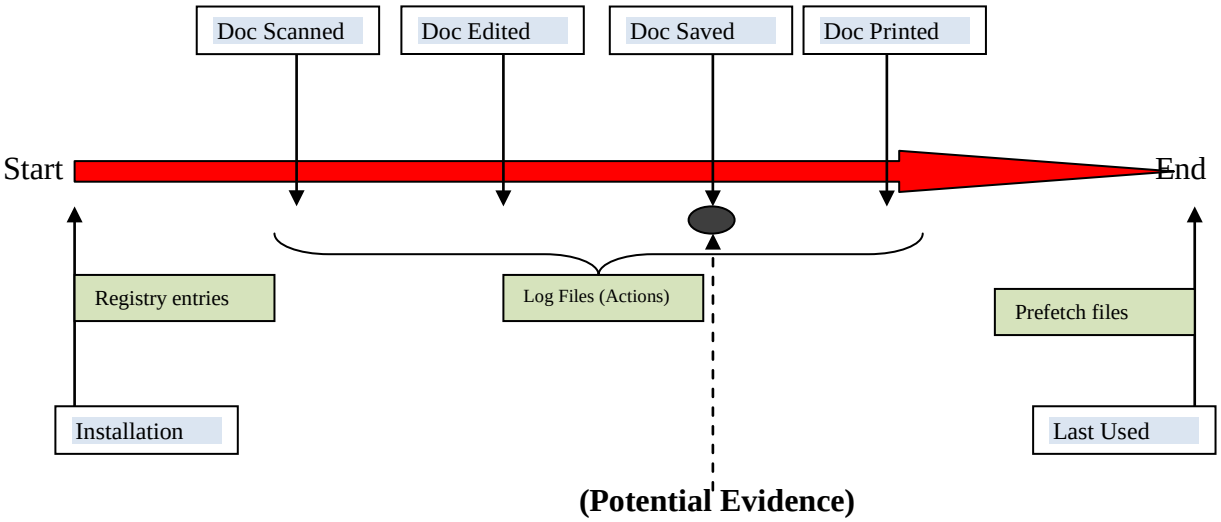

Figure 9.4. Timeline analysis in graphic format

In figure 9.4, **Start** indicates the first action performed on the application while **End** shows the time of the final action taken on the application. The green boxes represent a series of input types while the blue boxes represent the actions that the application user performed by using the graphic design application. The dashed line represents the output from the application that was created by the save action (this is vital evidence in any investigation of this kind). The red arrow shows the sequential direction followed by the timeline.

9.6 Conclusion

This chapter illustrates and explains the nature of a timeline analysis by describing the artifacts that confirm installation and application execution. The researcher obtained the installation artifacts from the registry and the application execution artifacts from the prefetch files. By making use of the artifacts that were created by user-generated and system-generated evidence, the investigator was able to construct a timeline to illustrate the sequence of counterfeiting activities. It is important to qualify these assertions by stating that the artifacts that were gathered are unique to the graphic design applications that are used as a basis for this study.

Timelines of this kind can be used to determine whether or not the actions that were taken during the editing of the document actually occurred within the time period between the time when the application was installed and the time when it was last used. All of this is vital information to support case in court against someone suspected of counterfeit activities. It is of course possible to construct a timeline for other applications if one takes their unique circumstances and settings into account.

The researcher has now presented three chapters about forms of evidence that are user-generated, system-generated and associated with timelines. In the following chapter, chapter ten, the researcher proposes a counterfeiting investigation process that can be effectively used in the investigation of counterfeit documents.

CHAPTER TEN COUNTERFEITING INVESTIGATION PROCESS

10.1 Introduction

In the previous three chapters, the researcher has demonstrated the digital evidence that may be accumulated from specific graphic design applications. In this chapter, the researcher discusses and explains how the system-generated evidence, user-generated evidence and timeline associated evidence can be utilised during a counterfeiting investigation process to obtain sufficient legal evidence to secure a conviction in court of an alleged counterfeiter. In this chapter, therefore, the researcher makes use of all the information and techniques that were set out in the previous three chapters. As was mentioned in the high level model overview chapter, the counterfeiting investigation process involves a two-pronged effort. These two approaches are explained in detail in this chapter. It is important to mention that the counterfeiting investigation process is both application-independent and platform-independent. This means that the kind of investigation process described here can, with the necessary adjustments, be applied to any graphic design application or any operating system.

This two-pronged approach is based on two hypotheses. The first hypothesis, referred to as “without suspect file”, named the blue route is based on a system suspected to be used for counterfeiting purposes, even though there is no suspicious digital evidence. The system in question may, for example, be suspect because hard-copy counterfeited documents were found in the vicinity of the system. The investigator's task is then to establish whether or not if the system had actually been used for counterfeiting or not. The second hypothesis is concerned with investigating a file for which there is prima facie evidence of counterfeiting, and it is referred to as “with suspect file”, also named the green route. This approach is based on the existence of a digital file that is assumed to be implicated in the creation of a counterfeit document. The investigator’s task is then to establish whether the file is actually a counterfeit document or not. Throughout the investigation process, an investigator is concerned with accumulating the kind of digital evidence that was discussed, explained and illustrated in the previous three chapters. This kind of counterfeiting investigation process becomes a subset of any recognised digital forensic model, digital forensic investigation framework or digital forensic process such as the

"Investigation Principles and Processes" working draft ISO/IEC 27043 [81]. This would mean that the counterfeiting investigation process proposed in this chapter makes use of more detailed procedures that will supplement the process recommended in the ISO/IEC 27043 document. What follows in the next section is an explanation of one approach to the two-pronged process mentioned.

10.2 Counterfeiting investigation process (with suspect file)

The investigator who uses this approach assumes that an acquired digital document is counterfeit. It is possible for an investigator to identify a document saved in a graphic design file type and to open-view it by using any pre-installed application. This may result in an assumption that it is a counterfeit document. One may arrive at the same assumption from examining the naming of a document. The document that is named "Jacob_Meyer_passport", for example, is likely to arouse suspicion. When there are reasonable grounds for suspicion, it becomes necessary for an investigator to accumulate sufficient digital evidence to support the suspicion that the acquired document is actually a counterfeit document.

In the counterfeit investigation process, the investigator then undertakes a number of successive logical steps to gather all the evidence which can be used to establish whether the acquired document is counterfeit or not. The progress of investigating a suspect file is illustrated in Figure 10.1.

10.2.1 Signature verification

Once a suspect document has been obtained, the first step is to verify its file type (file format). This is achieved by making use of its hexadecimal signature, as has been shown in the chapter on content identification for user-generated evidence. This means by verifying a files signature an investigation is initialised on the true identity of that file. It is then necessary to document the file signature from the analysis.

10.2.2 Obtain and prioritise metadata

The second step in this investigation is to examine all the necessary metadata from the file. This would include an identification and documentation of the location from which the suspect file was obtained. This can be the location to which the created documents were saved. The

investigator then examines the metadata from a suspect file in the way that was explained in the chapter that describes the implications of user-generated evidence. The investigator takes care to prioritise the metadata by giving a higher priority to inserted images, author names, and document titles because these can later be used for verifying acts of counterfeiting. In the same way, the investigator will assign a lower priority to time stamps even though it is necessary to record the significance of all available time stamps for any subsequent timeline analysis, as was illustrated in the chapter of the explained how a timeline analysis can be assembled.

10.2.3 Determine creator tool

The next step is for the investigator to determine the identity of the application that created the file. This can be confirmed from the metadata that indicates the creator tool or the software agent. Because the artifact identifier varies according to its file type, any such analysis is therefore file type-specific. The identification of the application tool will further the investigation into the application's system files.

10.2.4 Search for log files

Once the application tool has been recognised, the following step is for the investigator to search for the log files that were generated by that application. If the log file's location or name is unknown, then another "system-generated evidence" investigation may be conducted (as described in chapter seven). If the log file's location or name is known, then the investigator will be able to direct scan the locations that are indicated, as, an example, in the *appdata/roaming/<app-name>/* folder.

10.2.5 Examine log files

Once the investigator has located the relevant log file, using a position to undertake a log file examination. The log file is then minutely examined in an attempt to identify those digital artifacts that indicate any counterfeiting actions. These may be indicated by the names of inserted elements and names of suspicious counterfeit files.

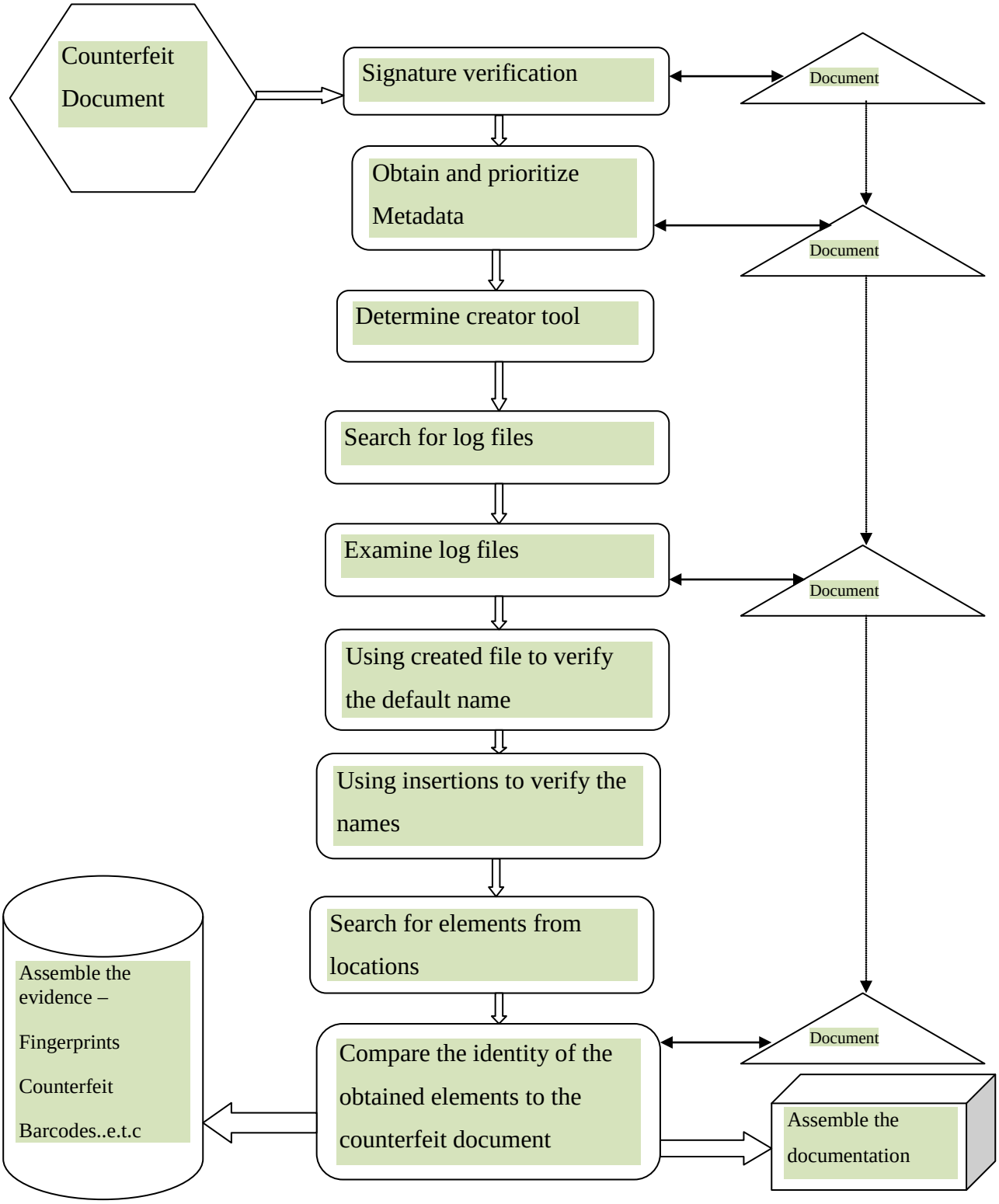

Figure 10.1: The steps undertaken during an investigation into counterfeiting on the basis of a suspect file available

10.2.6 Using created file to verify the default name

An investigator is able to obtain the name or title of a counterfeit document from a system's log file by examining the log file of the graphic design application. These titles could then be compared to the default name of the suspect file in the way that was explained in the chapter that dealt with user-generated evidence. An investigator also needs to obtain the tagging that was used for embedding the title of the created document. The metadata has to be obtained from the actual file type because it will be different from the metadata of other types. In such a case, an identity match will verify that the application was used to create a suspect file.

10.2.7 Using insertions to verify the names

The names of inserted elements from the log file's digital artifacts can be verified by comparing it to similar artifacts that are recognised from the suspect file. This kind of verification is similar to the one that is mentioned in the previous section 10.2.6. An identity match in such a case will substantiate any suspicion that the document was created by inserting the element, for example *ximage.jpg* into *suspect1.psd*.

10.2.8 Searching for elements from locations

As was indicated in the previous section about insertions, the location from which the inserted element was obtained is used by the investigator to scan for other inserted elements. If any such elements are identified, the investigator will examine each of them in turn. Such an examination will enable the investigator to identify the specific camera or tool that was used to create the image. A fingerprint, for example, might have been captured with the aid of a digital signature capture. This would be a positive indication of counterfeiting activity.

10.2.9 A comparison of the identity of the identified elements to the counterfeit document

By using the identified elements, the investigator will be able to determine whether the insertion is the same as the one in the suspect file. The researcher should be able to determine whether, for example, an inserted bar code element is the same as the one in the suspect files, thereby determining if the suspect file is counterfeit or not.

10.2.10 Assembling the evidence

All the identified digital evidence, including the suspect files, the extra suspect files, and the inserted elements, have to be properly and systematically assembled by the investigator who will take great care while doing so to maintain and preserve the integrity of the evidence.

10.2.11 Assembling the documentation

The investigator needs to assemble all the documentation from the initial phase of signature verification to the point of evidence contamination. All this documentation has to be assembled in a careful and logical fashion so that it reflects the metadata, the properties, the file paths, and, above all, the steps of the process that were followed in the investigation.

10.3 Counterfeiting investigation process (without suspect file)

This approach is followed when there are reasonable grounds for suspecting that a particular system was involved in the counterfeiting process. Such a suspicion may have been generated by prima facie circumstantial evidence, or the fact that a hardcopy counterfeit document has been discovered by investigators in close proximity to the suspect computer system. When the correct and due processes of law have been meticulously observed, duly appointed investigators are assigned the necessary jurisdiction to investigate a suspect computer system. The paragraphs that follow describe the kind of process that in investigator will follow when investigating a suspect computer system. The rationale for this approach is to gather a sufficient amount of evidence to prove that counterfeiting has indeed taken place. The progress of the investigation process is illustrated in Figure 10.2

10.3.1 Scanning for installed programs

When they are reasonable grounds for suspecting that a particular computer system might have been used for counterfeiting purposes, the first step that in investigator undertakes is to scan the computer’s system for any installed or pre-installed graphic design applications. These installed applications may be determined from the registry entries, as was explained in the chapter about timeline associated evidence (installation artifacts). Once the investigator has confirmed that graphic design applications were installed, the following step is to identify the log files of the applications.

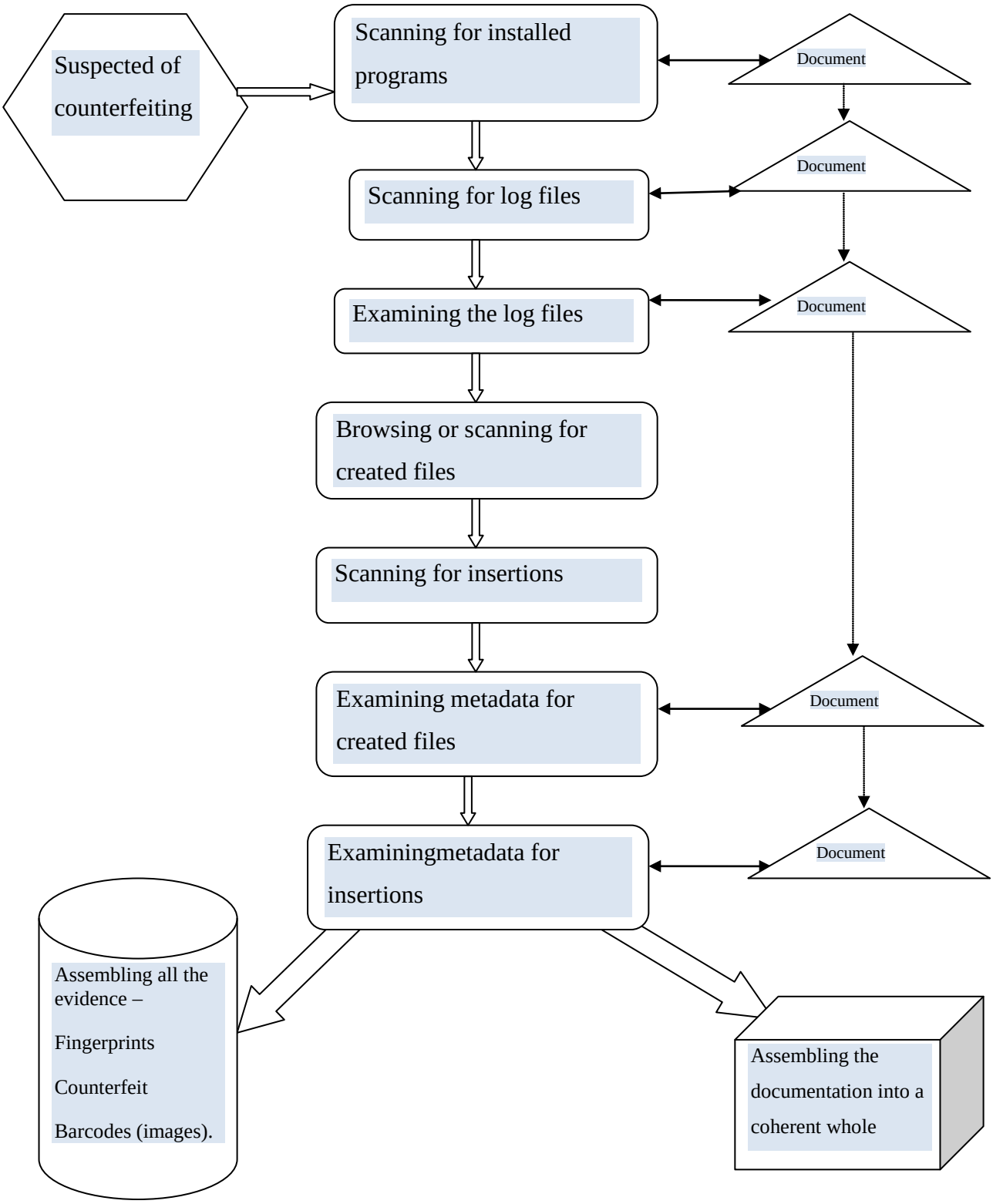

Figure 10.2: Counterfeiting investigation process in the absence of a suspect file

10.3.2 Scanning for log files

Log files may be scanned from program files, a temporary file folder, or from application data paths. When they have been identified, the investigator then pinpoints the exact log files that contain digital forensic information.

10.3.3 Examining the log files

The investigator will then be able to examine these log files for the presence of digital artifacts that indicate document creation.

10.3.4 Browsing or scanning for created files

By making use of those digital artifacts that have been identified for saved files, an investigator will be in a position to search through the relevant files or documents that have been created. Such a search may be based on the file extensions that identify specific graphic design applications or on anomalies or clues embedded in the file names themselves. The file names would be the same as the names of the created files that were identified in the log files.

10.3.5 Scanning for insertions

By making use of the digital artifacts in the log files identified for insertions, an investigator will be able to search for the insertions or elements that were used to create documents. Such a search may also be based on the use of file extensions or the use of known file names.

10.3.6 Examining the metadata for created files

Once the created files have been identified, a forensic investigator will be able to scrutinise the metadata of the files for the presence of digital forensic information that indicates that counterfeiting actions have been performed.

10.3.7 Examining metadata for insertions

Once an investigator has located and identified the inserted elements, the metadata can then be scrutinised for the presence of any digital forensic information that indicates how the insertion was acquired. Such digital forensic information could be names of digital signature capture machines or fingerprint capture machines.

10.3.8 Assembling the evidence

It is necessary for an investigator to assemble all the acquired digital evidence, including the suspect files, the extra suspect files, and the various inserted elements, into one coherent whole while maintaining the integrity of the evidence.

10.3.9 Assembling the documentation

The investigator should then assemble all the documentation from the initial phase of signature verification to the point where indicative evidence was generated. It is necessary for all documentation to be combined into one coherent and logical whole that reflects the metadata that was identified, the properties, the file paths, and, above all, the steps of the investigatory process that the examiner followed throughout the investigation.

10.4 Conclusion

This chapter illustrated a two-pronged investigation approach for identifying counterfeiting actions and for assembling the necessary evidence to secure a conviction in a court of law. The two main differences in the nature of investigation were necessitated by the fact that, in some investigations, the investigator is given an actual document or object that is suspected of being counterfeited, whereas, in the other kind of investigation, there is no actual document on which to base an investigation. Both of these approaches can be used, with suitable modifications, to investigate any graphic design application on any platform. The second kind of approach is therefore that in which the investigator is given no suspect document. Such an investigation proceeds on the assumption that a particular computer system was involved in the counterfeiting process. The first kind of approach is based on the assumption that a particular document is in fact a counterfeited document.

Such a two-pronged investigation process can be used to gather digital forensic evidence of counterfeiting from any kind of tool or graphic design application such as the Adobe graphic design applications that were selected for the purposes of this study. Now that the research has explained and described how coherent evidence can be accumulated to obtain a conviction in a court of law, the researcher will now turn his attention to the design of an appropriate graphic image forensic tool. In order to understand what may be expected from such a tool, it is

necessary first for the researcher to discuss and describe the requirements that have to be met before any digital forensic evidence can be assembled from graphic design applications. The next chapter (chapter 11) will therefore describe the requirements that have to be met before an investigator will be able to obtain digital forensic evidence from graphic design applications. The chapter then discusses the prototype designed to demonstrate the purpose of this study.

Part IV: Prototype

CHAPTER ELEVEN PROTOTYPE

11.1 Introduction

The previous four chapters were dedicated to explaining and discussing the components of the model. It has already been stated that although graphic design applications can be used to create counterfeit documents, there are no currently available digital forensic tools that have been specifically designed for scrutinising a system that is suspected of having been the medium for the creation of counterfeit documents.

In this chapter, the researcher identifies and explains what is required for accumulating digital forensic evidence. The requirements outlined in this chapter are mainly used as a guideline for implementing the prototype. It should be noted that these model requirements may also be used as a basis for developing those more detailed procedures such as the international standard Investigation Principles And Processes working draft that is contained in ISO/IEC 27043.1 [81] (as was mentioned in the previous chapter).

Thereafter the researcher describes and explains the development and function of the prototype developed for the purpose of gathering digital forensic evidence from graphic design applications. The researcher demonstrates how the digital evidence assembled was incorporated into the prototype of the tool. The tool resembles a functional representation of the research that the researcher conducted. The researcher also defines the *prototype* and demonstrates how the tool can be used. The chapter ends with a conclusion.

11.2 Model Requirements for gathering evidence from graphic design applications

The table of requirements that the researcher lists in Table 11.1 are assembled from all the data and information contained in the previous chapters. The requirements have to be met in order to gather evidence from graphic design applications.

While it may be noted that the requirements set out in the table are not listed in any particular order, the investigator has ordered the requirements described in this section so that they can be used sequentially for the purpose of identifying and describing counterfeit documents. Each of

the requirements set out in Table 11.1 is explained in more detail in each of the subsections that follow.

	Requirement
1	Identify the evidence that resides in the application or in the system files.
2	Ascertain the file types that have been generated by the application in question.
3	Verify the identity of the recognised file types.
4	Identify the source of the evidence.
5	Establish the path locations on the basis of available evidence.
6	Retrieve the component content of the evidence.
7	Identify the digital artifacts that function as the repository of evidence.
8	Identify the binary addresses for the digital artifacts.
9	Establish how the digital artifacts may be identified.
10	Indicate the digital artifacts that confirm all previous events or actions in the system.
11	Scrutinise and interpret the evidence contained in the digital artifacts.
12	Examine the time stamps relevant to the evidence by conducting a time analysis.
13	Assemble a coherent narrative about the digital artifacts that will be able to establish a conviction in a court of law.
14	Describe the process that the investigator utilised during the investigation.

Table 11.1: Requirements for assembling digital forensic evidence from graphic design applications

11.2.1 Identify the evidence that resides in the application or in the system files.

In those cases in which an application was used for committing a digital crime, an investigator has to search the system files to find corroborative evidence. The system files for a windows application may, for example, be found in the program files, the temp files or in the *appdata* path. The researcher demonstrated in the experiments described in earlier chapters that forensic evidence is created and maintained by the application that the counterfeiter used to commit the digital crime. An ability to identify and explain the nature of the digital evidence that was

obtained from an application presupposes that an investigator has a clear conception of the steps that a counterfeiter would have had to take to commit the crime.

11.2.2 Ascertain the file types that have been generated by the application in question.

The experiments conducted by the researcher clearly showed that graphic design applications generally save created files in several different file types. Because of this, it is necessary for an investigator to be familiar with all the types of files that can be generated by the graphic design application in question. An ability to recognise each of these file types enables an investigator to recognise and identify the file type in question. This can be accomplished because each file type has a different style or method of storing evidence.

11.2.3 Verify the identity of the recognised file types.

The identity of any recognised file type has to be verified by its file signature. The signature indicates the true identity of a file because any file extension can be changed by an offender to conceal the actions taken when creating the counterfeit documents. This signature verification process is to be carried out for each recognised file. Thus, for example, the signature of an *idml* file has to be verified to be *PK*.

11.2.4 Identify the source of the evidence.

It is necessary for an investigator to identify the source of the digital evidence. If, for example, the evidence was acquired from the application path, it would be necessary to examine the log files of the particular graphic design application that was used. This process would supply an additional amount of digital forensic evidence from the graphic design application in question.

11.2.5 Establish the path locations on the basis of available evidence.

Whenever evidence has been accumulated, the path on which it was located should be noted. This path location can enable an investigator to accumulate additional evidence because it is possible that the location might have been used by the offender for saving any created files from the application paths.

11.2.6 Retrieve the component content of the evidence.

It is necessary for an investigator to scrutinise every bit of evidence that is associated with the contents of a file. The researcher's experiments enabled him to realise that an analysis of the content of a file can enable any investigator to extract the digital forensic information that is required to confirm counterfeiting activities. This kind of activity is referred to as a metadata analysis.

11.2.7 Identify the digital artifacts that function as the repository of evidence.

A careful examination of the evidence enables the researcher to identify digital artifacts from their content. Such digital artifacts may be author names, the names or titles of documents, or insertions. Such digital artifacts enable the researcher to determine whether there is sufficient evidence for asserting that a document has been counterfeited.

11.2.8 Identify the binary addresses for the digital artifacts.

It is necessary for a digital forensic investigator to be able to identify the binary addresses of the digital artifacts. These addresses enable the investigator to observe the distribution of digital artifacts within a file structure. Knowledge of how digital artifacts are distributed can facilitate the examination process.

11.2.9 Establish how the digital artifacts may be identified.

An investigator can identify digital forensic artifacts by their tags and prefixes. An investigator should, for example, be able to recognise that the prefix *%%For:* enables the identification of a digital artifact because it represents the name of the file author or the log in the name on the system.

11.2.10 Indicate the digital artifacts that confirm all previous events or actions in the system.

An investigator should be able to recognise and identify those digital artifacts that indicate that certain events such as, for example, the scanning, editing, saving or printing of documents, have been performed in the system. Metadata generally reflects a string of action events, as was

explained in a chapter about user-generated evidence. Such strings of events are vital for understanding the sequence of all relevant previous events in the system.

11.2.11 Scrutinise and interpret the evidence contained in the digital artifacts.

An investigator should carefully scrutinise any digital artifact that might be the source of further forensic evidence. Such additional forensic evidence may have been, for example, the insertion of a bar code or a fingerprint. It would be necessary in such a case to determine from an inserted fingerprint the particular tool from which it was created.

11.2.12 Examine the time stamps relevant to the evidence by conducting a timeline analysis.

Time stamps that are associated with modification, metadata and creation dates are crucially important evidence to proving the installation of the application that has generated digital artifacts. An investigator can use the time stamps to conduct a time analysis. Such a time analysis will determine whether or not the suspect document was actually created within the time frame between the moment when the application was installed and the time when it was last used.

11.2.13 Assemble a coherent narrative about the digital artifacts that will be able to establish a conviction in a court of law.

Digital artifacts can be verified by determining the different sources from which they were obtained. Thus, for example, the name of the counterfeit documents that were created can be obtained for the binary structure of the evidence and from the system log files. Being able to connect the same artifacts to different sources strengthens the evidence that will be presented during the course of court proceedings.

11.2.14 Describe the process that the investigator utilised during the investigation.

It is important that an investigator be able to delineate, during the course of court proceedings, the process followed to obtain the evidence that is presented to the court. It is possible to use any of the components of the proposed two-pronged counterfeit investigation process to examine a suspect document or a system that is suspected of having been used for counterfeiting purposes. It should also be noted whether the investigation took place with or without any suspect document being available for investigation.

All of these requirements are an indispensable part of the process of gathering digital forensic evidence from graphic design applications. The researcher established these requirements by conducting the experiments that are described in the previous chapters of this study. By meeting these requirements, an investigator will be empowered to accumulate and interpret a sufficient amount of digital evidence from graphic design applications to enable the successful prosecution of perpetrators of counterfeiting activities in a court of law. Now that the researcher has established the necessary requirements for any digital forensic investigation, it remains for him to describe a tool developed for conducting a forensic analysis of graphic design applications. In the following section the researcher describes the development of the prototype, a graphic design image forensic tool (GIFT).

11.3 Defining *the prototype*

In order to accomplish the purposes for which this study was designed, the researcher designed a tool that would assist an investigator to accumulate evidence indicative of how counterfeiting activities can be carried out by making use of graphic design applications. By using the Java programming language on a Net-beans platform, the researcher developed a prototype tool to demonstrate the purposes for which it was designed for this research study. The tool was named "Graphic Image Forensic Tool" (GIFT) because it is enables an investigator to examine and perform digital forensic tasks on the basis of the graphic design applications that were selected for this research. GIFT[1] is capable of performing the following functions, which the researcher based on the requirements that were described in the previous chapter:

- Searching for a suspect document on a system
- Searching for the log files of a graphic design application
- Browsing and displaying documents in a directory
- Verifying file signatures
- Searching for graphic design application file formats

[1]Please note that not all the requirements have been incorporated in this tool because of time limitations. But the requirements that were not implemented have been recommended for future research. (Author)

- Reading, extracting and displaying the contents of a log file
- Reading, extracting and displaying all the metadata from a suspect file
- Recognising and displaying essential digital forensic information
- Identifying the time stamps from suspect files

Figure 11.1 shows the Graphical User Interface (G.U.I) for the Graphic Image Forensic Tool that the researcher developed. The tool consists of eight tabs with various functions that enable a user-friendly interface. There is also a general tab briefly describing the tool functions. What follows is a detailed description of the functions of the tool.

Figure 11.1: Graphic Image Forensic Tool (GIFT)

The tool consists of several tabs for searching files. These search functions can be conducted with other tools or the operating system. The search functions have only be added to allow the application flexibility for the user to use the search function within one tool. Nonetheless the main functions developed for the tool are for signature verification and file examination as this form the basis to which the research was conducted. Therefore the tool user does not necessarily need to use the search tabs but specifically to examine files presented for investigation. The tool

is used based on the user intentions, meaning that the tabs can be used in any particular order and the user does not need to use all tabs to examine a file. The tool has been developed mainly to demonstrate the ability that the research conducted is tool implementable.

11.4 Graphic Image Forensic Tool (GIFT)

The functions of the tool are illustrated by means of a practical example that involves the investigation of a suspect file that is assumed to be counterfeited. This is the kind of investigation that the researcher designed when an investigator has been given a suspect document, previously referred to as the "with suspect document" approach. Consider the following scenario. An original copy of an identity document was acquired, scanned and saved on a system by an offender. The offender later edited this copy by using a graphic design application, and in the process, changed the bar code and image of the human face. After completing the editing, saved the newly created document on the system, and later printed it. This electronic counterfeit document was recognised by another individual who presented it to the authorities as a suspect counterfeit document. Thus far, the scenario that has been described is exactly similar to the kind of procedure that the researcher recommended in the chapter that described the scenarios that were chosen for the experiments that the researcher conducted. In order to achieve greater clarity, the illustration of the procedure in this chapter incorporates two file types, one called *double identity.psd* and another called *pASSP2.indd*.

As the investigator makes use of the counterfeiting investigation process, particularly the "with suspect document" approach an investigator could use GIFT to facilitate his investigation into whether the document is indeed counterfeit or not. The main difference between investigations with a suspect document and those without a suspect document, are in the initial steps of the investigation. Since the output result will always be the same, what is explained here is only an investigation with a suspect document. In order to provide an explanation of the tool functions, the researcher has made the titles of the sub-headings in this section the same as the names of the tabs in the tool. The investigator is then required to tackle each of the following steps if GIFT is used as the basis for a forensic digital investigation: a suspect file search, a directory search, a file extension search, a signature verification, a target examination, scanning the log files, displaying the full binary, and generating a report. An explanation of the tool 'forensic evaluation' and 'file exception handling' function will follow later.

Each of the subsections conforms to the essential requirements for a forensic digital investigation, as they were listed and described in the previous chapter.

11.4.1 Suspect file search

Given that the name of the suspect file is known, the investigator can use the tab "Suspect file search" to search for the suspect file on the system by specifying the full name and the directory in which the search is assumed to be located. By initiating the find file command, the tool begins to search for the file. Once the search has been completed, the search results will be displayed in a display panel entitled "Binary display". The displayed information will include the last date of modification of the suspect file, as recorded by the system. Figure 11.2 shows the results of a "*double identity.psd*" file search. The assumption in the scenario is that the file entitled "double identity" was given to the investigator and identified as a prima facie suspect file. The investigator could also conduct a directory search as an alternative to the suspect file search. The directory search is explained in the following subsection.

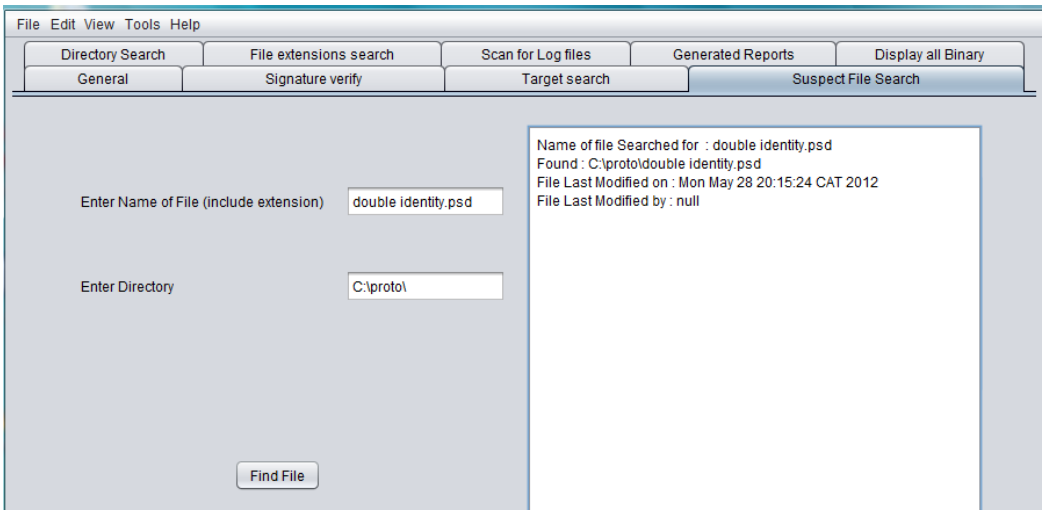

Figure 11.2: Suspect file searching

11.4.2 Directory search

If an unknown suspect file is assumed to be located in a certain directory, or if a directory is assumed to have been used for the saving of the counterfeited documents, an investigator can search the directory by using the tab *Directory search*. Once it has been given the name of the directory, GIFT will be able to search the directory and display all the contents of that directory,

including its sub-folders, as is shown in Figure 11.3. As an alternative to a directory search, an investigator could also conduct a file extension search. The file extensions search is discussed in the following subsection.

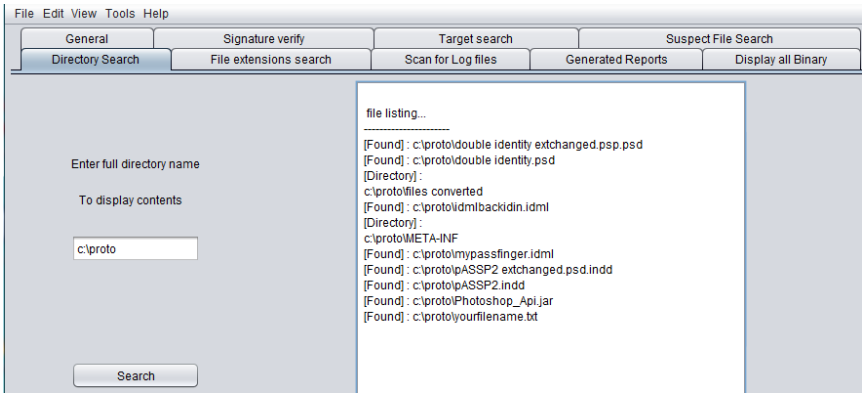

Figure 11.3: Directory searching

11.4.3 File extension search

If the suspect file's name is not known, but is assumed to have a certain extension, the investigator can initiate a search for files with that specific extension. Once the command search has been activated after the tool has been given the name of the extension, GIFT will browse and locate all those files that are tagged with the suspect extensions. It will then display all these files, as shown in Figure 11.4. After GIFT has identified the suspect file, investigator will engage in a file identification analysis by means of the process of signature verification. The signature verification process is described in the following subsection.

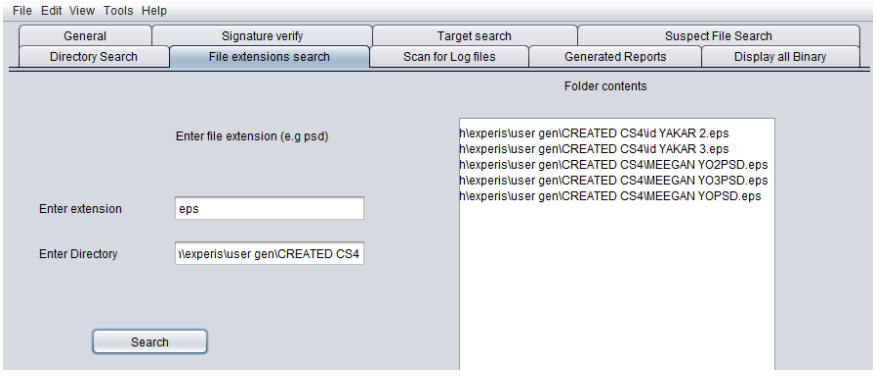

Figure 11.4: File extension search

11.4.4 Signature verification

Once the suspect file has been located, the investigator will scrutinise its signature. As was explained in chapter eight in the section about user-generated evidence, it is up to the investigator to verify the signature in order to find out whether the suspect file has been assigned its true identity. This constitutes a crucial part of the evidence in the criminal case that will follow. Once an investigator has been appraised of the type of file dealt with, it becomes easier for him to recognise how evidence of criminal activity will be embedded through the use of tags. This is the case because different file types neutralise different styles of embedding evidence.

GIFT analyses the binary signature of the file by using the signatures, as was explained in chapter eight. Thus, for example, if a file contains an *idml* extension, the tool will search for a *P* and *K*, at byte 0 and byte 1 respectively. If the signature is the same as was described in chapter eight, the GIFT display will indicate the true signature for the selected extension. But if the signature does not match, the message "Signature is false" will be displayed. Figure 11.5 illustrates the signature verification for an *indd* file. On the basis of the research that conducted, and from information found in online signature sources, the researcher has made the assumption that only the extension *idml* has the signature of *PK*. In this particular examination of the suspect file "double identity", the investigator verified the signature because, as *8BPS*, it is the correct signature for a *psd* file. In the following step of the forensic digital investigation, the investigator uses GIFT to conduct a target examination. The target examination will be explained in the following subsection.

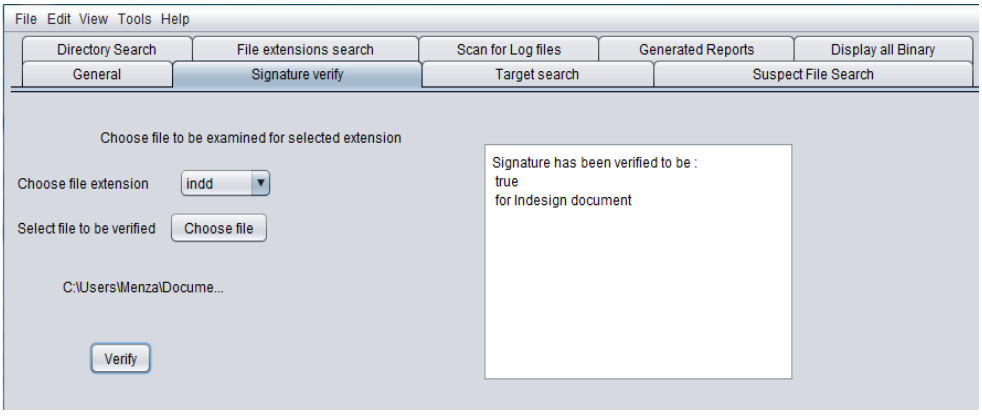

Figure 11.5: Signature verification

11.4.5 Target examination

After the signature has been verified, the next step is for the investigator to examine the file by using the tab, *Target examination.* By using the button Select file, an investigator will be able to browse and locate the suspect file by using GIFT's file browser, as shown in Figure 11.6.

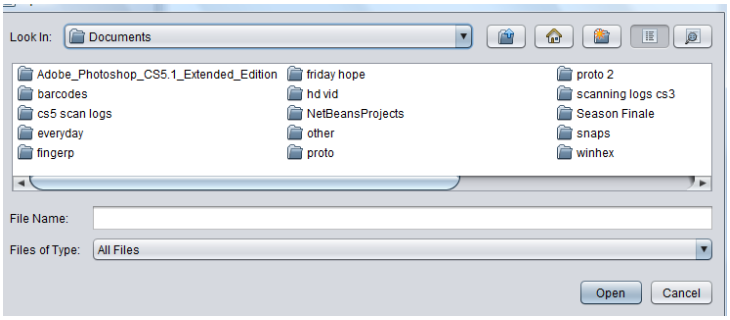

Figure 11.6: File browsing

GIFT will then display the identified file in a text field entitled "Selected file". It is necessary first for the investigator to select the file extension because this will ensure that GIFT searches only for the required digital forensic information that is specific to that extension. Since each file type utilises different digital forensic information, the selection of the applicable file extension will maximise the search results. GIFT will then analyse the selected file when the researcher activates the Analyse button. GIFT will then display all the digital forensic information on the right panel – information that will include a description of the metadata, which is part of the evidence that can be collected from the file. GIFT can then generate a report from the displayed information after the investigator has supplied a report name. The report will be saved in a *doc* format. Figure 11.7 illustrates the displayed metadata from an *indd* file and the message that is shown once a report has been successfully created.

If the counterfeiter inserted images such as digital signatures, fingerprints or barcodes into the document, the content of these insertions would be listed under the title "Name of object insertions". If such objects are identified in the way shown in figure 11.8, an investigator will conduct a further examination for each of the identified objects. GIFT will then be able to browse and further analyse the relevant contents. *File editing instances* refers to a string of events that occurred during the editing the document. Thus, for example, as was indicated in figure 11.8, <*stEvtwhen*> refers to the date of the event file created <*stEvtaction*>,and the metadata was then changed (<*stEvtchanged*>) by an Adobe In-Design application

(*<stEvtsoftwareAgent>).* A table with full descriptions of the tool displayer, shown as a description of contents, is contained in Appendix C1.

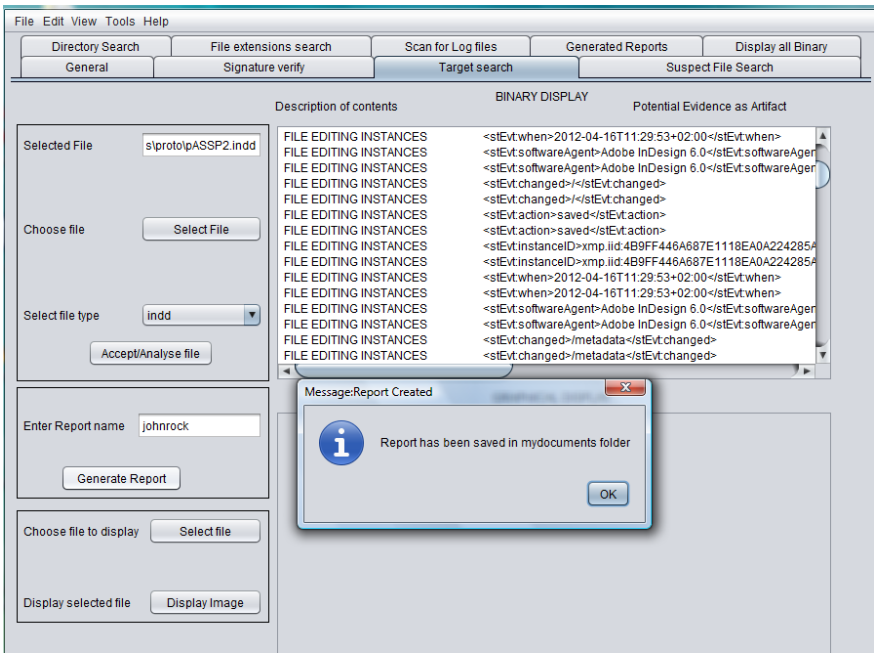

Figure 11.7: Target examination and displaying metadata

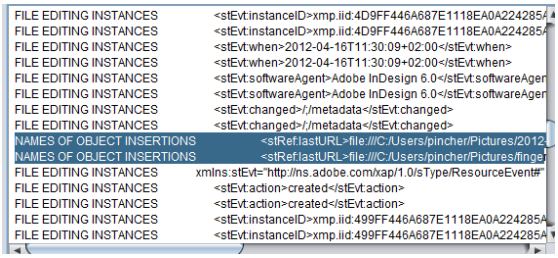

Figure 11.8: The recognition of inserted objects

By utilising the *select file to display* function on the middle panel, it is possible for the investigator to locate the inserted image file, which GIFT will then display in the bottom right-hand panel entitled *Graphical display*, as shown in figure 11.9. The figure shows that an identified bar code was inserted into the document, the suspect file. The bar code is one of the fraudulent components of the suspect file that the researcher is investigating. GIFT demonstrates that the suspect file was counterfeited by having a bar code inserted into it. The same display

function is repeated to identify the human face inserted into the suspect document. The GIFT displayer only reads and displays *jpg* and *png* file types. Other file types can be displayed using operating system image viewers or any image readers. The tool itself could easily be extended to display a larger number of file types. This part of the investigation which identifies inserted images is vital to the forensic process because it enables the identification of the elements or the components of a suspect file. These components would alternatively enable an investigator to determine whether or not a document is counterfeit.

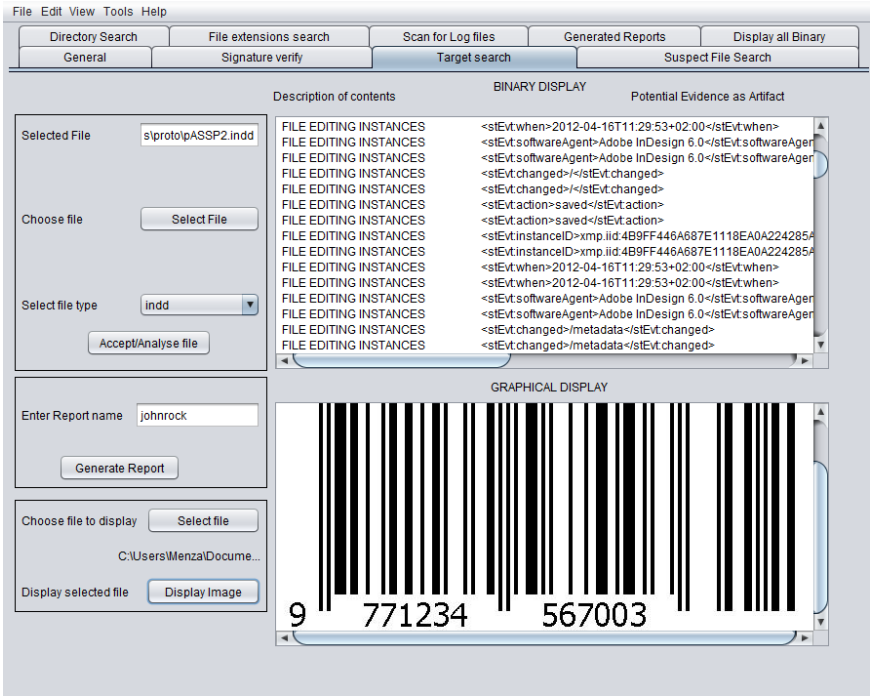

Figure 11.9: Graphic display of inserted objects

Once the metadata from the application tool has been identified, either as *software agent* or *creator* metadata, investigator will be able to look more closely at the software agent. The investigator will be able to examine the particular application that the counterfeiter used by scrutinising its log files.

11.4.6 Scan for Log files

An activation of the tab *Scan for log files* enables GIFT to search for the graphic design application log files on the system. If GIFT recognises the graphic design application that was used, the tool displays the location and the names of the log files, as is shown in figure 11.10. In

this particular scenario, the investigation of the suspect file entitled “double identity”, GIFT recognised that the software application that the counterfeiter used was Adobe Photoshop.

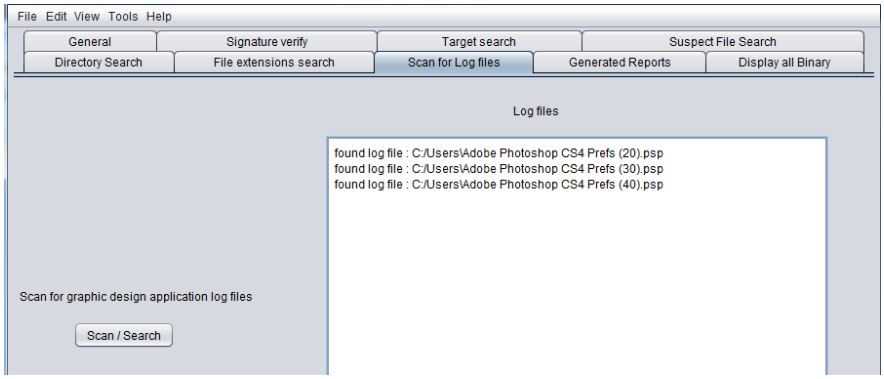

Figure 11.10: Scanning for log files

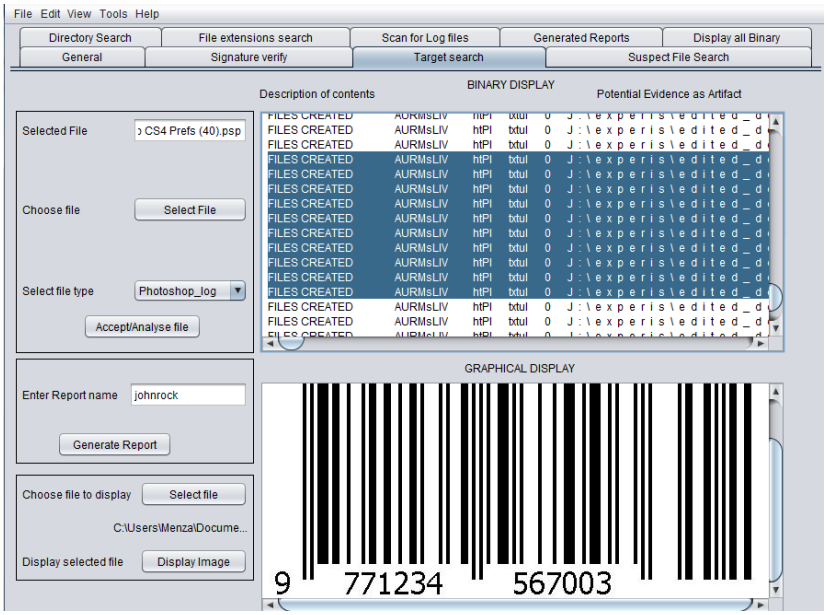

Figure 11.11: Results from log file

The investigator will then be able to analyse the log file in greater detail by using GIFT’s “Target examination” facility again, after having selected the type of log file (*Photoshop log)* from the drop box, as was previously explained. As an alternative, an investigator may decide to conduct a full binary search so as to be in a position to display all the metadata from the log file, as shown in figure 11.12. For the earlier option, the log file contents will be displayed, including the names of the created files, as highlighted in figure 11.11. An investigator has to note that data displayed from a log file can also include program file locations. Therefore one has to scroll to

last page to identify all the required digital forensic data. The alternative method that makes use of a full binary search is described in the following subsection.

11.4.7 Full binary display

An investigator may decide to use a full display if he/she deems it necessary. The full displayer does not provide any description of the metadata, in the way that it did for the target examination. Since the metadata is displayed as a dump on the displayer, the examiner has to search for evidence by scrolling through the pages. The researcher designed this tab mainly for the investigation of unknown log files or unrecognised file types that are outside the terms of reference for this research. The displayer works the same way as a hex editor. It does not have an editing function, but only includes a reader. Figure 11.12 illustrates an example of a full display, which may involve analysing thousands or tens of thousands of pages of metadata. A 1MB *indd* file can contain, for example, 1 300 pages of metadata, and a log file from a graphic design application may be about 15 000 pages long. It is therefore better for an investigator to conduct a target examination.

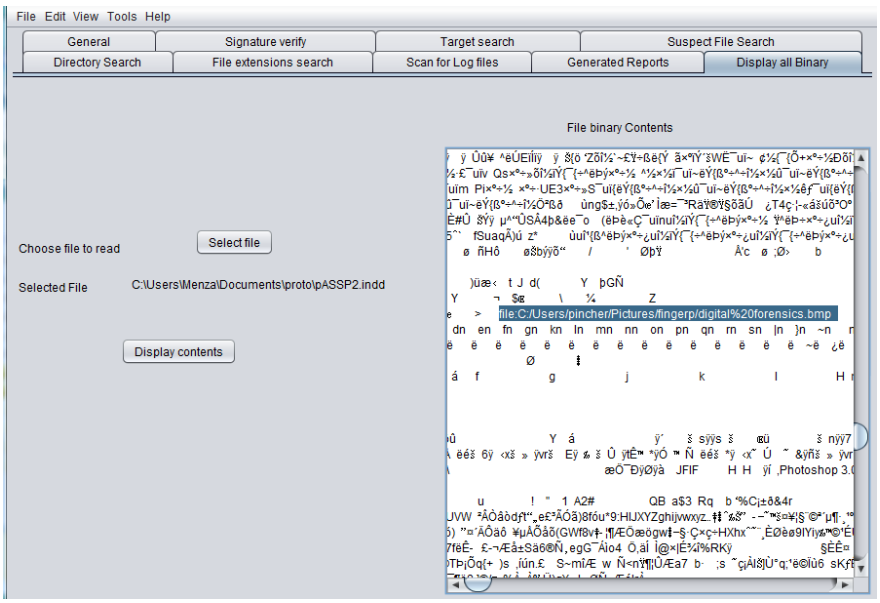

Figure 11.12: Full binary display

An investigator will be able to identify the scan folder paths and the names of insertions from an examination of the log file and the names of the created files. Figure 11.13 shows a list of identified names of created files from a log file that has been examined. By using the name of the

suspect file, the investigator will be able to verify whether or not the document is on the list. The investigator uses this information to determine whether or not particular application under consideration was used to create the suspect file. The investigator also uses the scan folder path to locate the original copy of the document which the counterfeiter scanned. The name of the application that created the file can also be used to determine the time of the application installation and the time when the application was last used, as was explained in chapter nine, the chapter dealing with time-associated evidence.

The investigator uses the same full binary scan to examine prefetch files and printer spool files, as was described in the sections dealing with printer artifacts in system-generated evidence. The prefetch file will display the last time that the application was used, and the printer spool files will display the name of the documents that were printed and the printers that were used. Since the whole purpose of any investigation is to produce a coherent explanation of the evidence in the form of documentation, the researcher designed GIFT to produce the necessary reports.

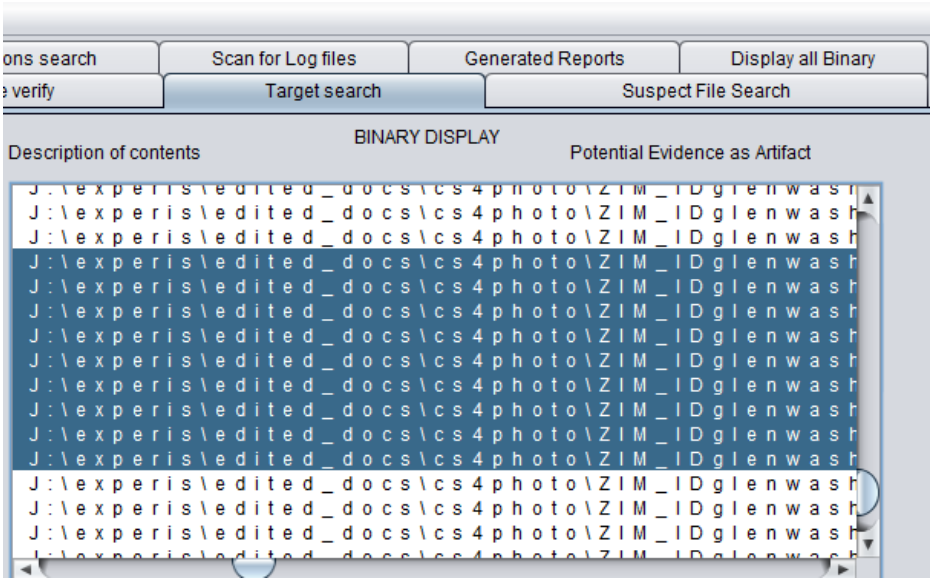

Figure 11.13: Created documents

11.4.8 Generated reports

The reports that are generated from using the “target examination” function are necessary for documenting the examination process. In this scenario, the investigator generated a report on the forensic investigation into the suspect file. Figure 11.14 shows an example of the kind of reports that GIFT generates. In its present state, the tool can only generate reports with a *.doc* extension.

It would nevertheless be simple task to extend the capability of GIFT so that it is able to generate reports to various file types. An alternative for investigations is provided in the next subsection.

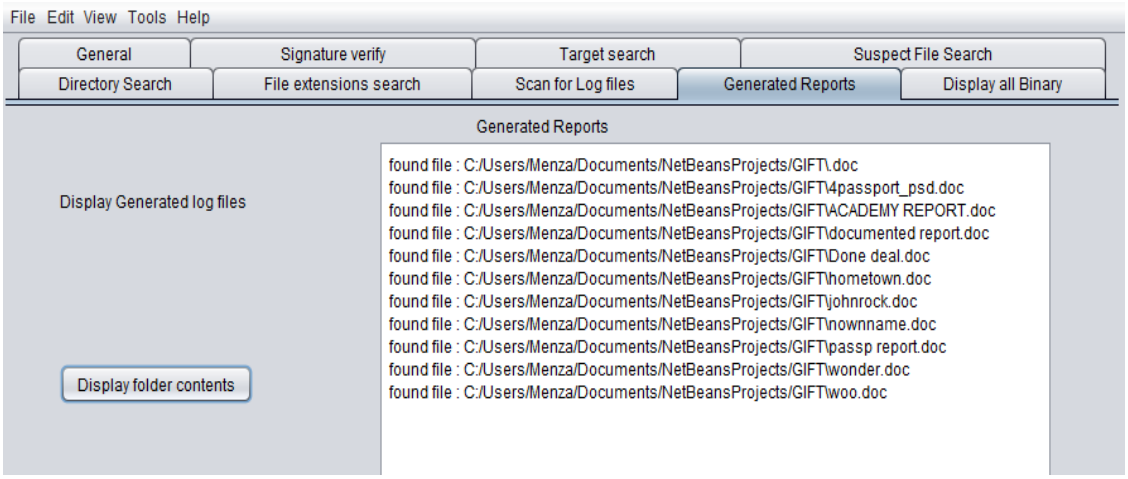

Figure 11.14: Generated reports

11.5 Forensic Evaluation

Given that an investigator wishes to examine a file and accumulate forensic information from the suspect file without using the described process in section 11.4. The “forensic evaluation” tab can be used. By using the button Select file, an investigator will be able to browse and locate the suspect file by using GIFT’s file browser, as shown in Figure 11.15.

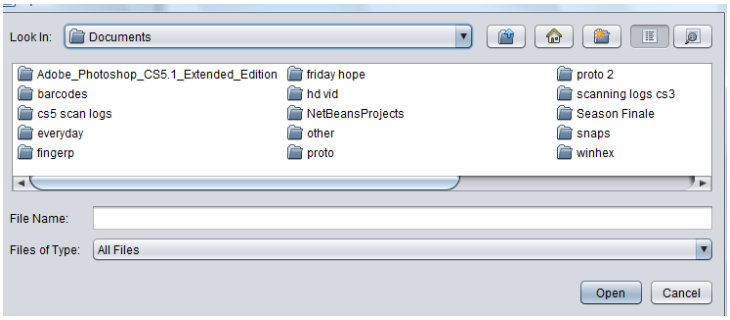

Figure 11.15: File browsing

If an investigator is suspecting a particular application user, one can specify the name in the “Author under investigation field” shown in figure 11.16. The application user can be a name of a well known counterfeiter, and is suspected of having created the document under investigation.

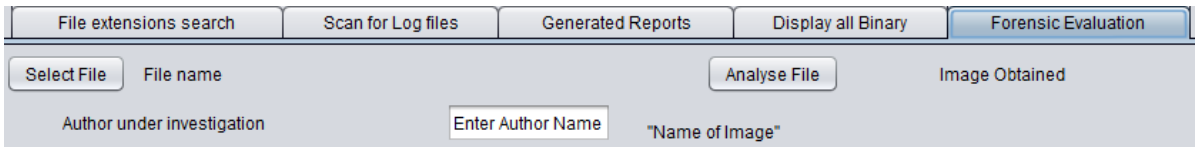

Figure 11.16: Select Buttons for forensic evaluation

After clicking the analyse button, the tool automatically verifies if the author of the document is the same with the one under investigation. If so, it displays “Match found’ in red font. The tool also verifies if the title of the document is similar to the title when the document was first created, if not the same, the tool displays “Title changed” in red font. This serves to inform an investigator the possibility of the counterfeiter trying to hide their actions, as previously demonstrated in section 2.9.1.

The tool calculates the number of times the document was edited, the number of artifacts recognised from the suspect file and the number of images inserted into the file during possible counterfeiting. From these counts, the tool calculates a probability of the document being counterfeit based on the rationale of all counts, the author under investigation and the title of the document changed. The probability is merely used for comparing suspect files and in no way provides an accurate calculation of the possibility of a document being counterfeit. For example, the higher the number of times the document was edited and if there exists an author match, the greater the counterfeiting probability. The probability gives a hint to the investigator to zoom into a file with a higher probability in a bid to acquire valuable counterfeiting evidence. The probability calculator is included in appendix C2.

The tool automatically checks the names of the inserted images and displays the images in three separate image displayers. This gives the investigator the ability to evaluate the suspect file without having to browse through the contents as explained for “target examination”. Figure 11.17 displays the results from a file under investigation for forensic evaluation.

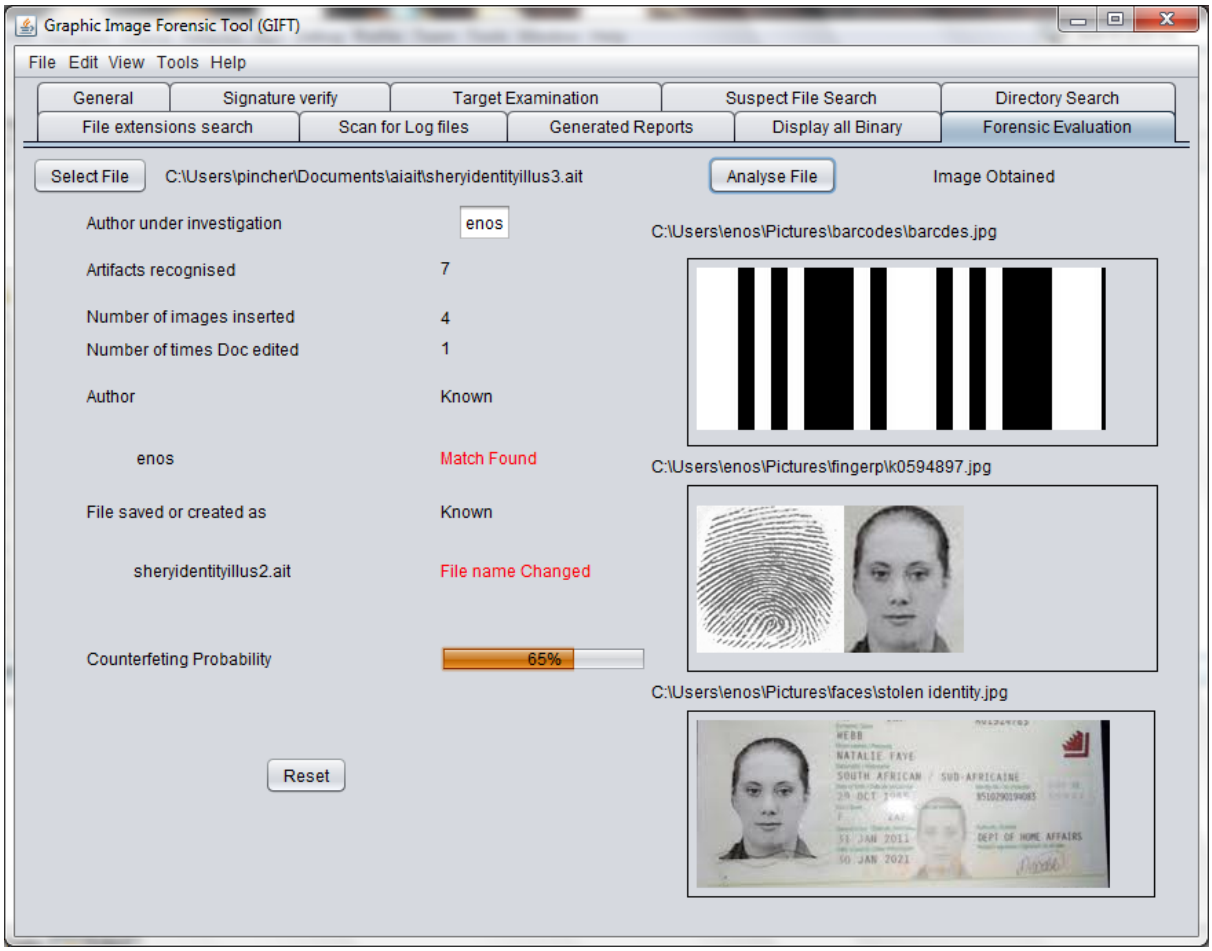

Figure 11.17: Forensic evaluation of a suspect file 1

Figure 11.17 displays an example of a suspect file loaded for forensic evaluation. The investigator supplied the author of the document under investigation and a match was found. The tool also obtained four images and displayed the first three into the image displayers, as it has only three displayers due to space limits. The full path of the obtained image is displayed above the image displayer. From the images the investigator can recognise that the potential criminal created a document through inserting a bar code, a finger print, a human face and a page of a passport. The same face is also recognised in the passport image. This confirms counterfeiting of a document. A reset button is used to clear the displayed results, so as to reload another file for examination. In order to clarify on the probability calculator, another example is shown in figure 11.18 on a different document supplied for investigation.

Figure 11.18: Forensic evaluation of a suspect file 2

In figure 11.18, a smaller probability of 37% was calculated because the document had fewer images inserted, fewer artifacts recognised and there was no author under investigation compared to the previous example. A lower probability does not hinder an investigation but assists an investigator to focus on files with a higher probability of being counterfeited; in the end facilitating the accumulation of substantial evidence during forensic investigations.

The researcher also designed GIFT to be able to handle exceptions. These exceptions are explained in the following subsection.

11.6 Exceptions

GIFT will return error messages if a suspect file cannot be located or if the file contents cannot be displayed. The exceptions explained in this section include empty fields, file directory errors, a wrong extension, and no display errors.

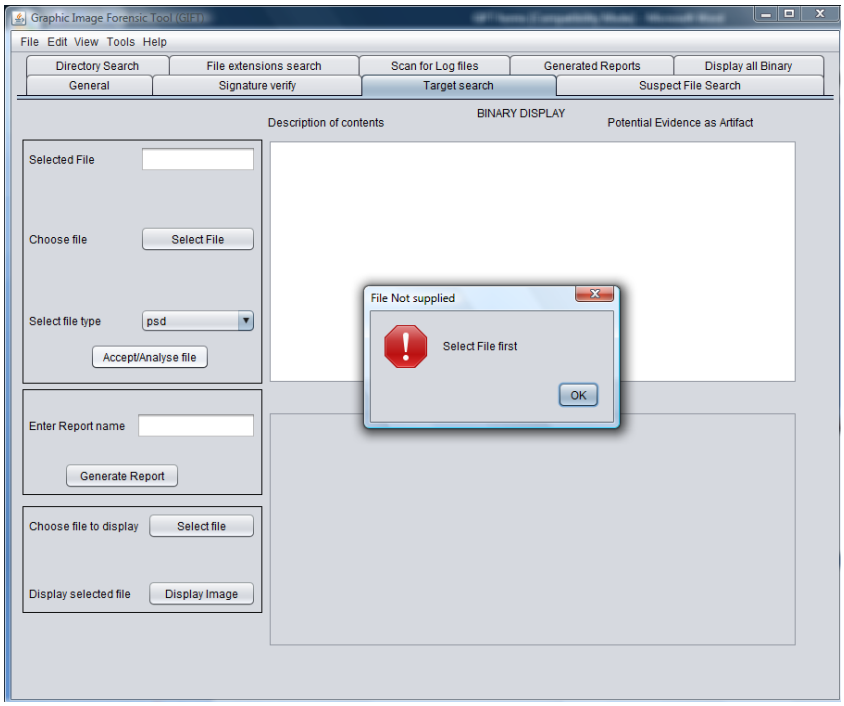

Figure 11.19: Empty fields' errors

If one executes an action command without specifying the file or field first, GIFT displays an error message, as shown in figure 11.19, to indicate which property or field needs to be supplied before the process can continue.

If a file is not found, the displayer returns the message, "Directory does not exist" or "This is not a directory", as shown in figure 11.20.

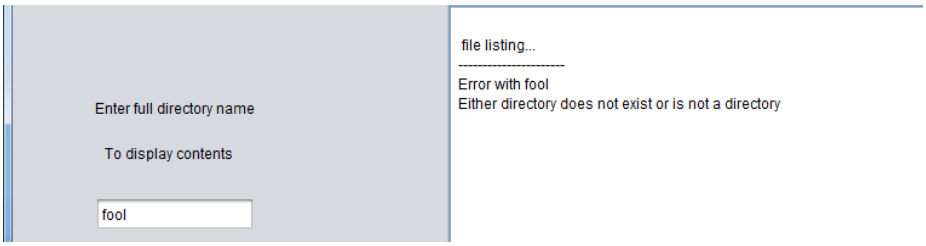

Figure 11.20: Directory errors

If GIFT does not recognise any of the information from a suspect file, this is either because a wrong extension has been selected or because the suspect file does not contain the digital forensic information. GIFT will then print a message on the binary display, as shown in figure 11.21.

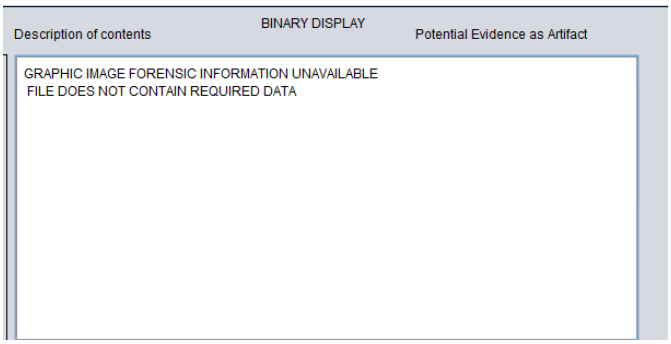

Figure 11.21: No information to be displayed

11.7 Conclusion

In this chapter, the researcher has explained how the functions of GIFT can be used in a digital forensic investigation, when pursuing an investigation on the basis of a suspect file. The researcher has omitted any demonstration of the other route, which is an investigation in the case and no suspect document has been given to the investigator, because such an investigation would be largely similar to the one described in this chapter but initiates at section 11.4.6 Scan for Log files . This chapter demonstrated how GIFT was able to determine that the suspect file was indeed a counterfeit document. It also showed how GIFT was able to provide essential supporting forensic evidence, and however it was able to identify names of the created documents. The same can be done to identify authors' names, time stamps and names of printed documents. It showed how GIFT was able to recognise the identities of the inserted objects by displaying the actual bar code. The same can be done for any image inserted for example an image of the human face that was inserted during the editing of a document. The image of the human face would then be visualised to verify if it matches the one contained in the suspect document. The researcher also undertook a full binary search to examine the printer spool files and the prefetch files. The researcher also examined the printer spool files to determine whether or not the suspect document was printed. Also the prefetch files were used to determine the time when the application concerned was last used. By doing this an investigator is able to determine whether or not the document was created in the timeframe between the installation of the application and the moment when it was last used.

A demonstration of a forensic evaluation function of the tool has been given. The function is an automated forensic acquisition of artifacts from a suspect document. The tool automatically

calculates the probability of a document being counterfeit and displays any images recognised to have been inserted in the document during possible counterfeiting.

The main function of GIFT is to extract the required digital forensic information from graphic design file types. The researcher has nevertheless designed additional functions to enhance and facilitate the examination process, such as browsing directories and searching for specific file extensions. The steps followed during the scenario investigation are based on requirements for a digital forensic examination that were strained in the last chapter. The researcher designed GIFT to investigate the file types that were selected as the basis for this research. It is nevertheless a simple procedure to enable GIFT to be used to function with other graphic design applications. An examination, for example, of the file type *idml* offers optimal results because this file type can be selected from the drop-down box that is incorporated in the tool design. Although it is necessary to examine unknown file types that are not listed in the drop-down box by means of a full binary search, no descriptions on the display are given. This renders minimal results. Since GIFT has also not been designed with a registry viewer, registry entries for installed application have to be obtained from *regedit* or other registry tools during any investigation. GIFT can nevertheless be further developed to handle additional file types, a registry viewer, and extra log files for both binary and graphic displays.

The following chapter contains an evaluation of the research that has been conducted, and a general conclusion.

Part V: Conclusion

CHAPTER TWELVE DISCUSSION AND CONCLUSION

This chapter, the conclusion, offers a summary of the dissertation. After the summary, the researcher immediately revisits the problem statement. After that there is a discussion that contains an evaluation of the study. Then there is a final conclusion, and some recommendations for future research in this field.

12.1 Summary

The researcher introduced the dissertation with an introductory chapter that explained the purpose of the research. This introductory chapter comprised the research objectives, the problem statement, the motives for the research, and the perceived limitations of the study. This was followed by the provision of a contextual background, with three chapters that reviewed the current literature pertaining to the topic of the study. The literature review consisted of three chapters, which successively reviewed the relevant developments and research that had taken place in digital forensics, counterfeit documents, and graphic design applications. In the chapter about digital forensics, the researcher offered definitions of digital forensics, processes, digital evidence, digital artifacts and techniques. In the chapter about counterfeit documents, the researcher defined counterfeiting and the various uses of counterfeit documents. In the chapter about graphic design applications, the researcher described and discussed the different types of graphic design applications in terms of functions and capabilities, for the applications selected for the purposes of this study.

Part three of the dissertation contained the original contributions that the researcher had made to this field of knowledge. This part was introduced with an evaluation of the experiments executed, and then presented a high-level overview of the model. This was immediately followed by three chapters in which the types of evidence assembled in a digital forensic investigation were illustrated. These three types of evidence are system-generated evidence, user-generated evidence, and timeline-associated evidence. After this, the researcher proposed a two-pronged counterfeiting digital forensic investigation process that could be adjusted either for an investigation with an electronic suspect document or for the investigation of a system that was prima facie suspected of being implicated in the production of counterfeited documents. The research explained how system-generated evidence can be ascertained from an examination of

the log file of an application's system. The researcher also explained how user-generated evidence can be obtained from the scrutiny of the eleven file types that are intentionally created by a user during the creation of counterfeited documents who uses a graphics application. In addition to this, it was explained how timeline-associated evidence can be identified in the registry of a system, and how prefetch files can be examined by an investigator in order to determine exactly when an application was installed on specific hardware and the precise moments when the application was used to create the various components of the fraudulent documentation.

Part four consists of two chapters. In the first of these chapters, the researcher explained the capabilities that a model would require before it could perform the task of conducting all aspects of a forensic digital investigation of suspect graphic design applications. This chapter about the indispensable requirements of a digital forensic model was followed by the chapter that introduced the researcher's prototype. In that chapter, the researcher introduced, explained, and discussed all aspects of the prototype developed for enabling investigators to conduct digital forensic investigations on the basis of graphic design applications. The researcher named this prototype a *graphic image forensic tool* (referred by means of the acronym: GIFT). After that, the researcher explained and demonstrated how GIFT was capable of searching for suspect files and extracting digital evidence from the contents of the files that had been identified as indicated in the counterfeiting investigation process.

12.2 Revisiting problem statement

Graphic design applications can be used to create counterfeit documents, moreover there are no dedicated current digital forensic tools that are specifically designed to examine a system that is suspected of having being implicated in the creation of counterfeit documents. In this study the researcher reviewed the graphic design applications that counterfeiters most frequently use, and explained how digital evidence of counterfeiting activities can be identified and assembled into a coherent narrative for the purpose of providing digital forensic evidence in subsequent court proceedings.

12.3 Discussion

A digital forensic investigator accumulates digital forensic artifacts because they form the basis of counterfeiting activities that have been perpetrated by using graphic design applications. The digital evidence that is gathered from log files enables an investigator to recognise all of the actions that a perpetrator has taken during the counterfeiting process. These actions are the scanning, editing, saving, and printing of counterfeit documents. By searching for and identifying all occurrences of these actions during the course of a digital forensic examination, an investigator is enabled to understand and explain the chain of events that a particular user executed to create a counterfeit document. If an investigator is unable to find any evidence of the four actions mentioned, during the course of a digital forensic investigation, then it becomes apparent that the system investigated could not have been implicated in the creation and production of counterfeit documents. There would, in other words, be no evidence of criminal wrongdoing, and it would be the responsibility of the investigator to explain this to the complainant or the relevant legal authorities. The researcher illustrated the various kinds of evidence that are gathered during the course of a forensic digital investigation by means of various tables, graphs and radar charts that showed, in a graphic way, how various kinds of evidence is distributed within log files. Such graphs and charts can be used during the investigation of a graphic design application to expedite the location and accumulation of digital evidence.

The researcher also explained how an investigator identifies and scrutinises, for content identification and content examination, the eleven kinds of file types that are intentionally created by a user who is engaging in counterfeiting activities. The researcher also detailed and defined the hexadecimal signatures for each of the file types that investigator examines in order to establish a file's true identity. The true identity of a file is a crucial component of a digital forensic investigation because it a straightforward matter for an offender to change the extension of a file in the hope of concealing his criminal activities from subsequent investigation. An investigator therefore scrutinises these signatures for the purposes of determining the true identity of a suspect file. The researcher also explained how the content of a file can be examined by means of a metadata analysis. Since each file type presents different types of evidence, such evidence can be recognised from an examination of the file identifiers, which are the tags and

prefixes that are attached to the evidence. These evidence identifiers are recognised and described per file type. Evidence identifiers are vital in an investigation because offenders can use various combinations of characters to name their documents with the purpose of concealing their criminal behaviour. It is therefore possible for an investigator to search for a particular identifier with the intention of identifying the name of a counterfeit document or the author of the document.

An investigator also supports the evidence accumulated by conducting a timeline analysis on the installation and prefetch files in the system. This kind of evidence indicates when the graphic design application was installed and when it was last run. By making use of the time stamps from the suspect documents and the actions that were executed during the counterfeiting process, an investigator is able to compile a timeline to illustrate the exact sequence of events that occurred during the counterfeiting process. A digital forensic investigator makes use of such timelines in order to illustrate the actual occurrence of events in the system because they form the basis of the evidence that indicates that perpetrators undertook various criminal activities at precise moments in the past.

Since an investigation into counterfeiting is initiated on the basis of varying degrees of suspicion or the existence of documents that are suspected of being counterfeited, the researcher proposed a two-pronged counterfeit investigation process to accommodate these variations in the circumstances that precipitated the investigation. The researcher therefore hypothesised that an investigation might be one in which an investigator had been presented with an electronic suspect document. In such a place it would be his task to determine whether the electronic document was actually a counterfeited document or not. We also hypothesised that there is another kind of investigation that is not based on the existence of a suspect document, but rather on reasonable prima facie suspicions that a particular system has been used to create counterfeit documents. The investigator's task would then be to determine whether or not the particular system investigated was indeed indicated in the creation and production of counterfeit documents or not, as the case may be. Such a two-pronged counterfeit investigation process can be used by an investigator to examine any graphic design application and gather evidence that indicates in a logical and coherent way that counterfeiting has taken place.

The researcher designed the tool named GIFT as a dedicated instrument for extracting digital forensic information from graphic design file types. The additional functions included in the design of GIFT are intended to facilitate the process of digital forensic investigation and to maximise its accuracy and relevance because it guides and investigator in a logical way through the various steps that are necessary for gathering evidence to prove that the suspect documents are indeed counterfeit or not. GIFT was specifically designed to function in an optimal way when it is utilised for investigating fraudulent activities that were perpetrated in conjunction with the file types that the researcher selected for the purpose of this study. GIFT can however also be used to investigate other graphic design applications, although with a minimal functionality and lower degrees of accuracy, as was explained in the chapter that describe the prototype. The researcher also pointed out that it would be a relatively simple matter to program GIFT to investigate the additional applications in which it is now not specifically programmed.

12.4 Final conclusion

The information assembled during the course of this research can facilitate investigations into the counterfeiting of documents for criminal purposes. The conclusions reached in this dissertation can be used by digital forensic examiners during their investigations into counterfeiting activities that are perpetrated by making use of graphic design applications. These conclusions can empower investigators to search for, find, and interpret evidence of criminal counterfeiting. The study also describes in detail the logical progression of any digital forensic investigation. It is absolutely necessary for a digital forensic investigator to move through of a digital forensic investigation in a logical and coherent way so that the evidence produced by the investigation will be able to secure an appropriate conviction in a court of law. As was noted, it is possible to extend an elaborate on the functionality of GIFT to accommodate any number of additional file types.

As the researcher noted in the motivation section of the introductory chapter, most related 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 has been tempered. The research carried in this dissertation can specifically be utilised to investigate the ways and means in which such images had been created as a technique to justify the forensics of tempered images.

Since this research was predicated on the creation of counterfeit documents on a Windows operating system, it must be noted that the main difference between a Windows operating system and other operating systems is reflected in the path locations for log files. From the research that undertaken, the researcher noted that the contents of log file and suspect files (i.e. the accumulated digital evidence) are similar, irrespective of the operating system that the perpetrators used.

Apart from conducting a counterfeiting investigation, the experimental results can be used for purposes of trouble shooting application faults, the monitoring of document designs, or application recovery.

The evidence that the researcher accumulated in this study was obtained from specific graphic design applications. It is nevertheless possible to carry out the same process by making use of other graphic design applications if one utilises the investigation process that the researcher proposed in chapter ten.

It should also be noted that an offender might make use of various other techniques and actions in an attempt to conceal his actions. Such techniques may range from the simple renaming of files, to file deletion and drive wiping as discussed in section 2.9.

12.5 Future work

The researcher has noted that the same research could be undertaken by making use of other graphic design applications such as Corel draw. Investigations that utilise other graphic design applications will also involve the log files and file types that are generated by those applications. The same research might also be replicated with the latest versions of graphic design applications. This will enable an investigator to be completely up-to-date and familiar with the latest innovations technology and software of this kind. A similar exercise could be carried out with other operating systems such as Linux and Mac OS. A profitable line of future research would be to extend the capabilities of the GIFT tool so that it includes more functions and has the ability to investigate new or additional file formats. The capabilities of GIFT could also profitably be extended with regard to the examination of registry files. A registry tool could be incorporated for the purpose of examining graphic design application installation time stamps. The capabilities of GIFT could also be extended to include object recognition, such as that of

OpenCV[82]. OpenCV could also be incorporated into the design of GIFT so that it has the capability of automatically identifying the status of suspect documents by scrutinising and identifying **specific** bar codes, digital signatures, images of human faces, and fingerprints.

Various academic papers were compiled to report on this research, three conference papers and a journal paper. The papers that were published included: Mabuto, E.K and Venter, H.S (2012d1, 2012d2, 2013d3 and 2013d4). These papers are included in Appendix D.

BIBLIOGRAPHY

[1] I.Rawoot, “Terrorists favour ‘easy’ fake SA passports”, *Mail and Guardian,* 17 June 2011

[2] J.Bargas, “Brazilian man attempted to open a bank account using a fake Jack Nicholson ID”, *International business times*,http://au.ibtimes.com/, 2 March, 2012

[3] N.Domanski, “ID crisis on the rise in Jackson Heights”, *Queens chronicle*, 1 March 2012

[4] J.Kell, "Adobe 2Q Net Up 54% On Broad Sales Gains, Higher Margins", *Dow Jones Newswires*, http://online.wsj.com, 21 June 2011

[5] The Digital Forensic Research Workshop (DFRWS), www.dfrws.org, Accessed 21 June 2011

[6] J.Wang, “Imageforensics based on manual blurred edge detection”,*Multimedia information networking and security(MINES)*, 2010, pp907-911.

[7] C.C.Lien, “Fast forgery detection with the intrinsic resampling properties”.,*Journal of information security. Vol 1 no1*, 2010, pp11-22

[8] Reuters, “Zuma needs to call for probe”, *News Time,* www.newstime.co.za, 6 July 2011, Accessed 20 July 2011,

[9] D.Jones, “Adobe 2Q Net Up 54% On Broad Sales Gains, Higher Margins”, *The Wall Street Journal*, 21 June 2011

[10] Digital Forensic Research Workshop, “*A roadmap for Digital Forensic Research*”,2001, pp 16.

[11] A.Reys and J.Wiles,*Cyber crime and digital forensics*,MA,Pedersen, 2007.pp 243-263

[12] E.Casey, *Digital Evidence and Computer Crime*, 2rd ed, London, Academic Press, 2004, pp10-65

[13] National Institute of Justice,*Electronic Crime Scene Investigation. A Guide for First Responders*,http://www.ncjrs.org/pdffiles1/nij/187736.pdf, July 2001

[14] E.Casey, *Handbook of Digital Forensics an Investigation*, Academic Press. 2009. p 567

[15] M.Solomon, G.Barett, and D.Broom, *Computer_Forensics*, London, Sybex, 2005, pp 51

[16] M. Van Horenbeeck, *Technology Crime Investigation* 24, http://web.archive.org/web/20080517022757/, Accessed 12 February 2011

[17] S.Koranne , *Handbook of open source tools*, Springer, London 2011, pp175

[18] M. Reith, C. Carr and G. Gunsch, "An examination of digital forensic models", *International Journal of Digital Evidence*, 2002,

[19] Ó.Séamus, and Cuardhuáin, "An Extended Model of Cybercrime Investigations", "*International Journal of Digital Evidence*", Volume 3, Issue1.2004

[20] A.Jones. C.Valli, *Building a digital forensic laboratory*, Burlington, Elsevier, 2008, pp 285

[21] F.Cohan, "Towards a science of digital forensic investigation", *IFIP Advances Digital Forensics VI*, China, 2010, pp 17-35

[22] T.J.Gardner, T.M.Anderson, *Criminal evidence principles and cases*,CA, USA, Wardsworth, 2007, pp37

[23] J. Grama, *Legal issues in information security*,MA, USA, ,Jones and Bartlett, 2011, pp465

[24] J. Ingram, "Criminal evidence", 11th ed, *John C Klotter Justice Administration Legal Series*,USA, Elseveir, 2012, pp846

[25] C.Altheide, H.Carvey. "*Digital Forensics with Open Source tools*", MA USA, Elsevier, 2011, pp 2

[26] M.V.Zelkowitz, "*Advances in computers; information security*", Elsevier, 2009

[27] A.S. Hornby, *"Oxford Advanced Learners Dictionary. 7th ed"*, Oxford University Press, 2006

[28] Microsoft, "Windows registry information for advanced users*"*, *Microsoft Knowledge Base article "256986"*, http://support.microsoft.com/kb/256986, Accessed 26 September 2012

[29] K.J.Jones, R. Bejtlich and C.W. Rose. *Real Digital Forensics*, NY, Addison –Wesley, 2009, pp

[30] G.Giuseppini, Mark Burnett, *Microsoft Log parser*, Syngress, MA. USA, 2005,pp50

[31] Winhex, *hexeditor*, www.x-ways.net/forensics, Accessed 13 June 2011

[32] HxD, *hexeditor*, http://mh-nexus.de/en/hxd/ , Accessed 13 June 2011

[33] HHD, *hexeditor*, http://www.hhdsoftware.com/hex-editor , Accessed 13 June 2011

[34] Hexedit, *hexeditor*, http://www.hexedit.com/, Accessed 13 June 2011

[35] Steg detect, http://freecode.com/projects/stegdetect, Accessed 13 June 2011

[36] Steg break, J. Vacca ,K Rodolph, *System forensics investigation and respons"* , Jones and Barnett, 2011 USA, pp167

[37] A.J.Marcella, and D.Menendez, *Cyber forensics and a manual*, 2rd ed, London, Auerbach Publications, 2008

[38] M.Cross, *Scene of the Cyber Crime*, 2rd ed, Syngress. 2008.pp 500

[39] Ophcrack, http://ophcrack.sourceforge.net/, Accessed17 September 2012

[40] K.Mandia, C.Prosise, and M.Pepe, *Incident_Response_and_Computer_Forensics*, 2nd ed NY, McGraw-Hill. 2003.p 72

[41] Prodiscover dft, http://www.techpathways.com/prodiscoverdft.htm, Accessed 4 May 2011

[42] Windows Forensic Toolchest (WFT), http://www.foolmoon.net/security/wft/index.html, Accessed 4 May 2011

[43] FTK, www.accessdata.com, Accessed 4 May 2011

[44] Encase, www.guidancesoftware.com, Accessed 4 May 2011

[45] Paraben, http://www.paraben.com Accessed 4 May 2011

[46] Hotpepper, http://www.hotpepperinc.com Accessed 4 May 2011

[47] Datalifter, http://www.datalifter.com Accessed 4 May 2011

[48] Digital Intelligence, http://www.digitalintelligence.com Accessed 4 May 2011

[49] E.Conne, *3d for graphic designers*, Wiley, USA, 2011

[50] Adobe, *graphic design application*, www.adobe.com, Accessed 26 September 2011

[51] CorelDraw,*Graphic design application*, www.corel.com, Accessed 26 September 2011

[52] AutoCAD,*graphic design application*, www.cadco.co.za, Accessed 26 September 2011

[53] Primo,*graphic design application*, www.primosoftware.com/, Accessed 26 September 2011

[54] Free DWG, *graphic design application*, www.freedwg.eu, Accessed 26 September 2011

[55] Sweet Home 3d, *graphic design application*, www.sweethome3d.com/, Accessed 26 September 2011

[56] Google SketchUP ,*graphic design application*, ketchup.google.com/, Accessed 26 September 2011

[57] Ulead,*graphic design application*, www.corel.com/corel/jump/us/en/10700008/, Accessed 26 September 2011

[58] Edraw Max, *graphic design application* www.edrawsoft.com/EDrawMax.php, Accessed 26 September 2011

[59] DAZ Studio, *graphic design application*, www.daz3d.com, Accessed 26 September 2011

[60] ChemDraw Ultra,*graphic design application*, www.cambridgesoft.com/, Accessed 26 September 2011

[61] Photo to Cartoon, *graphic design application*, www.cartoon.pho.to/, Accessed 26 September 2011

[62] Easy Flyer , *graphic design application*, www.easyflyer.fr, Accessed 26 September 2011

[63] PCB artist , *graphic design application*, www.4pcb.com/free-pcb-layout-software Accessed 26 September 2011

[64] Sothink, *graphic design application*, www.sothink.com/, Accessed 26 September 2011

[65] J.Ward and R.Bostick, "Adobe's Sales Exceed Estimates on Rebound From Japan Quake", *Business Chronicle*, 21 June 2011

[66] Software reviews, Photo editing software reviews, www.reviews.com/photo-editing-software/, Accessed 20February 2013

[67] National Security Agency (NSAC), "Hidden data and metadata in adobe pdf files", *Enterprise applications division*,27 July 2008, http://www.nsa.gov/. Accessed 26 September 2012

[68] Sofxpansion, *Metadata in office and pdf documents*, http://www.soft-xpansion.com/,Accessed 26 September 2012

[69] Photoshop features, http://www.adobe.com/products/photoshop/features.html, Accessed 26 September 2011

[70] In-design features, http://www.adobe.com/products/indesign/features.html, Accessed 26 September 2011

[71] Illustrator features,http://www.adobe.com/products/illustrator/features.html, Accessed 26 September 2011

[72] C. L. Lindsay, *The College Student's Guide to the Law*, USA, Taylor trade publishing, 2005, pp221-226.

[73] R.Cavanagh, *Operating systems market report*,.http://chitika.com/, 17 October 2011

[74] M.Long "Windows 7, Office Drive Record Microsoft Revenue", *Top Tech News*, Accessed 23 March 2013

[75] G.Kesler, File signatures, http://www.garykessler.net/library/file_sigs.html, Accessed 10 October 2011

[76] M. Reddy,Graphic design file format database, http://www.martinreddy.net/gfx/2d-hi.html Accessed 10 October 2011

[77] Metadata Extraction Tool, www.extractmetadata.com, Accessed 11 July 2012

[78] H.Carvey, *Windows Forensic Analysis Dvd Toolkit*, 2nd Ed, Elsevier, 2009, pp 296

[79] Adobe XMP. http://www.adobe.com/ products/ xmp/index.html, Accessed 11 November 2011

[80] Adobe bridge, www.adobe.com/products/bridge, Accessed 11 November 2011

[81] ISO/IEC 27043 Information Technology, Security techniques, Incident investigation processes and principles, Committee draft standard,2012

[82] Open Source Computer Vision, (OpenCV), www.opencv.org, Accessed 11 March 2013.

[83] J.R.Vacca, "Computer Forensics Computer crime scene investigation", 2nd ed, MA, Charles River Media, 2005, pp65

[84] E.Council. "Investigation procedures and Response". Cengage learning.NY.2010.pp53

[85] C.King, T.Vidas, "Empirical analysis of solid state disks, data retention when used with contemporary operating systems", *Digital investigations 8*. Elsevier, 2001, pp111-117

[86] J.Dean, S.Ghemawat. "Map Reduce: Simplified Data Processing on Large Clusters". *6th Symposium on Operating Systems Design and Implementation USENIX Association*, pp137-149

[87] B. P.Battula, B.Rani, R.Prasad & T .Sudha, "Techniques in Computer Forensics: A Recovery Perspective", *International Journal of Security (IJS), Volume (3) : Issue (2)*, pp27-35

[88] J.Luck, M.Stokes. "An Integrated Approach to Recovering Deleted. Files from NAND Flash Data". *Small Scale Digital Device Forensics Journal, Vol. 2, No. 1*, June 2008

[89] B.Dolan-Gavitt. "Forensic analysis of the Windows registry in memory". *Digital investigation 5 (2008*) S26 – S32. USA, Elsevier.

[90] H.Farid. "Image forgery detection", *IEEE Signal Processing Magazine* March 2009, pp 16-25

[91] N.Singh, B.Singh R.S.Bhati. "Raw Digital Image Steganalysis For Computer Forensic Investigation", *Computer Science & Information Technology (CS & IT)*, pp 161-168, 2012

[92] M.Kirchner, R.Bohme, "Tamper Hiding: Defeating Image Forensics", *9th International conference on Information Hiding*. pp 326-341, 2007.

[93] K.Cohen, "Digital Still Camera Forensics", *Small Scale Digital Device Forensics Journal*, Vol. 1, No. 1, June 2007, pp 2-8

[94] E.Acebo, M.Sbert. "Benfords law for natural and synthetic images", *Computational Aesthetic in Graphics, Visualisation and Imaging*, 2005

[95] C.C.Lien, "Fast forgery detection with the intrinsic resampling properties"., *Journal of information security. Vol 1 no1*, 2010, pp11-22

[96] M.C.Stamm. "Forensic detection of image tampering using intrinsic statistical fingerprints in histograms". *APSIPA Annual summit and conference*, Japan, 2009, pp 563-572

[97] K.Cohen. "Digital Still camera Forensics", *Small scale digital device forensics journal*, vol1, no 1, june 2007. Pp 2-8

[98] H.Farid. "Image forgery detection", *IEEE signal processing magazine*, 2009, pp 16-25

[99] S. Bayram, I. Avcibas, B. Sankur, and N. Memon, "Image manipulation detection", *J. Electron. Imaging*, vol. 15, no. 4, p. 41102, 2006.

[100] J.Wang, "Image forensics based on manual blurred edge detection", *Multimedia information networking and security (MINES)*, 2010, pp907-911.

[101] N.Memon. "Photo Forensics". *International workshop on information security*, NYU, 2012.pp1-27

[102] M.Fouche. M.Olivier, "Detecting non stereoscopic to stereoscopic image splicing with the use of disparity maps", *South African institute of computer scientists and information technologies, SAICSIT*, 2012, pp271-274

[103] R.Harris, "Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem", *Digital investigation 3S*, 2006 p44 –49.

[104] T.D.Morgan "Recovering deleted data from the windows registry", *Digital investigations*, 2008, pp33-41.

[105] P.Gutmanm, "Secure Deletion of Data from Magnetic and Solid-State Memory". *Sixth USENIX Security Symposium*, San Jose, CA, http://www.cs.auckland.ac.nz/~pgut001/pubs/secure_del.html. Accessed 15 March 2013. pp. 77–90.

[106] C.Wright, D.Kleiman, S.Sundhar, Overwriting hard drive data: the great wiping controversy.*ICISS 2008*. http://portal.acm.org/citation.cfm?id=1496285. pp 245-257. Accessed 15 March 2013.

[107] Carrier B. Spafford E., "Getting Physical with the Digital Investigation Process", *International Journal of Digital Evidence*, Vol.2, 2, 2002

[108] R. Leigland, A. Krings, "A Formalisation of Digital Forensics", *International Journal of Digital Evidence, 2004,Volume 3, Issue 2.*

[109] Ó. Séamus, Cuardhuáin, , "An Extended Model of Cybercrime Investigations", *International Journal of Digital Evidence,* Volume 3, Issue 1. 2004.

[110] M. Reith, C. Carr, G. Gunsch, "An examination of digital forensic models", *International Journal of Digital Evidence*.2002

[111] S. Garfinkel, P. Farrella, V. Roussev, G. Dinolt, , "Bringing science to digital forensics with standardised forensic corpora", *Digital Investigation 6* S2-S11, 2009.

[112] L.Basley, "Some students scalp basketball tickets", Diamond BackOnline, http://www.diamondbackonline.com/news/campus/article_1a48583a-8169-11e2-a127-0019bb30f31a.html, 28 February 2013

[113] J.Taranto, "The fallows principle", Market Watch, http://stream.marketwatch.com/story/markets/SS-4-4/SS-4-23674/, Accessed 15 March 2013

[114] T.Mcnichol, "Computers alter the science of document authentication", NewYork Times, http://www.nytimes.com/2004/09/23/technology/circuits/23docs.html?_r=0, September 2004, Accessed 15 March 2013.

[115] CCleaner, Piriform Ltd, http://www.piriform.com/ccleaner, Accessed 10 March 2013.

APPENDIX

Appendix A: Tables pertaining to System-generated evidence

Table A1 sets out the address offsets for scanned documents in the Adobe Photoshop CS4 log file. The column headings in Table A1 are briefly explained in what follows for purposes of clarity. The **number of scanned documents** refers to the number of documents that were scanned prior to the researcher's examination of the log file. The **size of the log file** represents the size of the log file size at the time of the researcher's examination. The **address offset** for a digital artifact entry represents the address pointer in hexadecimal format with the entry showing the path of the scanned documents. The **number of entries** indicates the number of entries that were recorded throughout the log file for that particular digital artifact. Thus, for example, if five documents were scanned, there will only be one digital artifact that indicates such an entry (whereas one might logically have expected five separate entries). The column headed **Position** shows the relative position that is calculated from the following equation (equation 7.1):

Address offset (in decimal format)/ Size of file = X (the calculated relative position)

Example from the second line (in table 7.1) 1ECE4 for a 199kb file size

1ECE4 from hex to dec = 126180

126 180kb / 199 999kb = 0.634

This calculation was made to determine the position of the digital artifact relative to the size of the whole log file. The *rationalised* position is the position of the entry relative to the size of the log file out of 10. This is for comparative purposes only since it is known that the address offset is variable. The results therefore indicate that for a number of the scanned documents, the entry's location has been maintained at a place that is about six tenths within the file. As the size of the log file increases, the entry's location begins to shift to another position, even though it is still in the same address range (at a place about six tenths within the file). This tells the investigator to look for this evidence at a place that is about six tenths (or three fifths) down the file. It may also be noted that if the location for saving scanned documents is changed, the most recent location will replace the earlier one in the log file. Note that there are several line entries in the Table A1. Each line entry represents the digital artifacts that were discovered from a specific number of

documents that were scanned. The information is set out in this way so that the researcher can monitor the changes that were effected in the digital artifacts as more documents were scanned.

A1

Number of Documents(CS4Prefs)	Size of log file (KB)	Data distribution	Number of entries	Position	Rationalised position(X/10)
1	198	1ECE4	1	0.637	6
2	199	1ECE4	1	0.634	6
5	199	1ECE4	1	0.634	6
10	200	1ECE4	1	0.634	6
15	206	21202	1	0.639	6
20	207	2022C	1	0.639	6

Table A1: Address offsets for scanned documents in Adobe Photoshop CS4

The first line, for example, represents digital artifacts from the log file after one document had been scanned; the second line represents the digital artifacts from the log file after two documents had been scanned, and so on.

A2

Number of Documents(CS3)	Size of log file (KB)	Data distribution (offset)	Number of Entries	Position	Rationalised position(X/10)
1	157	165A0	1	0.583	6
2	158	166FA, 167F0	2	0.5816	6
5	159	166F8, 167EC	5	0.5779	6
10	160	166F2,	10	0.574	6
20	161	166EA	20	0.571	6
30	162	1677A	22	0.568	6
40	162	166EC	22	0.567	6
50	162	167DA	22	0.569	6

Table A2: Address offsets for insertions in Adobe Photoshop CS3

A3

Number of Documents (CS5)	Size of log file (KB)	Data distribution (offset)	Number of entries	Position	Rationalised position (X/10)
1	191	4C97(f), 14672	2	0.103(f) 0.438	4
2	192	4C97(f) 14670	2	0.102(f) 0.435	4
5	193	4C97(f) 14698,	2	0.102(f) 0.437	4
10	193	1472A	2	0.434	4
15	194	14644	2	0.431	4
20	194	14602	2	0.430	4
30	194	1464C	2	0.431	4
40	194	146B6	2	0.431	4
50	194	14646	2	0.431	4

Table A3: Address offsets for insertions in Adobe Photoshop CS5

Table A3 shows the address offsets for version CS5. In the tables contained in this chapter, the ”f” in brackets, for example in columns 3 to 5 in Table 7.3 indicates that the entry only shows a folder location and not a full name. If the number is not bracketed with “f”, it means the full name of a created document is shown, and not a path.The entries for digital artifacts from Adobe Photoshop version CS5 can be seen to be located at about four tenths of the file size. These entries are maintained within the same range for any number of documents that have been created. If one compares this version to other versions, it will be seen that the latest version CS5 log file is of a relatively small file size, and that it will increase slightly in size and maintain the same size – irrespective of the number of documents that will be created.

Figure A4 sets out a comparison of the three versions of Adobe Photoshop. The line graph shows that version CS5 records digital forensic artifacts in its log file differently from the way they appear in the earlier versions CS3 and CS4 (they appear earlier – at the beginning of the file).

A4

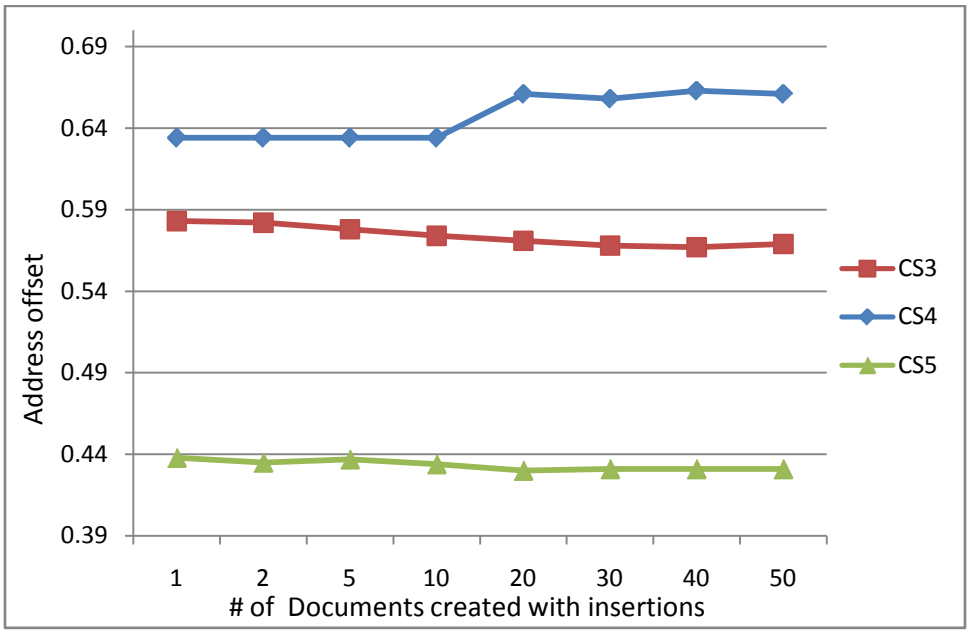

Figure A4: Comparing address offsets for insertions in Photoshop versions CS3, CS4 and CS5

A5

# of documents (CS3)	Size of log file (MB)	Data distribution (offset)	Number of Entries	Position	Rationalised position(X/10)
1	9.82	89916B(f)	2	0.918	2 and 9
2	9.82	18716B(f) 89916B(f)	2	0.163, 0.918	2 and 9
5	9.82	18716B(f) 82516B	2	0.163 0.870	2 and 9
10	9.82	18416B(f) 82516B(f)	2	0.163, 0.870	2 and 9
20	9.82	18416B(f) 82816B(f)	2	0.163, 0.870	2 and 9
30	9.82	18416B(f) 82816B(f)	2	0.163, 0.870	2 and 9
40	9.82	18416B(f) 83816B(f)	2	0.163, 0.870	2 and 9
50	9.82	18416B(f) 81A16B(f)	2	0.163, 0.870	2 and 9

Table A5: Address offsets for insertions in Adobe In-Design CS3

Figure A6 shows that, in the three versions of the software, the digital artifacts are located at the beginning and towards the end of its file size. This entry is similar and exactly the same respectively, but only occurs in a different location. The graph shows that Adobe In-Design creates a more uniform distribution for its different versions.

A6

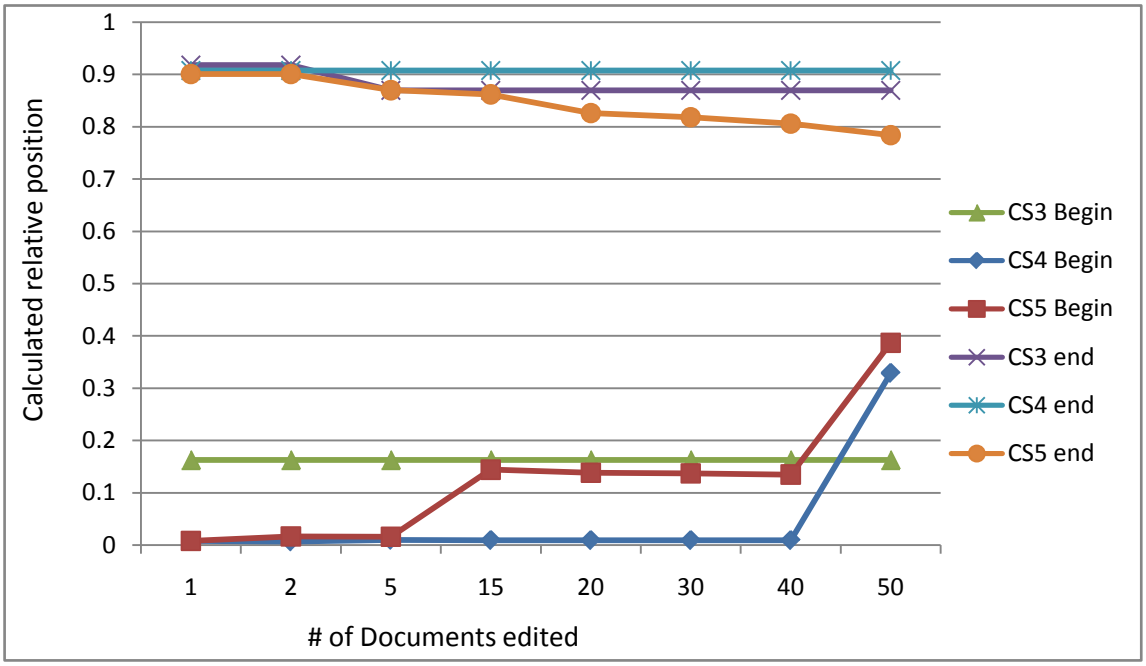

Figure A6: Comparing address offsets for insertions in In-Design in CS3, CS4 and CS5 respectively

A7

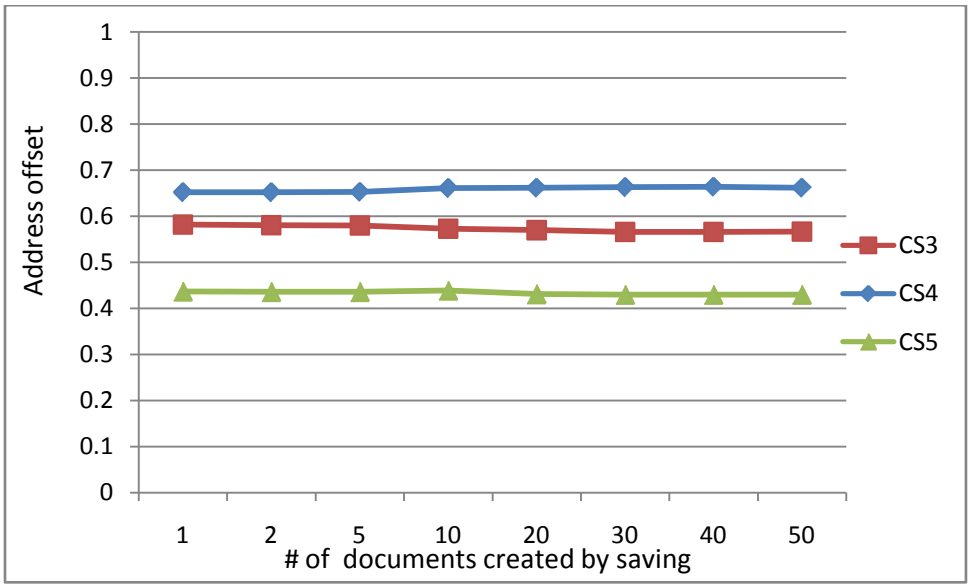

Figure A7: Comparing the address offsets for saved documents in Photoshop CS3, CS4 and CS5

A8

Number of Documents (CS4)	Size of log file (KB)	Data distribution (offset)	Number of entries	Position	Rationalised position (X/10)
1	198	4A19,1F88A	2	0.099, 0.652	1 and 6
2	199	4A19(60b),1FA66, 1FAC2(2)	3	0.095,0.651, 0.652	1 and 6
5	199	4A19,1FA66- 1FC28(5),	6	0.095, 0.651	1 and 6
10	200	4A19, 1FA66 (10)	11	0.095, 0.648	1 and 6
15	206	4A19,20FAE(5), 213FC(10)	16	0.092,0.656, 0.661,	1 and 6
20	207	4A19,20FAE(5), 212BA(5),21708(10)	21	0.092,0.653, 0.656, 0.662	1 and 6
30	209	4A19,20FAE(10), 2166C(5),21978(5), 21DC6(3)	23	0.091,0.646, 0.655,0.658, 0.663	1 and 7
40	209	4A19,20FAE(10), 21578(10), 21E3A(1)	23	0.091,0.646, 0.653, 0.664	1 and 7
50	208	4A19,20FAE(3), 211B2(1), 21320(1), 2141C(1), 21510 (10), 219E2(6)	23	0.091,0.649, 0.655,0.653, 0.654, 0.662	1 and 7

Table A8: Address offsets for saved documents in Adobe Photoshop CS4

In the following table, the number in brackets in column 3 of Table A7 represents the number of entries that exist for saved documents at that particular location.

A9

Number of Documents	Size of log file (MB)	Data distribution (offset)	Number of entries	Position	Rationalised position (X/10)
1	9.82	899196(f), 8BE867(1) 9921CB(1)	1	0.918(F) 0.9337	9
2	9.82	188867(1), 188D87(2) 25C1CB(1),25DEEB(1)	2	0.1637(F) 0.1638(2)	2
5	9.82	188867(1), 188D87(2) 83472E(1) ,837D87(5)	5	0.1637(F) 0.1638(2) 0.8775(5)	2
10	9.82	18A79E(1), 18AC56(1) 191C43(10), 834953(1) 837D87(5), 9921CB(1)	10	0.1676	2
20	9.82	191C43 (10) 837C83(10) 83A5BD(1)	20	0.1675 0.8774	2 and 8
30	9.82	18AB1F(13) 837C43(10) 87C41C(1) 87C5C6(1)	30	0.8774 0.1646	2 and 8
40	9.82	18AB1F(17) 26AB4D(1) 861B2F(10) 9921CE(1)	40	0.1646 0.8949	2 and 8
50	9.82	191A17(1) 1D4B2F(10) 350EB7(3) 81BB2F(10) 9921CB(1)	50	0.1954 0.8657	2 and 9

Table A9: Address offsets for saved documents in Adobe In-Design CS3

Adobe In-Design CS5 records entries for saved entries automatically throughout the log file, as it did in previous versions. However, this version records saved entries at about the middle of the file in comparison to what it did in the earlier version, where it recorded at the end of the log file. This is illustrated in the line graph, Figure A10.

A10

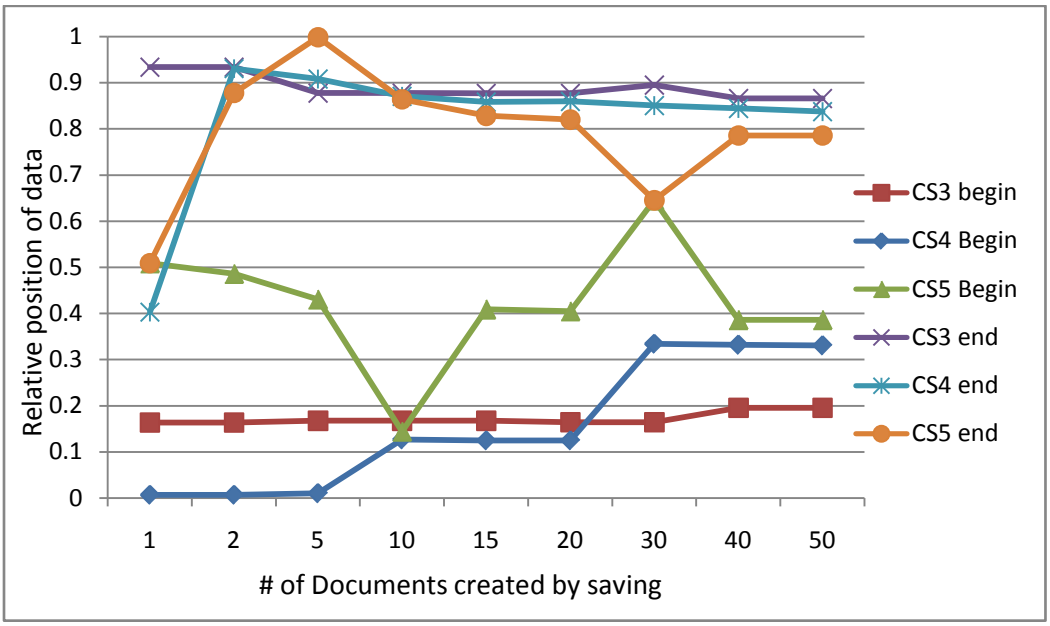

Figure A10: Comparing the address offsets for documents created in In-Design versions CS3, CS4 and CS5

A11

Number of Documents(CS4Prefs)	Size of log file (KB)	Data distribution	Number of entries	Position	Rationalised position(X/10)
1	155	158D8	1	0.569	6
2	155	15A2C	1	0.572	6
5	155	15A2C	1	0.572	6
10	155	15A2C	1	0.572	6
15	155	15A2C	1	0.572	6
20	155	15A2C	1	0.572	6

Table A11: Address offsets for scanned documents in Adobe Photoshop CS3

A12

Number of Documents(CS4Prefs)	Size of log file (KB)	Data distribution	Number of entries	Position	Rationalised position(X/10)
1	191	15540	1	0.457	5
2	192	15558	1	0.455	5
5	192	155D8	1	0.456	5
10	192	15684	1	0.457	5
15	192	156EC	1	0.457	5
20	192	157C0	1	0.458	5

Table A12: Address offsets for scanned documents in Adobe Photoshop CS5

A13

Number of Documents(CS4)	Size of log file (KB)	Data distribution (offset)	Number of entries	Position	Rationalised position(X/10)
1	198	1ECE4	1	0.634	6
2	199	1ECE4	1	0.634	6
5	199	1ECE4	1	0.634	6
10	200	1ECE4	1	0.634	6
15	206	21270	2	0.659	7
20	207	21242, 2150E	2	0.656, 0.659, 0.661	7
30	209	2157C,21900, 21BCC,	3	0.653, 0.658	7
40	209	21500,21DC2,	2	0.653, 0.663	7
50	208	21134, 21124, 2139E, 21492	5	0.651,0.651, 0.654, 0.655	7

Table A13: Address offsets for insertions in Adobe Photoshop CS4

A14

Number of Documents(CS4)	Size of log file (MB)	Data distribution (offset)	Number of entries	Position	Rationalised position(X/10)
1	7.36	D03B	1	0.0076	1
2	8.13	D03B	1	0.0066	1
5	8.32	13827, 733827,	2	0.0096, 0.908	1 and 9
10	8.69	13827	1	0.0092	1
15	8.80	13827	1	0.0091	1
20	8.81	13827	1	0.0091	1
50	9.03	2D5827	1	0.3290	3

Table A14: Address offsets for insertions in Adobe In-Design CS4

A15

Number of Documents (CS5)	Size of log file (MB)	Data distribution (offset)	Number of entries	Position	Rationalised position(X/10)
1	6.39	D033 (f)	1	0.0083	1
5	7.33	1E263(F), 64C263(f),	2	0.0168, 0.9008	1 and 9
10	7.59	10E263 (F), 64C263 (f),	2	0.0163, 0.8700	1 and 9
15	7.66	10E263(f) 64C263(f)	2	0.1444 0.862	1 and 9
20	7.99	10E263(f) 64C263(f)	2	0.1384 0.8264	1 and 9
30	8.07	10E263(f) 64C263(f)	2	0.1371 0.8182	1 and 9
40	8.19	10E263(f) 64C263(f)	2	0.1351 0.8062	1 and 9
50	8.42	31B263(f) 64B263(f)	2	0.3868 0.7842	4 and 8

Table A15: Address offsets for insertions in Adobe In-Design CS5

A16

Number of Documents(CS3)	Size of log file (KB)	Data distribution (offset)	Number of entries	Position	Rationalised position(X/10)
1	157	3AD3 (f), 16522	1	0.096 0.582	1 and 6
2	158	3AD3 (f), 16676, 16772	5	0.095 0.581	1 and 6
5	159	3AD3 (f), 16676, 16770, 16864	5	0.094 0.580	1 and 6
10	160	3AD3 (f), 16676,	10	0.094 0.573	1 and 6
20	161	3AD3 (f), 16676	20	0.094 0.570	1 and 6
30	162	3AD3 (f), 16676	22	0.093 0.566	1 and 6
40	162	3AD3 (f), 16676	22	0.093 0.566	1 and 6
50	162	3AD3 (f), 166DE	22	0.093 0.567	1 and 6

Table A16: Address offsets for saved documents in Adobe Photoshop CS3

A17

Number of Documents (CS5)	Size of log file (KB)	Data distribution (offset)	Number of entries	Position	Rationalised position(X/10)
1	191	4cd3, 145f6(14672)	1	0.103, 0.437,	1
2	192	4c97, 14602(14670), 146e3	2	0.102, 0.434, 0.436	1 and 4
5	193	4c97, 1460e(14698),14710(14790), 14808(14884), 148fc(14978)	5	0.102, 0.433, 0.435, 0.436	1 and 4
10	193	4c97, 1460e(2)(1472A), 147a2, (148a6), 1491e,(14a2e), 14aa6(14b26), 14b9e(14c1a),	10	0.102, 0.432, 0.437, 0.439, 0.439	1 and 4
15	194	4c97(f), 146FE	15	0.101 0.431	1 and 4
20	194	4c97(f), 145CA	20	0.101 0.430	1 and 4
30	194	4c97(f), 145CA	22	0.101 0.430	1 and 4
40	194	4c97(f), 145CA	23	0.101 0.430	1 and 4
50	194	4c97(f), 145CA	23	0.101 0.430	1 and 4

Table A17: Address offsets for saved documents in Adobe Photoshop CS5

A18

Number of Documents	Size of log file (MB)	Data distribution (offset)	Number of entries	Position	Rationalised position (X/10)
1	7.36	D05E(f), 2d4227(1)	2	0.0072(f), 0.403	1 and 4
2	8.13	D05E (f), 2d4227(1), 7367f7(f), 737153(2), 738d7b(1)	6	0.0065(f), 0.9302(f), 0.9305(2)	1 and9(f), 9
5	8.32	13827(f), 733827(f), 14bcb (4),14dco(5), 6f1ec8(1), 734bcb(4), 734dc0(4),795d08(1),795 d78(1), 7fc753(1),	23	0.00960, 0.9075(f), 0.0102(4), 0.0103(5), 0.9081(4),	1 and 9(f), 1 and 9
10	8.69	13827(f), 733827(f), 14d5c(1), 10daf7(9), 10dddf(7), 2d4da0(1), 6b1d00(1), 6b1e5c(1), 6f41ef(1), 736c0c(16)	39	0.0015(f), 0.8689(f), 0.0098(1), 0.1271(9), 0.8077(1), 0.8705(16)	1 and 9(f), 1 and 9
15	8.80	13827(f), 733827(f), 10cd57(1), 2d2c0c(16), 6b1d48(1), 6b2bd0(1), 6b2d34(1), 6b2e5c(1), 736c0c(16), 82ad9b(1)	47	0.0090(f), 0.858(f), 0.1251(1), 0.336(16), 0.7981(1), 0.8595(16)	1 and 9(f), 1,3,and 9
20	8.81	13827(f), 10cd57(1), 2d2c0c(16), 6b1d48(1), 6B2BD0(1),6B2D34(1) 734827(f),738AF7(10), 738E28(6),	53	0.0090(f), 0.8575(f), 0.1249(1),0.336(16), 0.859(10), 0.8595(6)	1 and 9(f), 1, 3 and 9
30	8.90	10C827(f), 2D2D6B(1), 2D2ECD(1),2D5AF7(10) , 2D5E26(6), 6B5D35(1), 6B5E97(1), 734827(f), 737AF7(10), 737E26(6),	64	0.1236(f), 0.334(10), 0.3341(6), 0.8498(f), 0.851(10), 0.8504(6)	1 and 9(f), 1 and 9
40	8.96	10C827(ff), 2D2D6B(1), 2D2ECD(1),2D5AF7(10) , 2D5E26(6), 733827(f), 734D6B(1), 736C12(14),	75	0.1227(f), 0.331(10), 0.3318(6), 0.8427(f), 0.8442(14)	1 and 8(f), 1and 8
50	9.03	2D5827(f), 2D6D68(1), 2D8AF7(10), 2D8E74(4), 6B7D7A(1), 6B8E62(2), 732827(f), 735AF7(10), 735E69(4),	85	0.329(f), 0.330(10), 0.835(f), 0.8372(10)	3 and 8(f), 3 and 8(10)

Table A18 Address offsets for saved documents in Adobe In-Design CS4

A19

Number of Documents	Size of log file (MB)	Data distribution (offset)	Number of entries	Position	Rationalised position(X/10)
1	6.39	D033 (f), 31B0D7(1)	1	0.0083(f), 0.509(1)	1and 4
5	7.33	1E296 (f), 10DD77(4), 3651F2(1), 365420(1), 36564B(1), 6587A(1), 626B13(1), 64C290 f), 64DD77(4),713647(1)	5	0.0168(f), 0.90087(f) 0.1507(1)0.877864)	1and 4
10	7.59	10E296 (f),64C290(f), 31DC57(14),3651F2(1), 365420(1),36564B(1), 365877(1),6268E7(1), 626B13(1),64FC57(14),7060BB(1),7131F0(1), 72A513(1),73A1F1(1),73A418(1)73A63E(1),	10	0.1457(f), 0.87001f), 0.430(14), 0.871(14), 0.998(1)	1and 4
15	7.66	10E28C(f),31DC57(15) 64FC57(15),72A0BC(1) 74383D(1)	15	0.1444(f) 0.4266 (15) 0.8639	1and 4
20	7.99	10E28C(f) 31DC57(15) 64FC57(15) 7538E4(1)	20	0.1385(f) 0.4089(15) 0.8283	1and 4
30	8.07	10E290(f) 31DC57(15) 64FC57(15) 754770(1)	30	0.1371(f) 0.4049(15) 0.8200	1and 4
40	8.19	10E290(f) 3267671(15) 64EC57(15) 7C30B9(1) 7D94ED(1)	40	0.1351(f) 0.6453(15)	1 and 6
50	8.42	31B299(f) 31EC57(15) 64EC57(15) 76A3B4(1)	50	0.386(f) 0.3885(15) 0.7855(15)	1 and 4

Table A19: Address offsets for saved documents in Adobe In-Design CS5

Appendix B: Tables pertaining to User-generated evidence

B1

Size of file(kb)	Version of App	Number of images inserted	Offset for time stamps	Modification time	Metadata time	Creator tool
324	CS3	1	1BF	1F8	249	290
346	CS3	2	1BF	1F8	249	290
371	CS3	3	1BF	1F8	249	290
2302	CS4	1	1F0	234	27A	1A9
2384	CS4	2	1F0	234	27A	1A9
2749	CS4	3	1F0	234	27A	1A9
192	CS5	1	331	375	3BB	3F6
237	CS5	2	331	375	3BB	3F6
262	CS5	3	331	375	3BB	3F6

Table B1: Address offsets for Metadata gathered from Photoshop *psd* file.

B2

Size of file(kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted images	Name of document	Author name	Creator tool
848	CS3	1	CA886	C4895	n/a	n/a	CA948
880	CS3	2	CABC7	C8501	n/a	n/a	CAC94
932	CS3	3	CC28A	1C349	n/a	n/a	D1ED1
1124	CS4	1	10F24E	C8CD0	n/a	n/a	10F31F
1176	CS4	2	10F24E	C8D03	n/a	n/a	10F31F
1224	CS4	3	11A24E	C93DF	n/a	n/a	11A31F
332	CS5	1	4A20D	3AB34	n/a	n/a	4A2DE
368	CS5	2	4B20D	3A5A4	n/a	n/a	4B2DE
408	CS5	3	5120D	3A5D8	n/a	n/a	512DE

Table B2 Address offsets for Metadata gathered from In-design *indd* files.

B3

Size of file(kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted images	Name of document	Author name	Creator tool
868	CS3	1	CA9D4	C4895	n/a	n/a	CAA96
896	CS3	2	10C0F	1044C	n/a	n/a	10CD1
940	CS3	3	10E4C	1044C	n/a	n/a	10F0E
1140	CS4	1	1128B	C8CD0	n/a	n/a	1135C
1188	CS4	2	1128B	C8CF8	n/a	n/a	1135C
1236	CS4	3	1128B	C92F8	n/a	n/a	1135C
412	CS5	1	4A20D	3AB34	n/a	n/a	4A2DE
372	CS5	2	5120D	3AC78	n/a	n/a	512DE
408	CS5	3	5820D	3AD00	n/a	n/a	582DE

Table B3 Address offsets for Metadata gathered from In-design *indt* files.

B4

Size of file(kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted images	Name of document	Author name	Creator tool
95	CS3	1	E8CA	E231	17901	n/a	E98C
99	CS3	2	EB05	E342	18759	n/a	EBC7
102	CS3	3	ED37	E342	194B9	n/a	EE04

Table B4: Address offsets for Metadata gathered from Indesign incx files.

B5

Size of file(kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted images	Name of document	Author name	Creator tool
44	CS3	2	4D56	4A2C	n/a	n/a	4DD9
44	CS3	3	4D56	4A2C	n/a	n/a	4DD9

Table B5 Address offsets for Metadata gathered from In-design snippet *inds* file.

B6

Size of file(kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted images	Name of document	Author name	Creator tool	String events
48	CS5	1	5A15	542F	n/a	n/a	59D7	BDF6
49	CS5	2	5C7C	508F	n/a	n/a	5C3E	C092
49	CS5	3	5C7C	508F	n/a	n/a	5C3E	C092

Table B6: Address offsets for Metadata gathered from In-design markup snippet *idms*..

B7

Size of file(kb)	Version of App	Number of images inserted	Offset for time stamps	Inserted images	Name of document	Author name	Creator tool	String events
66	CS5	1	A304	9D1E	n/a	n/a	A2C6	106E5

Table B7 Address offsets for Metadata gathered from In-copy markup snippet *icml file*

Appendix C: Table for Prototype Displayer

C1

Display in Prototype	Full Description	Associated Tag / Prefix
FILE EDITING INSTANCES	Events of document creation, can be saving, editing or changing	stEvt <stEvt:when> <stEvt:Action>
DATE_THE_DOCUMENT_WAS LAST MODIFIED	The timestamp for the last date of modification for the document	</xap:ModifyDate>
DATE_THE_DOCUMENT_WAS_FIRST CREATED	The timestamp for the date the document was first created	%%CreationDate </xap:CreateDate>
METADATA_DATE_OF_DOCUMENT	The timestamp for the date the metadata was last changed	</xap:MetadataDate>
APPLICATION_USED_TO CREATE THE DOCUMENT	The name of the graphic design application used to create the document	Creatortool %%Creator
NAME OF AUTHOR	The name of the author or the log in name when the document was created	%For
ORIGINAL DOCUMENT NAME	Is the default name in which a document was named when it was initially created	%%Title x-default Title Rdf:li Link Self AsMt
NAMES OF OBJECT INSERTIONS	The name of an image that was inserted into the document during editing	<stRef:file, %%Document file stRef:lastURL <clnk

Table C1: Description of Binary display for prototype

C2

Probability formulae

Please NOTE: the probability is calculated only for comparison purposes for the investigator to compare different documents under investigation. The main purpose of the forensic evaluation is for the investigator to be presented with the evidence automatically without having to physically examine each evidence on its own.

5 items considered under investigation for calculating probability

Each equivalent to (100/5 items) = 20 % (per item 1 – 5 below)

1. Author under investigation (A)

If author is match author probability A = 20 (otherwise = 0)

***Motivation**: If a particular author is known specially for counterfeit documents, a match results in an increase in the probability that the document under investigation might be counterfeit.*

2. File name changed (B)

If filename is changed file probability B = 20 (otherwise = 0)

***Motivation**: If a file name is changed by an author, this action can be seen as a way to hide the perpetrator actions; this action increases the probability that the document might be counterfeit.*

3. Number of images inserted (C)

if (Number of images inserted = 0) probability C =0;
if (Number of images inserted = 1) probability C =5;
if (Number of images inserted = 2) probability C=7;
if (Number of images inserted = 3) probability C=10;
if (Number of images inserted >= 4) probability C=15;

***Motivation**: The more the number of images inserted the higher the probability that the document might be counterfeit.*

4. # of times doc edited (D)

if (total editions <= 5) probability D =5;

if (total editions <= 10) probability D=7;
if (total editions <= 20) probability D=10;
if (total editions > 20) probability D=15;

***Motivation**: The more the number of times the document was edited the higher the probability that the document might be counterfeit.*

5. # of artifacts recognised (E)

if (# of artifacts recognised <= 10) probability E= 5;
if (# of artifacts recognised <= 20) probability E= 7;
if (# of artifacts recognised <= 30) probability E= 10;
if (# of artifacts recognised > 30) probability E = 15;

***Motivation:** The more the number of artifacts recognised within the document the higher the probability that the document might be counterfeit.*

TOTAL PROBABILITY = PROBABILITY (A + B + C + D + E)

Note: The accuracy of the probability is weak if it is used to compare different file types. This is because different file types contain different evidence identifiers. For example, Photoshop files do not contain the artifact that records the number of times a document was edited. Unlike an illustrator file, that records this artifact. In the end the Photoshop has already has a lesser probability before comparison.

Appendix D: Published Papers

The papers are added in an 'as is' format in order to preserve the formatting in which the papers were published, this includes retaining original page numbering format. These papers are presented in a chronological order of date published. The first paper, which is concerned with demonstrating an overview of how digital evidence can be gathered from graphic design applications, is titled “Finding digital forensic evidence in graphic design applications.” The second paper entitled “User-generated digital forensic evidence from graphic design applications” demonstrates the evidence that is created by the user intentionally as described in chapter eight of this dissertation. The third paper is a journal paper titled “Analysing registry, log files and prefetch files in finding digital evidence in graphic design applications”. The paper focuses on querying the system if there is any evidence of installation or usage of a graphic design application and if there is any link to the crime under investigation.

These papers start on the subsequent page.

(2012d1)Finding Digital Forensic Evidence in Graphic Design Applications

7th International Workshop on digital forensics and Incident Analysis, WDFIA 2012 Crete, Greece, pp12-26

Enos K. Mabuto[1], H. S Venter[2]

[1, 2] Department of Computer Science
University of Pretoria, Pretoria, 0002, South Africa
Tel: +27 12 420 3035
emabutos@cs.up.ac.za[1], hsventer@cs.up.ac.za[2]

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 like identity documents (IDs), driver's licenses or passports among others. However the use of any graphic design application leaves behind traces of digital information which 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. This paper reviews the digital forensics analysis process involved in the creation of counterfeit documents by determining and corroborating the events that previously occurred. The analysis is achieved by associating the digital forensic information gathered to the possible actions taken, precisely, the scanning, editing, saving and printing of counterfeit documents. The digital forensic information is gathered by analyzing the files generated by the particular graphic design application used for document creating. Another analysis is conducted on user generated files, the actual files that can be used as potential evidence to establish file structural contents and the relationship with the associated actions. This involves analyzing the user generated files associated with these applications and determining their signatures and related metadata. Contextually, the authors illustrate an evaluation disclosing the digital forensic evidence gathered from graphic design applications.

Keywords

Digital evidence, Digital forensics, Digital forensic artifacts, Graphic design applications

1. Introduction

A great number of professions and industries such as advertising, newspaper printing, architecture, fashion and design, project management and manufacturing, depend upon being able to create complex graphic designs in the course of their work.It is for this reason that graphic design applications have numerous image-enhancing tools such as paint brushing, vector drawing, digital pen and pencil drawing and many others.Such graphic design applications use computer-aided design to create unique art for company logos, magazine advertisements and many other purposes. There are numerous individuals who rely upon being able to use graphic design applications to create visual presentations that utilize pictorial images to communicate and express ideas.

In another related development, the use of forged documents has become ubiquitous all over the world.Ilham Rawoot observes, in an article in the "Mail and Guardian" that terrorists make particular use of forged South African passports because of the ease with which these can be faked (Mail and Guardian Website, (2011)). But counterfeit documents are in circulation all over the world. The same graphic design applications that are used by professionals in their work can also be used for illegitimate purposes such as creating counterfeit documents. The problem is that, with the editing and design capabilities of these graphic design applications, they can be used to create extremely convincing counterfeit documents such asIDs, passports and drivers licenses. Criminal activities such as these necessitatethe need for digital forensic investigations.

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 followingphases:the acquisition, examination, analysis and reporting (U.S National Institute of Justice, 2001). Wherever an individual is suspected of creating counterfeit documents, the regular process of acquisition is followed. Generally the phases of acquisition and

reporting are similar in different cases; therefore focus is on the examination and analysis phases. The focus is also on determining what the examiner needs to know prior to examining digital evidence. This paper identifiesand discusses the digital traces that are left behind after a counterfeiter has used graphic design applications. This is achieved by associating the actions taken during document creation to the traces left behind. In addition, a file analysis of files generated by a user from within the application is conducted. To address the problem, the authors focus on the following two steps. First, identify the digital forensic information that shows whether a document was scanned, edited, saved and printed.Digital forensic information can be found in graphic design applications where the source of the evidence is mainly system-generated. The second step entails identifying the contents of user-generated files by looking at the file signatures and related metadata. In so doing, over above these two steps, an association with the potential criminal may be achieved. However, it is not the focus of this paper to link the crime to an actual person.

The remainder of the paper is structured as follows. In the second section,some background about digital forensics is presented,and this is followed by a brief survey of graphic design applications. The third section,which is the contributing section, is divided into two parts. The first part highlights the potential evidence that the authors refer to as “digital forensic artifacts”. The source of potential evidence referred to above equates to the results from actions taken. More precisely the actions involved could be document scanning, editing, saving and printing. Most of this would originate from the system registry and application log files. The second part is an examination of user-generated files. The source of potential evidence referred to in this part involves results from content identification and content examination of files utilized by graphic design applications. The authors also name the tools that can be used in aiding the analysis where applicable. The fourth sectioncontains an evaluation of the kind of evidence that may be extracted from the graphic design applications. The fifth section concludes the paper.

2. Background

In the following section the authors provide some brief background literature on digital forensics including an explanation of digital evidence. The authors also define what is meant by digital forensic artifacts. The second section of the background consists of a very brief literature survey on graphic design applications.

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 unauthorized actions shown to be disruptive to planned operations.

The goal of a digital forensic investigation on a system is to find out what happened and who was responsible for a particular incident or crime. Digital forensic investigations focus on finding digital evidence after a computer or network security incident has occurred or locating data from systems that may form part of some litigation, even if it has been deleted. In this context, evidence is the most critical in any case.Therefore any items that can be considered to be of evidential value should be identified and collected (A.Jones. C.Valli, 2008).

2.1.1 Digital Evidence

Computer evidence or digital evidence is defined as any hardware, software or any data that can be used to prove one or more of the “who, what, when, where, why and how” questions of a security incident (M.G Solomon, D.Barett, N.Broom 2005). Computer evidence furthermore consists of digital files and their contents left behind after an incident. Casey defined digital evidence as any data that can be used to establish that a crime was committed or can prove a link between a crime and its victim or an offender (E.Casey, 2000). Digital evidence consists entirely of sequences of binary values called bits (F. Cohan, 2010). Traces that are left behind from the use of an application or from an operating system can be referred to as digital forensic artifacts.

2.1.2 Digital Forensic Artifacts

An examiner reveals the truth of an event by discovering and exposing the remnants of the event that have been left on the system. These remnants are known as artifacts, which can be referred to as digital evidence (C.Altheide, H.Carvey, 2011)However, due to the loaded legal connotations binding the term “evidence” the term “artifacts” is used more often. Evidence is referred to as something to be used during a legal proceeding. Artifacts are traces left behind due to activities and events, which may or may not be innocuous. 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. Also an artifact does not become evidence unless its ability to prove a fact has been established (M.V.Zelkowitz , 2009). Therefore it is necessary to reconstruct events that occurred by gathering all the possible digital information from a system.

In an investigation, 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

Although many graphic design applications are currently available to users, Adobe Systems Incorporated is regarded as the largest software maker in the graphic design software category (Wall-street Journal Website, 2011). For the purposes of this research, the authors therefore undertook a case study by investigating Adobe graphic design applications. Adobe Photoshop, Adobe In-Design and Adobe Illustrator are Adobe applications that are used for graphic design purposes. Any one of these applications can be used for the editing of a document. It is therefore necessary to conduct an exclusive examination of the potential digital forensic evidence produced by these applications. Since most graphic design users prefer to use the latest editions, the authors used the latest version of Adobe (Version, CS5) in the experiments. It should be noted, however, that additional experiments with the two previous versions (CS4 and CS3) produced similar results. Any slight differences that are attributed to different versions will be mentioned wherever necessary throughout the paper.

3. Digital forensic evidence in graphic design applications

The authors created dummy counterfeit documents by using Adobe graphic design applications, and carried out various experiments in order to search for pertinent evidence left behind from the use of these graphic design applications.The contribution is divided in two parts. The first part highlights digital forensic artifacts found in graphic design applications where the source of the potential evidence is mainly system-generated with results mostly from registry entries and application log files. The second part of the experiments, which involves examination of user-generated files, highlights results from file content identification and examination.

Software reviews from 2011 revealed that the Windows operating system is still the most popular operating system (Gartner Research Website, 2011). The authors therefore conducted the analysis for forensic artifacts on a Windows 7 platform. For future work, focus can also be placed on other popular operating systems like Linux and Mac OS.

To respond to the problem stated earlier, that graphic design applications can be used for creating counterfeit documents, firstly four possible actions taken during the creation of a document were used as a hypothesis to gather digital forensic information related to graphic design applications.These actions are document scanning, document editing, document saving and document printing. The analysis is formulated to find the digital forensic information that indicates that these actions were actually taken. By following the actions taken an investigator is able to conduct an investigation in a uniform manner that helps to acquire the actual images like a human face used to create the document and the created counterfeit document. For example, if the document was scanned, then probably it was then edited. If not scanned then probably it was edited by acquiring a copy of the original from another source. If not edited then probably it was printed only after being edited from another source. If none of the four actions were taken then there is no need to ascertain whether or not the application was used in the creation of the document.

Furthermore to respond to the same problem, a user-generated file analysis section follows, with two sub-sections dealing with content identification and content examination.

Experimental results gleaned from finding the four actions are elaborated in the each of the subsections to follow.

3.1.System-generated digital forensic artifacts

"System-generated digital forensic artifacts" refer to those artifacts created by the application without user intervention, while "user-generated digital forensic artifacts refer" to artifacts created by the user intentionally. The latter are discussed later in the paper.

For the experiments conducted, the following section describes the techniquesused on Adobe graphic design applications. Four sub-sections follow in this section, namely artifacts related to document scanning, editing, saving and printing. It should be noted, however, that not all applications have the same capabilities to perform all these actions. Therefore, not all actions are described for each graphic design application. However, initiation of one of the actions can lead to possible identification of potential evidence relating to the creation of counterfeit documents. The authors explain the artifacts gathered from each action precisely for each graphic design application. Adobe Illustrator does not record any information regarding the four actions in any of its log files. Therefore, for Adobe Illustrator essential information will be acquired from the exclusive examination of user-generated files still to follow in section 3.2.

3.1.1.Artifacts relating to document scanning

Generally, if one is to attempt to create a fraudulent document, one has to acquire an original document so as to imitate or copy it. Scanning is an option which results in a copy of the original document being available on pc for digital editing. Many different models of scanners are available, using various software packages for executing scan commands. Therefore, for this research, focus is on commands generated from within the graphic design application used for editing the scanned document, rather than determining if a document is a scanned document.

Out of the three graphic design applications under consideration, only Adobe Photoshop has the capability of scanning a document using the "import WIA support" document menu option. "Import WIA support" is a function that Adobe Photoshop uses to connect to available printers or scanners. The document scanned is loaded into a destination folder as prompted. The application then creates a folder, saves the scanned image, and opens the scanned image in the application.

After a document is scanned the application records the entry into one of its log files under the name of *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 particular Adobe version in use. This may be either version 3, 4, or 5. After analyzing the log file's binary data an entry with the location of the scanned file is located usually about mid section of the file size. For example, if the file is 165kb the scanned file information will be located at hex byte offset 0x17004. After analyzing the content at this hex location, the folder locations of all the scanned documents can be found there.

The regular process followed by a potential criminal is to edit the acquired document in order to falsify some of its content.

3.1.2.Artifacts relating to document editing

Document editing is one of the critical stages of creating a counterfeit document as it allows one to place or import objects of interest, for example a human face, a bar code or a fingerprint. These objects can be inserted onto the scanned document. In relation to the inserted documents or files, experiments were executed to establish what can be found from a system that indicates to the examiner what was inserted and from which location it was inserted from. All three graphic design applications in question have the capacity to edit a document through attaching or placing an image. The terms "attaching" or "placing" an image is seen as the same action, used interchangely in various applications. In this paper, the term "attaching" is used from here on. Attaching is one of the main functions that is used in graphic design applications.

3.1.2.1. Editing in Adobe Photoshop

The same log file mentioned earlier *(Adobe Photoshop CSX Prefs)* contains information with the name of the attached file and the location from which it was attached usually at a byte offset of about 0x17F40. With this information the authors managed to establish the names and location of attached documents. Furthermore, by looking at the stated location the actual image with the human face or fingerprint was found.

3.1.2.2. Editing in Adobe Indesign
A file named *InDesign SavedData* without a file extension is located in the located in the folder *C:\Users\<username>\AppData\Local\Adobe\InDesign\Version 5.0\Cache*. It contains information indicating the name of the attached file and the location from which it was attached usually in the beginning of the file.

3.1.3.Artifacts relating to document saving

Once a document has been edited, usually a user (or potential criminal) might need to save it, either for printing or further editing. In this section the authors look at what is found in the system that relates to saved documents. This information is vital as it can point to an examiner where a file was saved to. If deleted or moved, search commands can be executed based on the names of the files saved. This is done by specifying the name of the file when searching thereby reducing time spent during an investigation. All three applications under consideration have the capability to save edited documents in various file types. An exclusive examination on each of the file types created from saving actions is explained in section 3.2.

3.1.3.1. Saving in Adobe Photoshop
The same log file *(Adobe Photoshop CSX Prefs)* contains information about save entries. The file contains information about the name of the saved file, the location in which it was filed, and type of the file, located at mid offset of the file after the entries for attached files. The names are arranged in order of the last saved file first. This information about saved locations 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\11.0\VisitedDirs.*

3.1.3.2. Saving in Adobe Indesign
The same log file *(InDesign SavedData)* that was earlier mentioned in connection with editing actions, contains information about the name of the file saved, type of the file and the location saved to. This information is located from mid offset of the file with the last saved file first. This information is located up to the end of the file depending on the number of documents saved.

Generally, saved files from any graphic design application can be verified or checked also by looking at the recent documents accessed from *C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Recent.*

3.1.4.Artifacts relating to document printing

Printing is one if not the last stages of potential counterfeit document creation. A user might need to create the 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 under consideration via the menu command: print.

To locate which printer(s) are used to generally print a document one uses the registry. The keys from which a list of printers connections could be established from are (1)*HKLM\soft\Adobe\Photoshop\11.0\Plugin* path.
(2)HKEY_CURRENT_CONFIG\System\CurrentControlSet\Control\Print\Printers
(3)HKEY_USERS\<user id>\Software\Microsoft\Windows NT\CurrentVersion\PrinterPorts
(4)HKEY_USERS\<userid>\Software\Microsoft\Installer\Products\41E0A130314079C4792762937B284FF6\SourceList

After the names of the printers have been established, an investigator can verify the physical existence of the printer. This helps an investigator usually in cases where the printers have been physically removed. Moreover, given that the option to keep printed documents was enabled in the printers' properties before printing a counterfeit document. For each print job there are two spool files generated by the operating system located in *C:\Windows\System32\spool\PRINTERS.* The first is *XXX.shd* and *XXX.spl* where *XXX* represents the job number.

Analyzing the binary data of these files indicates the name of the printed document in the beginning of the *.*spl* file. Towards the end of the *.*shd* file is the name of the printed file, the location from which it was printed from and the name of the printer used to print the document. The timestamp of the *.*spl* and *.*shd* file indicates the date and time the document was created. This information is vital in establishing which counterfeit documents were actually printed.

Once the names and locations of the files have been established, an investigator needs to examine the actual identified files. These are the files that can be used as potential evidence in legal proceedings. This process is described in the following section.

3.2User-generated artifacts from file examination

In order to conduct an exclusive examination on a crime conducted within an application the digital forensic examiner has to understand the nature of the files that are generated from that particular application, in this case, graphic design applications. This is so that the examiners are able to uncover and exploit any digital forensic artifacts present in the identified files (C.Altheide, H.Carvey, 2011).

As previously stated, user-generated digital forensic artifacts refers to files created by the user intentionally. User generated file artifacts can be divided into two distinct categories, which are, content identification and content examination. Content identification is the process of determining or verifying the type of a specific file. Content examination is the retrieval of any embedded metadata that may be present in a given file.

In the case of the examination of counterfeit documents the digital forensic examiner might need to identify potential changes inside files consistently, for example, the involvement of a fingerprints, barcodes or human faces embedded inside graphic design application file formats. The four graphic design applications discussed above are associated with more than thirty nine file types. However, for this research the authors focus was only on file types that are specific to the three graphic design applications, thus ignoring well-known file types like jpeg, bitmap, tag, tiff, tga etc. Gary Kesler and Martin Reddy keep a list of these common file signatures online, which is a continuing work in progress database (Gary Kesler and Martin Reddy Website (2011)).

3.2.1 Content identification

As already been stated, content identification involves verifying the identity of a file extension. An offender can alter the file extension of a particular file in order to promote ambiguity. Therefore there is need to identify a files integrity by file signature analysis. An examiner needs to know what a particular file type is. A file is normally analyzed within its first bytes to determine the specific signature (H.Carvey, 2009). The file signature is therefore located at specific offsets usually in the beginning of a file.

It can be noted from the research conducted that known digital forensic tools like FTK can detect various file types but not for graphic design applications discussed in this paper. For example, digital forensic tools can verify file types like tga, bmp, gif, tif and png amongst others, but not the file types of graphic design applications as discussed in this paper.

The analysis to determine a graphic design file signature was also conducted using a hex editor. These values are generally hexadecimal values. Table 1 contains the list of file signatures identified and specific to the graphic design applications previously discussed. The file type in Table 1 represents the named form of the particular graphic design file. Proof of the real file identity resides within the content of the file, usually known as the file signature. The file extension is merely a suffix that represents the encoding of a file's content, usually three or four characters separated by a dot from the file name. However, the file extension should never be trusted as it can be renamed to anything else. One should rather focus on the file signature to determine the correct file type. The ASCII column in Table 1 represents the entry in text-readable format. The file signature columns represent the entry in hexadecimal format. Both these entries appear exactly as shown in the hex editor. The digital forensic examiner can use the information in Table 1 to identify the particular files for the graphic design applications in question.

File Type	File extension	ASCI II	File signature
Illustrator file	ai	%PDF-1.5	25 30 44 46 2D 31 2E 35
Photoshop	psd	8BPS..	38 42 50 53 00 01
Indesign markup	idml	PK…….	50 4B 03 04 14 00 00 00
Indesign interexchange	incx	<?xml version="1.0"	3C 3F 78 6D 6C 20 76 65 72 73 69 6F 6E 3D 22 31 2E 30 22
Illustrator Postscript	eps	ADOE	C5 D0 D3 C6
Photoshop dcs2	eps	ADOE	C5 D0 D3 C6
Indesign document	indd	…… ……	06 06 ED F5 D8 4D 46 E5 BD 31 EF E7 FE 74 B7 1D 44 4F 43 55
Indesign template	indt	…… ……	06 06 ED F5 D8 4D 46 E5 BD 31 EF E7 FE 74 B7 1D 44 4F 43 55
Illustrator template	ait	%PDF-1.5	25 30 44 46 2D 31 2E 35

Table 1: Graphic design file signatures

3.2.2 Content Examination

Content examination involves determining the metadata of files, in this case, graphic design application file types. Metadata refers to data about data (C.Altheide, H.Carvey, 2011). On Windows systems this includes modified, accessed, creation times only to mention a few. The same hex editors, as previously stated, are used to examine the content of files associated with graphic design applications. Metadata is essential during an investigation as this reveals what useful information can be extracted from a particular file, for example this can be time stamps or name of the user who created the file.

Table 2 shows the metadata acquired from graphic design file types. The offset is the address pointer of the described metadata. In other words, if an examiner searched for a certain offset, the hex editor would skip to the particular metadata. Several experiments however revealed that the offset can vary slightly by plus or minus 780 bytes per metadata, which is usually in the same page view depending on the size of the file and quantity of metadata present in the file. Therefore the tabulated values can still be used on graphic design files of different sizes. The metadata is embedded in Extensible Metadata Platform (xmp) tags, which is Adobe's way of embedding metadata in its various file types (Adobe XMP, 2011).

File type	File extension	Description of Metadata	Offset (Address pointer to Metadata)	Example of the Metadata (As presented in a hex editor)
Indesign document	indd	File location for any imported image files	D9EB	file:C:/Users/<username>\/Pictures/ dvd%20picture%20sleeves/Capture_005% 20%282%29.JPG
		Name of application that created the file	E510B or E6E16	<stEvt:softwareAgent>Adobe InDesign 6.0</stEvt:softwareAgent>
		String events of saving history	F0D0C to F12FE	<stEvt:action>created</stEvt:action><stEvt:when>2011-05-04T15:13:25+02:00</stEvt:when>stEvt:action>saved</stEvt:action><stEvt:when>2011-05-04T15:15:43+02:00</stEvt:when>
		Date file was created	F5263	CreateDate>2011-05-04T15:13:25+02:00
		Metadata Date	F52A7	MetadataDate>2011-05-04T15:18:24+02:00</xmp:MetadataDate
		Modify Date	FD2EA	<xmp:ModifyDate>2011-05-04T15:18:24+02:00</xmp:ModifyDate>
Illustrator Postscript file	eps	Name of application that created the file	57	%%Creator: Adobe Illustrator(R) 14.0
		Date file was created	8E	%CreationDate: 9/17/2011

Table 2: Graphic design file types related metadata

File type	File extension	Description of Metadata	Offset (Address pointer to Metadata)	Example of the Metadata (As presented in a hex editor)
		Login name of user that created the file	73	%%For: <username>\ %
Illustrator file	ai	Metadata Date	3A7	<xmp:MetadataDate>2011-05-04T15:51:17+02:00</xmp:MetadataDate>
		Date file was modified	3ED	<xmp:ModifyDate>2011-05-04T15:51:17+02:00</xmp:ModifyDate>
		Date file was created	431	<xmp:CreateDate>2011-05-04T15:51:17+02:00</xmp:CreateDate>
		Name of application that created the file	476	<xmp:CreatorTool>Adobe Illustrator CSX</xmp:CreatorTool>
Photoshop file	psd	Name of application that created the file	1A9	<xmp:CreatorTool>Adobe Photoshop CSX Windows</xmp:CreatorTool>
		Date file was created	1F0	<xmp:CreateDate>2011-05-04T14:39:08+02:00</xmp:CreateDate>
		Date file was modified	234	<xmp:ModifyDate>2011-05-04T14:50:23+02:00</xmp:ModifyDate>
		Metadata date	27A	<xmp:MetadataDate>2011-05-04T14:50:23+02:00</xmp:MetadataDate>
		String events of saving history	6FF to 717	<stEvt:instanceID>xmp.iid:DE0657134D76E011B00EFDC555D228CB</stEvt:instanceID><stEvt:when>2011-05-04T14:50:23+02:00</stEvt:when>
Illustrator template	ait	Name of application that created the file	1F3 or 452	<xmp:CreatorTool>Adobe Illustrator CSX</xmp:CreatorTool>
		Metadata Date	383	<xmp:MetadataDate>2011-05-04T15:51:17+02:00</xmp:MetadataDate>
		Date file was modified	3C9 or 16323	<xmp:ModifyDate>2011-05-04T15:51:17+02:00</xmp:ModifyDate>
		Date file was created	40D	<xmp:CreateDate>2011-05-04T15:51:17+02:00</xmp:CreateDate>
		String events of saving history	D02B or D5D3	<stEvt:action>saved</stEvt:action><stEvt:instanceID>xmp.iid:FF7F11740 7206811B628E3BF27C8C41B</stEvt:instanceID><stEvt:when>2011-05-22T16:23:53-07:00</stEvt:when>
		Name of user that created the file	17FB9	%%For: (Pinchers) ()
		File path for any imported images	D727	%%DocumentFiles:C:\Users\<username>\Pictures\Sizzla-Soul Deep-Front.jpg %%+C:\Users\<username>\\Pictures\Tulips.jpg
		List of previous files names used	180A8	/Title(illustrator .ait template)
Indesign interexchange file	incx	Date file was created	BFD3	<xmp:CreatorTool>Adobe InDesign 6.0</xmp:CreatorTool>
		Metadata Date	C019	<xmp:MetadataDate>2011-05-04T15:17:21+02:00</xmp:MetadataDate>
		Date file was modified	C05F	<xmp:ModifyDate>2011-05-04T15:17:21+02:00</xmp:ModifyDate>
		Date file was created	BD3A	<xmp:CreateDate>2011-05-04T15:17:21+02:00</xmp:CreateDate>
		Name of application that created the file	C0A4	<xmp:CreatorTool>Adobe InDesign 6.0
		String events of saving history	108C2 or 115F7	<stEvt:instanceID>xmp.iid:972E234B5076E011AAFBC6ED1F893037</stEvt:instanceID><stEvt:when>2011-05-04T15:17:21+02:00</stEvt:when>
		Last file path used	119D8 or 11C4d	%%DocumentFiles:C:\Users\<username>\\Pictures\Sizzla-Soul Deep-Front.jpg %%+C:\Users\<username>\\Pictures\Tulips.jpg

Table 2: Graphic design file types related metadata (continued)

File type	File extension	Description of Metadata	Offset (Address pointer to Metadata)	Example of the Metadata (As presented in a hex editor)
	incx	Previous file format used	15BD2	<xmpGImg:format>JPEG</xmpGImg:format>
Indesign template file	indt	File path for any imported images	CF1E0 or D4F03	%%DocumentFiles:C:\Users\<username>\\Pictures\Sizzla-Soul Deep-Front.jpg %%+C:\Users\<username>\Pictures\Tulips.jpg
		Date file was created	D72AB	<xmp:CreateDate>2011-05-04T15:17:21+02:00</xmp:CreateDate>
		Metadata Date	D72F1	<xmp:MetadataDate>2011-05-04T15:17:21+02:00</xmp:MetadataDate>
		String events of saving history	D3DBA to D3F46	<stEvt:instanceID>xmp.iid:972E234B5076E011AAFBC6ED1F893037</st Evt:instanceID> <stEvt:when>2011-05-04T15:17:21+02:00</stEvt:when>
		Name of application that created the file	D400C or D737C	<xmp:CreatorTool>Adobe InDesign 6.0>

Table 2: Graphic design file types related metadata (Continued)

4. Discussion

From the case study the authors managed to establish the location from which scanned documents were saved to. In this location several other documents were also recognized to indicate the names and original identities of documents. For the action of editing the authors established the names file types and file locations of attached documents. These were fingerprint and human face images inserted onto a copy of the original documents. Following editing, saving actions produced artifacts revealing the names of the saved files, their file types and their locations. These saving actions enabled recognition of potential evidence as they contained the actual counterfeit documents. For the printing action results from registry and log files indicated the names of the printers used and the names of the printed documents.

For user-generated file analysis all graphic design application file types analysed have timestamps as part of their metadata. However only a few of them have the user name of the creator of thefile as part of the metadata. Table 4 summarises the user-gernerated file types. “Yes” in this table indicates that the described metadata is present while “No” denotes that the file type does not contain the described metadata. The headings of the columns are brief names of descriptions of the metadata that was previously tabulated in Table 2.

File format extension	Date of creation	Date of modification	Meta data date	Creator user name	Creator tool	Location of imported images	String events
indd	Yes	Yes	Yes	No	Yes	Yes	Yes
indt	Yes	Yes	Yes	No	Yes	Yes	Yes
incx	Yes	Yes	Yes	No	Yes	Yes	Yes
ai	Yes	Yes	Yes	No	Yes	No	No
ait	Yes	Yes	Yes	Yes	Yes	Yes	Yes
psd	Yes	Yes	Yes	No	Yes	No	Yes
eps	Yes	No	No	Yes	Yes	No	No

Table 4: Summary of User generated file analysis

Given that a digital forensic investigation was initiated on a suspected counterfeit document creation crime, and the document was generated using a graphic design application. And using the identified digital forensic artifacts a digital forensic examiner is able to establish the route at which the document was created and to corroborate the gathered evidence. For example the digital forensic examiner is able to discover the human face, fingerprint, and or bar code images used to create the document. Together with the actual counterfeit document these can be presented in the court for prosecution. By presenting proof of the actions taken during document editing the process followed can provide valuable support in the court.

For content identification, the digital forensic examiner can use the recognized file signatures and the corresponding ASCII text representation to determine the file type of the graphic design applications in question. The file signatures can also be used when searching files from a formatted hard drive. Also an in-depth analysis of user-generated files can assist an examiner in knowing which particular metadata to acquire from graphic design file types and at what offset address.

Recalling that computer evidence is defined as any hardware, software or any data that can be used to prove one or more of the "who, what, when, where, why and how" of a security incident. By reviewing all the artifacts gathered the definition of digital evidence can be confirmed. This is so because all the six questions, "who, what, when, where, why and how" of the digital evidence definition are validated from the results acquired. Briefly clarifying the results: the "who" was specified by an artifact with the user name, the "what", specified by identifying the particular files types from the application, the "when", specified with a registry artifact indicating time of incident, the "where" specified with an artifact showing the file location, the "why" specified with a file metadata extraction revealing the file contents and the "how" with an artifact indicating which application was used for document editing. These results are essential for a digital forensic examiner to know where to look for digital forensic information, guided by knowing what information to find at a named particular location. This speeds up the process of an investigation where graphic design applications were used.

5. Conclusion

The approach outlined in this paper is particularly useful for solving those cases in which document editing is largely associated with a particular application. The approach only addresses case studies involving Adobe products but the same can be done for other graphic design applications. However, the approach doesn't tackle issues where the user only edits a hard copy, scans and prints without using any pre-installed application. Recalling the problem that graphic design applications can be used to create counterfeit documents, and that 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 .The techniques discussed can be incorporated in bigger digital forensic tools like FTK and Encase or possibly the design of a crime specific tool similar to a Porn detection stick, (Parabens software Website, 2011) which is a thumb drive device that will scan and detect pornographic content on a computer. Also, future work can be conducted by carrying out this exercise on other graphic design applications like CorelDraw.

6. References

M.G Solomon, D.Barett, N.Broom (2005). "Computer Forensics Jumpstart", Sybex, London, pp 51.

E.Casey (2000), "Digital evidence and computer crime", London, Academic Press, pp10.

Gartner Research Website (2011), "Which operating system will be 2011's bestseller", http://www.gartner.com/technology/research.jsp (Accessed 11 August 2011)

Wall Street Journal Website (2011), Dow Jones, "Adobe 2Q Net Up 54% On Broad Sales Gains..,"http://www.wsj.com, (Accessed 21 June 2011)

A.Jones. C.Valli (2008), "Building a digital forensic laboratory", Burlington, Elsevier, pp 285.

F.Cohan, (2010), "Towards a science of digital forensic investigation", IFIP Advances Digital Forensics VI, China, pp 17-35

M.V.Zelkowitz (2009), "Advances in computers; information security".Academic Press-Elsevier

Digital Forensic Research Workshop (2001), "A roadmap for Digital Forensic Research", pp 16.

U.S National Institute of Justice (2001) "Electronic Crime Scene Investigation Guide: A guide for First Responders", *NIJ Special report, 2nd Ed*, pp 10-47.

H.Carvey (2009), "Windows Forensic Analysis Dvd Toolkit", 2nd Ed, Elsevier, pp 296.

Mail and Guardian Website (2011), "Terrorists favour 'easy' fake SA passports", Mail and Guardian online, http://mg.co.za/article/2011-06-17-terrorists-favour-easy-fake-sa-passports (Accessed 17 June 2011)

H.Carvey (2009), "Windows Registry Analysis", 2nd Ed, Elsevier, pp 194.

Parabens software Website (2011), www.paraben-sticks.com/porn-detection-stick (Accessed 9 August 2011)

C.Altheide, H.Carvey (2011), "Digital Forensics with Open Source tools".Elsevier.MA USA, pp 2.

Appendix D: Published papers

Gary Kesler Website (2011), File signatures, http://www.garykessler.net/library/file_sigs.html, (Accessed 10 October 2011).

M. Reddy Website (2011) Graphic design file format database, http://www.martinreddy.net/gfx/2d-hi.html (Accessed 10 October 2011)

Adobe XMP Website, (2011) http://www.adobe.com/ products/ xmp/index.html (Accessed 11 November 2011)

(2012d2)User-generated Digital Forensic Evidence in Graphic Design Applications

International conference on cyber security cyber warfare and digital forensics, CyberSec2012, pp 195-200.

Enos K. Mabuto[1], H. S Venter[2]

[1, 2] Department of Computer Science
University of Pretoria, Pretoria, 0002, South Africa
Tel: +27 12 420 3035
emabutos@cs.up.ac.za[1], hsventer@cs.up.ac.za[2]

***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 like identity documents (IDs), driver's licenses or passports among others. However the use of any graphic design application leaves behind traces of digital information which can be used during a digital forensic investigation. Current digital forensic tools do not examine a system specifically for the creating of counterfeit documents. The paper in hand reviews the digital forensics analysis process involved in the creation of counterfeit documents by determining and corroborating the events that previously occurred. The analysis is conducted on user generated files, the actual files that can be used as potential evidence to establish file structural contents. The acquired digital forensic information is corroborated to the creation of counterfeit documents and interpreted accordingly.**

***Keywords* - Digital evidence; Digital forensics; Digital forensic artifacts; Graphic design applications.**

I. 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.

The use of forged documents, however, is noticed all over the world.A report by Ilham Rawoot of the Mail and Guardian newspaper stated that terrorists target fake South African passports because of the ease with which one can be faked [1]. This shows that counterfeit documents are in circulation all over the world. The problem is that, with the editing and design capabilities of these graphic design applications, they can be used to create counterfeit documents like ID's, passports or drivers licenses. Criminal activities such as these necessitatethe need for digital forensic investigations.

Nevertheless, 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 followingphasesofacquisition, examination, analysis and reporting [2]. Generally the phases of acquisition and reporting are similar in different cases; therefore focus is on the examination and analysis phases. This paper identifiesthe digital traces from graphic design applications' files, those files generated by a user intentionally from within the application itself. To address the problem, the authors focus on identifying the contents of user-generated files by looking at the file signatures and related metadata. In so doing, an association with

the potential criminal may be achieved. However, it is not the focus of this paper to link the crime to an actual person.

The remainder of the paper is structured as follows. In the second section,some background of digital forensics is given, followed by a brief background on graphic design applications. The third sectionis an examination of user-generated files. The fourth section is an evaluation of the evidence that is extracted from the graphic design applications. Lastly a conclusion is given to end this paper.

4. BACKGROUND

In this section the authors provide some brief background literature on digital forensics including an explanation of digital evidence. A definition of digital forensic artifacts is found thereafter. The second section of the background consists of a very brief literature survey on graphic design applications.

4.1 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 unauthorized actions shown to be disruptive to planned operations[3]. In this context, any items that can be considered to be of evidential value should be identified and collected [4].

4.1.1 Digital Evidence

Computer evidence or digital evidence is defined as any hardware, software or any data that can be used to prove one or more of the “who, what, when, where, why and how” of a security incident [5]. Furthermore, Casey defined digital evidence as any data that can be used to establish that a crime was committed or can prove a link between a crime and its victim or an offender [6]. Traces that are left behind from the use of an application or from an operating system can be referred to as digital forensic artifacts.

4.1.2 Digital Forensic Artifacts

An examiner reveals the truth of an event by discovering and exposing the remnants of the event that have been left on the system. These remnants are known as artifacts, which can be referred to as digital evidence [7]. However, due to the loaded legal connotations binding the term “evidence” the term “artifacts” is used more often. Also an artifact does not become evidence unless its ability to prove a fact has been established [8].

In an investigation, 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.

4.2 Graphic design applications

Many graphic design applications are currently available in the industry; however, Adobe Systems Incorporated is regarded as the largest software maker in the graphic design software category [9]. Therefore, for this research, a case study was conducted with Adobe graphic design applications. Adobe Photoshop, Adobe In-Design and Adobe Illustrator are Adobe applications utilised for graphic design purposes. Any one of these applications can be used for document editing. Therefore it was necessary to conduct an exclusive examination for potential digital forensic evidence.

5. DIGITAL FORENSIC EVIDENCE IN GRAPHIC DESIGN APPLICATIONS

In this section, the authors first explain the method used for this study, referred to as the experiments. Secondly the authors illustrate the results obtained from these experiments, referred to as the gathered digital forensic artifacts. Lastly, a summary is given to elaborate the results.

A. Experiments

"System-generated digital forensic artifacts" refer to those artifacts created by the application without user intervention, while "user generated digital forensic artifacts refers" to artifacts created by the user intentionally. The earlier are not analysed in this paper.

The research experiments were exercised in two stages, experiment one and experiment two.Experiment one was conducted in order to simulate the activities that can be exercised by an offender. Experiment two was exercised so as to trace the activities conducted by the offender. The two experiments are explained below.

1) Experiment one: Creating the counterfeit documents

Three hundred dummy counterfeit documents were created using the mentioned graphic design applications in the background section.Fifty documents were created for each application per version. 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. These changes will be explained later in the results section. Following is a description of the hardware and software tools used for experiment one.

- Product version

Given the notion that most graphic design application users prefer the latest editions, for this study the latest version for Adobe, CS5 was used. However, further experiments with two previous versions CS4 and CS3 produced the similar results. Slight differences will be mentioned where necessary throughout the paper.

- Hardware Tools

Three different computers were used, each using a different Adobe version. However, bond paper (standard white printer paper, 80g/m^2) was used as output material for printing as opposed to the high quality paper used at an authentic factory. Output material did not conversely affect the gathered digital forensic evidence.

- Platform used

Platform refers to the operating system on which the counterfeit documents were created on. Software reviews in 2011 revealed that the Windows operating system is still ranked the most popular operating system [10, 11]. Therefore, the analysis for digital forensic artifacts was conducted on a Windows 7 platform.

2) Experiment two: Searching for the evidence

After creating the counterfeit documents, experiments were then carried out in order to search for pertinent evidence left behind from the use of the graphic design applications.Following is a description of the hardware and software tools used for experiment two.

- Tools

The operating systems' registry editor tool, "regedit" was used to search for associated registry entries. Also a hex editor, [17] was used for analysing the binary data of the log files.

- Method

When examining counterfeit documents, the digital forensic examiner might need to identify potential changes inside files consistently, for example, the involvement of a fingerprints, barcodes or human faces embedded inside files which the authors refer to as elements.The three graphic design applications mentionedin the background section are associated with more than thirty nine file types. However, for this research the authors focus was on examining only file types that are specific to the three graphic design applications, thus ignoring well-known file types like jpeg, bitmap, tag, tiff, tga etc. Gary Kesler and Martin Reddy keep a list of these common file signatures online, which is a continuing work in progress database [12, 13]. Experimental results gleaned from the user-generated files are elaborated in the content identification and examination to follow.

B. Content identification

Content identification involves verifying the identity of a file extension. An offender can alter the file extension of a particular file in order to promote ambiguity. Therefore there is need to identify a files integrity by file signature analysis. A file is normally analyzed within its first bytes to determine the specific signature [14]. It can be noted from the research conducted that known digital forensic tools like FTK can detect various file typeslike tga, bmp, gif, tif and png amongst others but not for graphic design applications discussed in this paper.

Table 1 shows the hexadecimal signatures of the examined files. The file type in the signature table represents the named form of the particular graphic design file. The file extension is merely a suffix that represents the encoding of a file's content. However, the file extension should never be trusted as it can be renamed to anything else but rather focus on the file signature. The ASCII column represents the entry in text-readable format. The file signature columns represent therequired entry in hexadecimal format. Both these entries appear exactly as shown in the hex editor [17]. This information can be used to identify the particular file for the graphic design application in question.

Table 1: Hexadecimal signatures for Adobegraphic design applications.

File type	File extension	ASCI II	File signature
In-design and In-design template	Indd, indt	íôØFå½1ïçþt·DOCUMENTp	06 06 ED F5 D8 4D 46 E5 BD 31 EF E7 FE 74 B7 1D 44 4F 43 55 4D 45 4E 54 01 70 0F
In-copy markup document, In-design XML Interchange document, and In-design markup snippet	icml, inx, idms	<?xml version="1.0" encoding="UTF-8" standalone="yes"?>	3C3F786D6C207665 7273696F6E3D2231 2E302220656E636F 64696E673D225554 462D3822207374616 E64616C6F6E653D2 2796573223F3E
In-design markup	idml	PK.....	50 4B 03 04 14 00 00 00
Photoshop	psd	8BPS	38 42 50 53 00 01
Illustrator file and Illustrator template	ai, ait	%PDF-1.5 %âãÏÓ 1 0 obj	255044462D312E35 0D25E2E3CFD30D0 A312030206F626A
Encapsulated post script	eps	ÅÐÓÆ	C5 D0 D3 C6

Adobe Illustrator file signatures follow the *pdf* signature convention of starting with pdf-1.5 and differs in follow up characters.It can be seen that template files take the signatures of their default file types for example, file indd and indt have the same file signature.

C. Content Examination

Content examination involves determining the metadata of graphic design application file types. Metadata refers to data about data [7]. Metadata is essential during an investigation as this reveals what useful information can be extracted from a particular file.

For each file type a table is shown that displays the metadata acquired from it and the respective address offset. The offset is the address pointer of the described metadata. Adobe uses the conventional metadata scheme Extensible Metadata Platform (XMP) [16]. It is an open and extensible scheme allowing it to be used in various file types. For each file the tagging used to embed the metadata is described.

1) Illustrator (ai)

Adobe Illustrator artwork 15.0 file type with extension **.ai* is the default file type for documents created with Adobe illustrator. The file contains metadata including time stamps, author name and names of inserted elements. The author name is recognized as the log-in name used on the computer. The metadata is recorded at certain offsets which will be described in the metadata table. Adobe Illustrator version CS3, the metadata is embedded in *xap* tags. for example*<xap:CreateDate>2012-04-16T14:21:48+02:00</xap:CreateDate>*. Version CS4 and CS5 metadata is embedded in *xmp* tags. Metadata showing the names of inserted elements and respective locations from which they were inserted from is embedded in *<stRef: file Path>* tags or prefix *%%Document file* concatenated with

a *%%+* symbol. The earlier consists of a single entry and the later consist of all the inserted elements per file. The author name is prefixed*%%For*. The default name for the file is the original name saved for the file which cannot be easily renamed and is embedded with-in the tag *<rdf:li xml:lang="x-default">*. The creator tool is the name of the graphic design application used to create the file. Table 2 shows the metadata embedded in Illustrator *ai* files.

Table 2: Address offsets for Metadata gathered from Illustrator *ai* files.

File size(kb)	Adobe Version	# of items inserted	Time stamps	Inserted element	Default name	Author	Creator tool
1005	CS3	2	3F7	D9580	2A1	D94A0	3B6
1060	CS3	3	3F7	E6E49	2B3	E6D64	3B6
2113	CS4	3	430	19E4FB	2B3	19E410	475

2) Illustrator template (ait)

Adobe illustrator template with an extension *ait* is another Illustrator file. The file type is a replica of the previous *ai* file type in a compact state. The metadata is similar to the the default *ai* file. Nevertheless the metadata is located at different offset address. The file still observes the pattern of embedding metadata in *xap* tags or *xmp* tags. The file contains metadata similar to *ai*but a different tagging for the default name which is *Title(xxxxxxxxxx)>>* tag or *Rdf:li* tags for the name of the file. Table 3 displays the metadata from the illustrator template file.

Table 3: Address offsets for Metadata gathered from Illustrator *ait* files.

File size(kb)	Adobe Version	# of items inserted	Time stamps	Inserted image	Default name	Author	Creator tool
295	CS3	2	409	27BFE	2B3	27B25	3C8
296	CS3	3	409	27D9A	2B3	27CBB	3C8
176	CS4	3	3FE	11EFD	267	11E13	443

2) Illustrator (eps)

Encapsulated Post Script (EPS) file is based on post script language. When created from an Illustrator application the file contains similar metadata to the metadata from *ai* and *ait*. However, *eps* does not record the inserted images in a bundle but in single entries. Time stamps do not include modification time. The creator tool is prefixed with a *%%Creator* tag and the time stamp for date of creation is prefixed with *%%CreationDate*: The default name of the created document is prefixed with *%%Title.* The difference between Illustrator *eps* and Photoshop *eps* is that Photoshop *eps* does not record inserted images and author names.Table 4 displays the metadata from the illustrator *eps* file.

Table 4: Address offsets for Metadata gathered from Illustrator *eps* files

File size (kb)	Adobe Version	# of items	Time stamps	Inserted element	Default name	Author	Creator tool
2994	CS3	2	32FA2	3B389	64	A5	32F61
3566	CS3	3	32FE3	3B3A9	64	A5	32F62
4960	CS4	2	BA	346EB	271C2B	271 C14	7F

3) Photoshop (psd)

Adobe Photoshop Image 12 file with extension *psd* is the default and only save type for Photoshop. Other file types are export types files. Unlike Illustrator files, Photoshop files consist mostly of metadata for document resolution, pixel data, color spacing, and document pixel dimensions. The files follow the pattern of *xap*, xmp tagging. The file type has the least essential metadata compared to other file types examined in this research. Table 5shows the metadata from the Photoshop files which are mostly time stamps.

Table 5: Address offsets for Metadata gathered from Illustrator *psd* files.

File size (kb)	Adobe Version	# of items inserted	Creation Time	Modification time	Metadata time	Creator tool
371	CS3	3	1BF	1F8	249	290
238	CS4	2	1F0	234	27A	1A9
237	CS5	2	331	375	3BB	3F6

4) Indesign (indd)

In-design *indd* is the default file type from Adobe In-design graphic design application. Essential metadata consist of a last url file metadata which can be a path visited either to acquire an image or to save a file. Usually records most metadata type at two different address offsets within the file. The tags used for In-design are different from Photoshop and Illustrator except for timestamps. For inserted images, entriesare prefixedas++@ or just a @ sign or the preferred *<stRef:lastURL>file:* tag. The inserted images are listed on a metadata entry respectively. Table 6 shows the metadata from the In-design *indd* files.

Table 6: Address offsets for Metadata gathered from In-design *indd* files..

File size (kb)	Adobe Version	# of items	Time stamps	Inserted element	Creator tool
848	CS3	1	CA886	C4895	CA948
1224	CS4	3	11A24E	C93DF	11A31F
368	CS5	2	4B20D	3A5A4	4B2DE

6) In-design template (indt)

Table 7: Address offsets for Metadata gathered from In-design *indt* files.

File size(kb)	Adobe Version	# of items inserted	Time stamps	Inserted elements	Creator tool
940	CS3	3	10E4C	1044C	10F0E
1140	CS4	1	1128B	C8CD0	1135C
408	CS5	3	5820D	3AD00	582DE

Adobe In-design template with an extension *indt* is another In-design file. The file type is a replica of the previous *indd* file type in a compact state with similar metadata and tagging but different offset address. The number in the bracket after the inserted image indicates the number of times that image was utilised. The address offsets shown in the metadata table are for the first recognised entries in the file. Table 7 shows the metadata from the In-design *indt* files.

7) In-design interexchange (incx)

Adobe In-design interchange is an In-design XML Interchange document. It maintains the similar formatting tags like the other In-design file types with last file *url* tags and bracketed access number. However it also retains the name of the document as it was saved in a different prefix *AsMt hDPT="rf_* .Table 8 shows the metadata from the In-design *incx* files.

Table 8: Address offsets for Metadata gathered from In-design incx files.

File size (kb)	Adobe Version	# of items inserted	Time stamps	Inserted elements	Default name	Creator tool
95	CS3	1	E8CA	E231	17901	E98C
99	CS3	2	EB05	E342	18759	EBC7
102	CS3	3	ED37	E342	194B9	EE04

8) In-design markup (idml)

Adobe In-design markup with extension *idml* is an XML Interchange document. Saved in smaller sizes of about less than a hundred kilobytes, for example a counterfeit passport can be roughly thirty kilobytes. The file type does not contain any discovered forensic information indicating creation of counterfeit documents.

9) Indesign Snippet (inds)

In-design Snippet files type with extension *inds* does not record the default name of the file and the author of the file.

Table 9: Address offsets for Metadata gathered from In-design snippet *inds* file.

File size (kb)	Adobe Version	# of items inserted	Time stamps	Inserted element	Creator tool
44	CS3	2	4D56	4A2C	4DD9
44	CS3	3	4D56	4A2C	4DD9

Inserted images are embedded in a different tag compared to other file types. The tag for inserted image is *<clnk lURL="k_" letg="rk_" laID="k_" lstk="re_lnsk" LnkI="x_c_c….* and which is only a single entry. Table 9 shows the address offsets for the metadata acquired from the *inds* files.

10) In-design markup Snippet (idms)

Indesign markup Snippet with extension *idms* contains metadata showing inserted images' embedded in a tag *<Link Self="ucf" AssetURL="$ID/"AssetID="$ID/" LinkResource URI ="file*. The files metadata contains all the inserted images unlike the previous *inds*. It also has metadata showing string of saving events for the file tagged as*<stEvt:when>2012-04-16T13:52:19+02:00 </stEvt:when>*. Table 10 shows address offsets for the described metadata.

Table 10: Address offsets for Metadata gathered from In-design markup snippet *idms*..

File size (kb)	Adobe Version	# of items inserted	Time stamps	Inserted .	Creator tool	String events
48	CS5	1	5A15	542F	59D7	BDF6
49	CS5	2	5C7C	508F	5C3E	C092
49	CS5	3	5C7C	508F	5C3E	C092

11) Incopy mark up document Snippet (icml)

In-copy markup document snippet *icml* does not contain metadata indicating the name of the author and the default name of the file. Inserted images are tagged in a different tagging which is *Link Self="ucf" AssetURL="$ID/" AssetID="$ID/" LinkResourceURI="*file as a single entry. Table 11 shows the address offsets for the metadata acquired from In-copy markup snippet *icml file*

Table 11: Address offsets for Metadata gathered from In-copy markup snippet *icml file*

File size(kb)	Adobe Version	# of items inserted	Time stamps	Inserted element	Creator tool	String events
66	CS5	1	A304	9D1E	A2C6	106E5

VI. Discussion

The research presented in this paper revealed that metadata is mostly dependant on the application that generated that particular file type. From the three graphic design applications in question Adobe Illustrator records the most essential digital evidence relating to the creation of counterfeit documents. Adobe Photoshop records the least metadata in its file types.

It is important to note that the graphic design file types examined in this research contain more valuable information concerning the creation of counterfeit documents than well known file types for example jpeg and bitmap.

In context, given that a digital forensic investigation was initiated on a suspected counterfeit document creation crime, and the

document was generated using a graphic design application. And using the identified digital forensic artifacts a digital forensic examiner is able to establish the utilised elements and to corroborate the gathered evidence. The discovered elements (e.g. fingerprint), together with the actual counterfeit document can be presented in the court for prosecution.

V. CONCLUSION

This approach is appreciated in addressing cases where document editing is largely associated with a particular application, only addressing case studies involving Adobe products but the same can be done for other graphic design applications. Recalling the problem that graphic design applications can be used to create counterfeit documents, the techniques discussed can be incorporated in bigger digital forensic tools like FTK and Encase or possibly the design of a crime specific tool similar to a porn detection stick created by Parabens software [15]. Also, future work can be conducted by carrying out this exercise on other graphic design applications like CorelDraw.

REFERENCES

[1] I.Rawoot, "*Terrorists favour 'easy' fake SA passports*", Mail and Guardian, 17 June 2011

[2] U.S National Institute of Justice, "Electronic Crime Scene Investigation Guide: A guide for First Responders". (2001)

[3] "*A roadmap for Digital Forensic Research*", Digital Forensic Research Workshop, (2001), pp 16.

[4] A.Jones. C.Valli, "*Building a digital forensic laboratory*", Burlington, Elsevier, (2008), pp 285.

[5] M. Solomon, D.Barett, N.Broom, "*Computer Forensics Jumpstart*", Sybex, London, (2005).pp 51

[6] E.Casey, "*Digital evidence and computer crime*", London, Academic Press, (2000), pp10.

[7] C.Altheide, H.Carvey. "*Digital Forensics with Open Source tools*". Elsevier. MA USA.(2011).pp 2

[8]]M.Zelkowitz, "*Advances in computers; information security*".Academic Press-Elsevier.(2009)

[9] Bloomberg News, "*Stocks weaken after Fed Statements*", The New York Times, 12 June 2011.

[10] Gartner Research, "*Which operating system will be 2011's bestseller*", (Accessed 11 August 2011).

[11] Top Tech News, "Windows 7, Office Drive Record Microsoft Revenue". (Accessed 23 July 2010)

[12] File-signatures,G.Kesler, http://www.garykessler.net/library/file_sigs.html, (Accessed 10 October 2011).

[13] Graphic design file format database, M.Reddy. http://www.martinreddy.net/gfx/2d-hi.html (Accessed 10 October 2011)

[14] H.Carvey, "*Windows Forensic Analysis Dvd Toolkit*", 2nd Ed, Elsevier. (2009), pp 296

[15] Porn detection stick, www.paraben-sticks.com/porn-detection-stick (Accessed9August2011)

[16] Adobe XMP. http://www.adobe.com/ products/ xmp/index.html (Accessed 11 November 2011

[17] Winhex, www.x-ways.net/forensics (Accessed 13 June 2011).

(2013d3)Analysing Registry, Log Files and Prefetch Files In Finding Digital Evidence In Graphic Design Applications

International Journal of
Information Security (ISecure)
Accepted February 2013

Enos K. Mabuto[1], H. S Venter[2]
Department of Computer Science
University of Pretoria, Pretoria, 0002, South Africa
Tel: +27 12 420 3654
Email: nasbutos@yahoo.co.uk[1], hsventer@cs.up.ac.za[2]

Abstract

The products of graphic design applications, leave behind traces of digital information which can be used during a digital forensic investigation in cases were counterfeit documents have been created.
This paper analyses the digital forensics involved in the creation of counterfeit documents. This is achieved by first recognizing the digital forensic artifacts left behind from the use of graphic design applications, and then analyzing the files associated with these applications. When analysing digital forensic artifacts generated by an application the specific focus is on determining whether the graphic design application was installed, whether the application was used, and determining whether an association can be made between the application's actions and such a digital crime. This is accomplished by locating such information from the registry, log files and prefetch files. The file analysis involves analysing files associated with these applications for file signatures and metadata.

In the end it becomes possible to determine if a system has been used for creating counterfeit documents or not.

Keywords

Digital evidence; Digital forensics; 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 creating unique art for company logos, magazine advertising or computer-aided design, to mention only a few. Most industries make use of graphic design applications for visual presentations using pictorial expressions that aid communication and expressing of ideas.

The use of forged documents, however, is noticed all over the world. A report by Ilham Rawoot of the Mail and Guardian newspaper stated that terrorists target fake South African passports because of the ease with which one can be faked [15]. A similar report from the International Business times was also reported [31]. These reports show that counterfeit documents are in circulation all over the world. The same graphic design applications used in the industry today can also be used for illegitimate purposes like creating counterfeit documents. The problem is that, with the editing and design capabilities of these graphic design applications, they can be used to create counterfeit documents like ID's, passports or drivers licenses. Criminal activities such as these necessitate need for digital forensic investigations.

The use of graphic design applications leaves behind traces that can be revealed during a digital forensic investigation. This paper identifies the digital traces left behind after using graphic design applications. In addition, a file analysis of files associated with these applications is conducted. To address the problem, the authors focus on the following three steps. First, the digital forensic information that shows whether the specific graphic design application was installed is identified. The second step entails querying whether the application was actually used for document editing. Lastly, it is determined whether an association can be made between the application's actions and such a digital crime. In so doing, an association with the potential

criminal may be achieved. However, it is not the focus of this paper to link the crime to an actual person. After gathering the traces left behind, the authors focus on an analysis of files associated with these applications. This involves determining the file signatures and recognizing the metadata related to these files.
The remainder of the paper is structured as follows. In the second section, some background of digital forensics is given, followed by a brief background on graphic design applications. The third section, which is the contributing section, is divided into three parts. The first part highlights the potential evidence which the authors refer to as digital forensic artifacts. Digital forensic artifacts can be found in graphic design applications where the source of the evidence is mainly system-generated. The source of potential evidence referred to above equates to the results of the registry analysis, application log file analysis and system prefetch file analysis. The second part is an examination of user-generated files and a highlight of the potential evidence. The source of potential evidence referred to being results from content identification and content examination of files utilized by graphic design applications. The authors also name the tools that can be used in aiding the analysis where applicable. The last part of the third section is a methodological description of how to acquire the evidence contained in the paper. The fourth section is an evaluation of the evidence that is extracted from the graphic design applications. Lastly a conclusion is given to end this paper.

2. Background

In the following section the authors provide some brief background literature on digital forensics including an explanation of digital evidence. A definition of digital forensic artifacts and a discussion on image forensics is found thereafter. The second section of the background consists of a very brief literature survey on graphic design applications.

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 unauthorized actions shown to be disruptive to planned operations[11]. To reconstruct and understand what has happened in the past on a system, data has to be gathered and analyzed in a transparent manner.

A digital forensic investigator can use the digital forensic process which is made up of steps including acquisition, examination, analysis and reporting [12].

The goal of a digital forensic investigation on a system is to find out what happened and who was responsible for a particular incident or crime. Digital forensic investigations focus on finding digital evidence after a computer or network security incident has occurred or locating data from systems that may form part of some litigation, even if it has been deleted. In this context, evidence is the most critical in any case. Therefore any items that can be considered to be of evidential value should be identified and collected [6].

2.1.1 Digital Evidence

Computer evidence or digital evidence is defined as any hardware, software or any data that can be used to prove one or more of the “who, what, when, where, why and how” of a security incident [2]. Computer evidence furthermore consists of digital files and their contents left behind after an incident. Casey defined digital evidence as any data that can be used to establish that a crime was committed or can prove a link between a crime and its victim or an offender [3]. Digital evidence consists entirely of sequences of binary values called bits [7]. It is important to note, however, that the evidence should be presented in its logical form in court or disciplinary hearing.
When investigating crime related to the use of an application the first question would typically be whether the particular application was installed, then whether the application was used and, lastly, whether there is any relationship between the actions of the application and the computer crime or incident being investigated. In responding to these queries, one or more of the “who, what, when, where, why and how” questions usually asked about a security incident has to be proven. Traces that are left behind from the use of an application or from an operating system can be referred to as digital forensic artifacts.

2.1.2 Digital Forensic Artifacts

An examiner reveals the truth of an event by discovering and exposing the remnants of the event that have been left on the system. These remnants are known as artifacts, which can be referred to as digital evidence [22]. However, due to the loaded legal connotations binding the term “evidence” the term “artifacts” is used instead. Evidence is referred to as something to be used during a legal proceeding. Artifacts are traces left behind due to activities and events, which may or may not be innocuous. Trying to remove these artifacts leaves other artifacts. For example, in trying to remove log files from a

system one has to use a removal tool, thus leaving additional traces that indicate 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 does not become evidence unless its ability to prove a fact has been established [9]. Therefore it is necessary to reconstruct events that occurred by gathering all the possible digital information from a system.

The work covered in this paper continues from previously-published work by the authors on "User-generated digital forensic evidence from graphic design applications" [28]. The mentioned paper elaborates on gathering potential evidence on the actual files with counterfeit value created by the counterfeiter intentionally. The potential evidence referred is described by use of evidence identifiers such tags and prefixes that embed the evidence.

As opposed to the previous paper [28], 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. Several file types are then compared with regards to the type of metadata they contain. Furthermore this paper describes how the identified artifacts can be linked to identify counterfeiting.

2.1.3 Image Forensics

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 [32,33].

Lien [32], 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 [33] 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 [34] 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 derivates of other statistical images to detect forgery.

These, [32,33,34], and other related work focus on determining forgery using statistical data within the image [35,36,37,38].

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.

In an investigation, 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

Many graphic design applications are currently available in the industry; however, Adobe Systems Incorporated is regarded as the largest software maker in the graphic design software category [1]. Adobe Systems Incorporated owns software technologies that are used for online transactions, business applications and social technologies [10].

Therefore, for this research, a case study was conducted with Adobe graphic design applications. The following are Adobe applications used for graphic design purposes.

2.2.1 Adobe Acrobat

Adobe Acrobat is an application used for viewing, creating, manipulating, printing and managing files in the portable document format (PDF).PDF files are usually read-only documents that cannot be altered without leaving an electronic footprint [19].

2.2.2 Adobe Photoshop

Adobe Photoshop is a professional industry-standard application for digital image editing and creation. Adobe Photoshop has an interactive platform to change the picture format, join pictures, split pictures, and change the color and appearance of photos among the many features it can offer.

2.2.3 Adobe In-Design

Adobe In-Design is a professional layout and design application that delivers production workflows, complex graphics and typography. Adobe In-Design is also used for designing magazines, printing page layouts and facilitating digital distribution using built in creative typography tools, to name a few.

2.2.4 Adobe Illustrator

Adobe Illustrator is an application used for vector artwork in planning projects. It has drawing tools and brushes that can be of use in designing graphic art consisting of rigid shapes and various line drawings, to mention only a few.

Any one of these applications can be used for document editing. Therefore it is necessary to conduct an exclusive examination for potential digital forensic evidence.

3. Digital forensic evidence in graphic design applications

Various experiments were carried out in order to search for pertinent evidence in graphic design applications. Experiments were conducted on Adobe applications capable of graphic designing namely Adobe Acrobat, Adobe Photoshop, Adobe In-Design and Adobe Illustrator. The experiments were conducted in two parts. The first part highlights digital forensic artifacts found in graphic design applications where the source of the potential evidence is mainly system-generated with results mostly from registry analysis, application log file analysis and system prefetch file analysis. The second part of the experiments, which involves examination of user-generated files, highlights results from file content identification and examination.

Early 2011 software reviews revealed that the Windows operating system is still ranked the most popular operating system [10, 13]. Therefore, the analysis for forensic artifacts was conducted on a Windows 7 platform.

To respond to the problem stated earlier, that graphic design applications can be used for creating counterfeit documents, firstly three techniques are used to gather digital forensic information related to graphic design applications. These techniques are the registry analysis, application log file analysis and system prefetch file analysis. From the experiments conducted it was recognised that an offender can deny any of the following; running the application, installing the application or using the application for counterfeiting. Therefore the analysis is formulated by asking three questions for each of the techniques listed above. The first question is can one identify digital forensic evidence that shows that the application was installed? Secondly, the question is asked, was the application actually used for document editing? The third question determines whether there is an association between the application action's and the alleged digital incident or crime. By following these queries an investigator is able to conduct an investigation in a uniform manner. For example, if the application was not installed, then there is no need to ascertain whether the application was used. Furthermore to respond to the same problem, a user-generated file analysis section follows, with two sub-sections dealing with content identification and content examination respectively. A summary of results is tabulated at the end of the section.

Experimental results gleaned from asking the three questions about registry analysis, application log file analysis and system prefetch file analysis are applied to each of the subsections to follow.

3.1. System-generated digital forensic artifacts from graphic design applications

"System generated digital forensic artifacts" refers to those artifacts created by the application without user intervention, while "user generated digital forensic artifacts refers" to artifacts created by the user intentionally.

For the experiments conducted, the following section describes the techniques used on Adobe graphic design applications. Three sub-sections follow in this section, namely registry analysis, application log file analysis and system prefetch file analysis.

3.1.1. Registry Analysis

The Windows registry is a collection of data files that stores vital configuration data for the system including user activity [16]. The Windows registry contains a plethora of valuable information including, user activity history, system configurations and information about installed applications. Potentially all the registry information can be of use to an analyst attempting to establish a timeline of activity on a system. Registry information is organized in the form of key entries. Registry information retrieved from different keys can be correlated for a better understanding. Besides the default regedit tool available in Windows systems, other tools

that can be used to analyze the registry are Registry Lite [17] and Registry Viewer [18].

An in-depth search was executed for keys associated with graphic design applications. It can also be noted that a single registry key can reveal more than one value. In establishing whether the application was installed, registry keys containing values for application settings, the installation time, installation date and the installation path for Adobe Acrobat can be obtained from key, *HKEY_CURRENT_USER\ Software\Adobe\ Adobe Acrobat\9.0\Installer* as shown in Fig 1 and for Adobe Photoshop it can be obtained from key *HKEY_LOCAL_MACHINE\ SOFTWARE\Adobe\ Photoshop\11.0\ApplicationPath.* Thus, if these keys are found in the registry, it answers the first question that the application was installed.

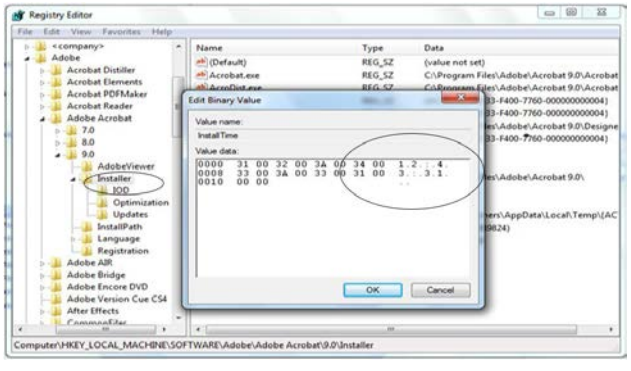

Fig 1. Registry view of Acrobat installation time

To query whether the application was actually used for document editing, values for the visited directories are acquired from the registry key, *HKEY_CURRENT_USER\Software\Adobe\Photoshop\11.0\VisitedDirs* and values for the home path of the application, as well as the name of the computer used to login (titled login server in registry) are obtained from the key, *HKEY_USERS\<user-id>\Volatile Environment.* These registry entries answer the second query that the application was actually used for document editing.

To query whether there is an association between the application's actions and a particular digital crime registry keys were obtained with values indicating the following: who used the application, the email address, the name of the department, the domain name and the name of the corporation. All these values are obtained from registry key *HKEY_CURRENT_USER \Software\Adobe\Adobe Acrobat\9.0\Identity* and similar values as above from *HKEY_CURRENT_ USER\Software\Adobe\Adobe Acrobat\9.0\ Security\ cMain.*

The registry keys contain a last used directory, which is created when the application is used. This establishes that the application was actually used to create a document.

In general, when a registry key is deleted, much like a file, it really does not disappear. In actual fact, when it is deleted, the size value is changed to a positive value [24]. In 2008, Jolanta Thomassen released a perl script known as "regslack" which uses this property to parse through a hive file which is the hierarchical file structure and retrieve deleted keys. It, therefore, comes to our attention that when an offender has deleted these keys a digital forensic investigator is able to retrieve the keys.

3.1.2. Application Log File Analysis

Application log files are files related to events from a particular application. Besides these, Windows also maintains system log files of events and actions that can be essential to an investigation. System log files contain important information about recently viewed documents, saved data, personal user information and other temporary data files. The focus of this paper is on log files created by the graphic design applications in question. Winhex from XWays [20] is used as the hex editor for analyzing data files, but any other hex editor can also be used.

In establishing whether the application was installed (the first query), a folder is created in the following path *C:\Users\<user>\AppData\Roaming \Adobe*. The time stamp on the folder denotes the date of installation. It should be noted that the "Appdata" folder is hidden by default.

To query whether the application was actually used for document editing (the second query), a history of viewed documents, history of file searches and other temporary files are obtained from the following location *C:\users\<user>\appdata\local\microsoft\ windows\history\<week or day>\computer.* This location contains the actual files saved after document editing, for example a **.psd* file saved from Adobe Photoshop.

To query whether an association exists between the application's actions and a digital crime a log file titled *InDesignSavedData* in the location *C:\Users\<user>\AppData\Local\Adobe\InDesign\Version 6.0\en_GB\Caches* provides an answer. The file contains data indicating which actions were taken during document editing like alignment, clearing text, moving an object, joining, importing files all starting at hex offset C544 and the location of any imported files at hex offset 10D7F7 as illustrated in Fig 2. An imported file can be any file for example a fingerprint photo attached to the file being created. Fig 2 illustrates the file location of imported files used during document editing and alignment actions carried out as an example. The location *C:\Users \<user>\AppData\Roaming\ Adobe\ Acrobat\ 9.0\Security* contains a file titled *shared data events,*

which shows that a digital signature was created with values supplied for email, department, corporation and name of user. This information helps a digital forensic investigator to establish a possible link to the criminal.

To further explain Fig 2; the first column is the byte count, also known as the byte offset, in base sixteen (hex). The proceeding paired columns are the hexadecimal representation of the file content. Each column represents two bytes. The last column on the far right represents the ASCII text rendition of the file. Non printable or non ASCII characters are displayed as dots as seen in the last column.

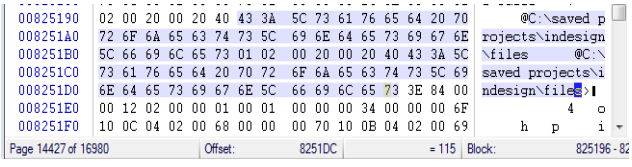

Fig 2: log file hex editor extract indicating location of imported files and an alignment action

Also the location *C:\Users\<user>\AppData\Roaming\Adobe\Adobe Illustrator CS4 Settings* contains a *ins* file extension titled *Recently used optimizations* which contains the format last used for document editing and the previous changes made to file type. The location *C:\Users\<user>\ AppData\Roaming\ Adobe\ Adobe Photoshop CS4\Adobe Photoshop CS4 Settings* contains a *Actionspallette.psp* file containing information relating to saving actions that took place during document editing. It also contains information about the file extension used to save documents and any messages displayed while saving. The *Prefs.psp* file in the same location contains objects used for editing like brushes used, shapes used, and the recent file location at offset 31AFE.

Application log files record information about the documents created, their names and if any images or objects have been used to create these documents. The list of the created documents can then be used to search for the potential counterfeit documents created. Furthermore the inserted images can be used to identify if indeed the created document is counterfeit or not. By analysing them individually thereby determining if it's a human face, fingerprint or barcode that was inserted into the document.

3.1.3. System Prefetch file Analysis

Prefetching was developed to improve the systems performance [14]. The purpose of prefetching is to allow regularly used applications to load faster by prestaging segments of loaded code in a specific location so that instead of searching for it (resulting in page faults), the operating system knows exactly where it is. It means when an analyst finds a prefetch file for a particular application, it indicates that the particular application was indeed run on the system. The creation date of that file will indicate when the application was first run, although assuming that a previous prefetch file wasn't deleted and a new one created in its place. This is because prefetch files are actually temporary files that can be deleted or overwritten by the operating system at any time. The prefetch file contains a 64 bit time stamp indicating when it was last run, as well as a count of how many times it was run. On Windows 7, the 64 bit last run time stamp is at offset 80 (128 bytes) within the binary contents of the prefetch file and the run count 4 bytes at offset 98 (156 bytes) as illustrated in Fig 3.

Once the data is processed, it is written to a **.pf* file in the systems prefetch directory. The **.pf* file will be referenced later when the program is run again. The file name is created using the application's name followed by a dash and then by a hexadecimal representation of the hash of the path of the application for example *ACROBAT_SL.EXE DC4293F2.pf*. That means the same program run from different locations will create different *.pf* files. In this way, the next time an application is launched, the prefetch directory is checked for a prefetch file. If it exists, the code within the **.pf* file is used to launch the application. If, however, the prefetch file is not present the application will still be launched but will load slowly.

Prefetch files are located in the folder: *%sytemroot%prefetch*. It should also be noted that one needs administrative privileges to access the prefetch folder. Within the prefetch file are values that correspond to the number of times the application was launched and a value containing the last time the application was launched. This information is obtained from analyzing the prefetch file with a hex editor as illustrated in Fig 3.

Therefore, prefetch files establish that the application was installed and that the application was used indicated by last run time and run count respectively. However, there is no established relationship between the application's actions and the digital crime in the prefetch files but the information found can be correlated to information gathered from the registry and log files. The operating system generates several different prefetch files. It is necessary for an investigator to know all prefetch files generated, for in some cases the name of the prefetch file will not be similar to the name of the application. Table 1 shows Adobe prefetch files that are obtained from *%systemroot%prefetch*.

Table 1: Adobe prefetch files

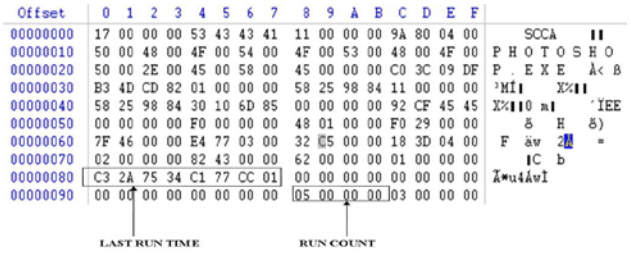

Fig 3: Hex editor extract of Adobe Photoshop prefetch file.

It should also be noted that any deleted log files or prefetch files could be recovered using any popular forensic tool like FTK and Encase.

3.2 User generated artifacts from file examination

In order to conduct an exclusive examination on a crime conducted from an application the investigator has to understand the nature of the files that are generated from that particular application, in this case, graphic design applications. This is so that digital forensic examiners are able to uncover and exploit any digital forensic artifacts present in the identified files [22].

As previously stated, user-generated digital forensic artifacts refers to files created by the user intentionally. User generated file artifacts are divided into two distinct categories, which are, content identification and content examination. Content identification is the process of determining or verifying the type of a specific file. Content examination is the retrieval of any embedded metadata that may be present in a given file.

File type	File extension	ASCI II	File signature
In-design	indd	íõØFå½1ïçþt·DOCUMENTp	06 06 ED F5 D8 4D 46 E5 BD 31 EF E7 FE 74 B7 1D 44 4F 43 55 4D 45 4E 54 01 70 0F
In-design XML Interchange document	incx	<?xml version="1.0" encoding="UTF-8" standalone ="yes"?>	3C3F786D6C2076657273696F6E3D22312E302220656E636F64696E673D225554462D382220737461 6E64616C6F6E653D22796573223F3E
In-design template	indt	íõØFå½1ïçþt·DOCUMENTp	06 06 ED F5 D8 4D 46 E5 BD 31 EF E7 FE 74 B7 1D 44 4F 43 55 4D 45 4E 54 01 70 0F
Photoshop	psd	8BPS	38 42 50 53 00 01
Illustrator file	ai	%PDF-1.5% âãÏÓ1 0 obj	255044462D312E350D25E2E3CFD30D0A312030206F626A
Illustrator template	ait	%PDF-1.5% âãÏÓ1 0 obj	255044462D312E350D25E2E3CFD30D0A312030206F626A
Encapsulated post script	eps	ÅÐÓÆ	C5 D0 D3 C6

In the case of the examination of counterfeit documents

Application Name	File Name
Adobe Acrobat	*ACROBAT_SL.EXE DC4293F2.pf*
Adobe Distributor	*ACRODIST.EXE1C2D8F2D.pf*
Adobe Reader	*ACRORD32.EXE DE3ACC1.pf*
Adobe Collaboration	*ADOBECOLLABSYNC.EXE621E7FA.pf*
Adobe Updater	*ADOBEUPDATER.EXE9AAD898.pf*
Adobe Service manager	*CSXSERVICEMANAGER.EXE B80CD935.pf*
Adobe Indesign	*INDESIGN.EXE C8D4FD6C.pf*
Adobe Tray	*VERSIONCUECS4TRAY.EXE D4DE4E1A.pf*
Adobe Photoshop	*PHOTOSHOP.EXE 4545CF92.pf*

the digital forensic investigator might need to identify potential changes inside files consistently, for example, the involvement of a fingerprints, barcodes or human faces embedded inside graphic design application file formats. The four graphic design applications discussed above are associated with more than thirty nine file types. However, for this research the authors focus was only on file types that are specific to the four graphic design applications, thus ignoring well-known file types like jpeg, bitmap, tag, tiff, tga etc. Gary Kesler and Martin Reddy keep a list of these common file signatures online, which is a continuing work in progress database [25, 26]. An online metadata extraction tool is also available for extracting metadata from these common file types [30].

3.2.1 Content identification

As previously stated content identification involves verifying the identity of a file extension. An offender can alter the file extension of a particular file in order to promote ambiguity. Therefore there is need to identify a files integrity by file signature analysis. An investigator needs to know what a particular file type is. A file is normally analyzed within its first bytes to determine the specific signature [14]. The file signature is therefore located at specific offsets usually in the beginning of a file.

It can be noted from the research conducted that known digital forensic tools like FTK can detect various file types but not for graphic design applications discussed in this paper. For example, digital forensic tools can verify file types like tga, bmp, gif, tif and png amongst others, but not the file types of graphic design applications as discussed in this paper.

The analysis to determine a graphic design file signature was also conducted using a hex editor. These values are generally hexadecimal values. Table 2 contains the list of

file signatures identified and specific to the graphic design applications previously discussed. The file type in Table 2 represents the named form of the particular graphic design file. Proof of the real file content resides within the content of the file, usually known as the file signature. The file extension is merely a suffix that represents the encoding of a file's content, usually three or four characters separated by a dot from the file name. However, the file extension should never be trusted as it can be renamed to anything else. One should rather focus on the file signature to determine the correct file type. The ASCII column in Table 2 represents the entry in text-readable format. The file signature columns represent the entry in hexadecimal format. Both these entries appear exactly as shown in the hex editor. The digital forensic examiner can use the information in Table 2 to identify the particular files for the graphic design applications in question.

Table 2: Graphic design file signatures

3.2.2 Content Examination

Content examination involves determining the metadata of files, in this case, graphic design application file types. Metadata refers to data about data [22]. On Windows systems this includes modified, accessed and creation times .The same hex editors, as previously stated, are used to examine the content of files associated with graphic design applications. Metadata is essential during an investigation as this reveals what useful information can be extracted from a particular file, for example this can be time stamps or name of the user who created the file. Table 3 shows the metadata acquired from graphic design file types. The offset is the address pointer of the described metadata. In other words, if an investigator searched for a certain offset, the hex editor would skip to the particular metadata. However, several experiments reveal that the offset can slightly differ by plus or minus 780 bytes per metadata, which is usually in the same page view depending on the size of the file and quantity of metadata present in the file. Therefore the tabulated values can still be used on graphic design files of different sizes. The metadata is embedded in Extensible Metadata Platform (xmp) tags, which is Adobe's way of embedding metadata in its various file types [27].

Table 3: Graphic design file types related metadata

File type	File extension	Description of Metadata	Offset (Address pointer to Metadata)	Example of the Metadata (As presented in a hex editor)
Indesign document	indd	File location for any imported image files	D9EB	file:C:/Users/<username>\/Pictures/ dvd%20picture%20sleeves/Capture_005% 20%282%29.JPG
		Name of application that created the file	E510B or E6E16	<stEvt:softwareAgent>Adobe InDesign 6.0</stEvt:softwareAgent>
		String events of saving history	F0D0C to F12FE	<stEvt:action>created</stEvt:action> <stEvt:when>2011-05-04T15:13:25+02:00</stEvt:when>stEvt:action>saved</stEvt:action><stEvt:when>2011-05-04T15:15:43+02:00</stEvt:when>
		Date file was created	F5263	CreateDate>2011-05-04T15:13:25+02:00
		Metadata Date	F52A7	MetadataDate>2011-05-04T15:18:24+02:00</xmp:MetadataDate
		Modify Date	FD2EA	<xmp:ModifyDate>2011-05-04T15:18:24+02:00</xmp:ModifyDate>
Illustrator Postscript file	eps	Name of application that created the file	57	%%Creator: Adobe Illustrator(R) 14.0
		Date file was created	8E	%CreationDate: 9/17/2011
		Login name of user that created the file	73	%%For: <username>\ %
Illustrator file	ai	Metadata Date	3A7	<xmp:MetadataDate>2011-05-04T15:51:17+02:00</xmp:MetadataDate>
		Date file was modified	3ED	<xmp:ModifyDate>2011-05-04T15:51:17+02:00</xmp:ModifyDate>
		Date file was created	431	<xmp:CreateDate>2011-05-04T15:51:17+02:00</xmp:CreateDate>
		Name of application that created the file	476	<xmp:CreatorTool>Adobe Illustrator CSX</xmp:CreatorTool>
Photoshop file	psd	Name of application that created the file	1A9	<xmp:CreatorTool>Adobe Photoshop CSX Windows</xmp:CreatorTool>
		Date file was created	1F0	<xmp:CreateDate>2011-05-04T14:39:08+02:00</xmp:CreateDate>
		Date file was modified	234	<xmp:ModifyDate>2011-05-04T14:50:23+02:00</xmp:ModifyDate>
		Metadata date	27A	<xmp:MetadataDate>2011-05-04T14:50:23+02:00</xmp:MetadataDate>
		String events of saving history	6FF to 717	<stEvt:instanceID>xmp.iid:DE0657134D76E011B00EFDC555D228CB</stEvt:instanceID> <stEvt:when>2011-05-04T14:50:23+02:00</stEvt:when>
Illustrator template	ait	Name of application that created the file	1F3 or 452	<xmp:CreatorTool>Adobe Illustrator CSX</xmp:CreatorTool>
		Metadata Date	383	<xmp:MetadataDate>2011-05-04T15:51:17+02:00</xmp:MetadataDate>
		Date file was modified	3C9 or 16323	<xmp:ModifyDate>2011-05-04T15:51:17+02:00</xmp:ModifyDate>
		Date file was created	40D	<xmp:CreateDate>2011-05-04T15:51:17+02:00</xmp:CreateDate>
		String events of saving history	D02B or D5D3	<stEvt:action>saved</stEvt:action> <stEvt:instanceID>xmp.iid:FF7F117407206811B628E3BF27C8C41B</stEvt:instanceID> <stEvt:when>2011-05-22T16:23:53-07:00</stEvt:when>
		Name of user that created the file	17FB9	%%For: (Pinchers) ()
		File path for any imported images	D727	%%DocumentFiles:C:\Users\<username>\Pictures\Sizzla-Soul Deep-Front.jpg %%+C:\Users\<username>\\Pictures\Tulips.jpg
		List of previous files names used	180A8	/Title(illustrator .ait template)

Table 3: Graphic design file types related metadata (continued)

File type	File extension	Description of Metadata	Offset (Address pointer to Metadata)	Example of the Metadata (As presented in a hex editor)
Indesign template file	indt	File path for any imported images	CF1E0 or D4F03	%%DocumentFiles:C:\Users\<username>\\Pictures\Sizzla-Soul Deep-Front.jpg %%+C:\Users\<username>\Pictures\Tulips.jpg
		Date file was created	D72AB	<xmp:CreateDate>2011-05-04T15:17:21+02:00</xmp:CreateDate>
		Metadata Date	D72F1	<xmp:MetadataDate>2011-05-04T15:17:21+02:00</xmp:MetadataDate>
		String events of saving history	D3DBA to D3F46	<stEvt:instanceID>xmp.iid:972E234B5076E011AAFBC6ED1F893037</stEvt:instanceID> <stEvt:when>2011-05-04T15:17:21+02:00</stEvt:when>
		Name of application that created the file	D400C or D737C	<xmp:CreatorTool>Adobe InDesign 6.0>
Indesign interexchange file	incx	Date file was created	BFD3	<xmp:CreatorTool>Adobe InDesign 6.0</xmp:CreatorTool>
		Metadata Date	C019	<xmp:MetadataDate>2011-05-04T15:17:21+02:00</xmp:MetadataDate>
		Date file was modified	C05F	<xmp:ModifyDate>2011-05-04T15:17:21+02:00</xmp:ModifyDate>
		Date file was created	BD3A	<xmp:CreateDate>2011-05-04T15:17:21+02:00</xmp:CreateDate>
		Name of application that created the file	C0A4	<xmp:CreatorTool>Adobe InDesign 6.0
		String events of saving history	108C2 or 115F7	<stEvt:instanceID>xmp.iid:972E234B5076E011AAFBC6ED1F893037</stEvt:instanceID><stEvt:when>2011-05-04T15:17:21+02:00</stEvt:when>
		Last file path used	119D8 or 11C4d	%%DocumentFiles:C:\Users\<username>\\Pictures\Sizzla-Soul Deep-Front.jpg %%+C:\Users\<username>\\Pictures\Tulips.jpg
		Previous file format used	15BD2	<xmpGImg:format>JPEG</xmpGImg:format>

3.3 Methods to gather digital evidence

Digital forensic investigators should be able to identify digital evidence from graphic design applications and interpret the evidence appropriately. In the sub-sections that follow, the authors describe a method to identify the evidence presented in this paper.

3.3.1 Examine the system

As discussed in section 3.1, an investigator has to recognise digital evidence from the system. This enables one to identify the particular graphic design application installed on a system using the registry and prefetch files. The identified graphic design applications can then be examined for log files embedded within the system. The log files are examined to recognise the documents that were created by that application. Recognising the particular graphic design application also enables one to be able to recognise the file types associated with the application. The files types referred to in this case being user-generated artifacts discussed in this paper.

3.3.2 Examine file types

As discussed in section 3.2, an investigator next task would be to identify all the file types associated with the graphic design application. For example, *psd*, *indd*, *ait*, *inx* file types from Adobe graphic design applications. The identified file types are examined for file signatures as described in section 3.2.1 content identification. After the files signatures are noted, the examination continues to determine the contents of the graphic design file types as described in section 3.2.2.

3.3.3 Co-relate the evidence

The final task for an investigator would be to identify the artifacts obtained from the system and from the file types. This includes determining the names of the counterfeit documents obtained from system generated artifacts. The names can then be searched for using any application or operating system. The last task would be to view these created documents using any image viewers or any application capable of viewing graphic images to visualise the products of graphic design applications. In the end an investigator would be able to tie the evidence and recognise if the documents produced are counterfeit or not.

3.4. Summary

The analysis for digital forensic artifacts can be summarized in Table 4. To briefly explain the table, only one technique is discussed in detail. The remainder of the table can be read in a similar fashion. From the second technique (Log file analysis) the query “was the application installed” (indicated by “installed” in the “Query” column in Table 4) comprised of an artifact consisting of a folder with temporary files created from the application. The query “was the application used” (indicated by “used”) included a cache list consisting of saved data actions made during document editing. For the same technique the query of “establishing an association with the crime concerned” (indicated by “Link”) reflected a file relating to a security policy file and the name of the user. The remainder of the results is self explanatory in Table 4.

Table 4: Summary of gathered digital forensic artifacts

Technique	Query	Artifact type	Details of contents
Registry analysis	Installed	Key	Path, time, date
	Used	Key	Visited directory
	Link	Key	Epic name, server name
Log file analysis	Installed	Folder	Temporary files
	Used	Cache list	Saved data
	Link	File	Security policy name
Prefetch file analysis	Installed	File	Program name
	Used	File	Hash of path location

For user-generated file analysis all graphic design application file types analysed have timestamps as part of their metadata. However only a few of them have the user name of the creator of the file as part of the metadata. Table 5 summarises the user-generated file types. “Yes” indicates that the described metadata is present and “No” denotes that the file type does not contain the described metadata. The headings of the columns are brief names of descriptions of the metadata tabulated in Table 3.

Table 5: Summary of user-generated file analysis

File format extension	Date of creation	Date of modification	Metadata date	Creator username	Creator tool	Location of importations	String events
indd	Yes	Yes	Yes	No	Yes	Yes	Yes
indt	Yes	Yes	Yes	No	Yes	Yes	Yes
incx	Yes	Yes	Yes	No	Yes	Yes	Yes
ai	Yes	Yes	Yes	No	Yes	No	No
ait	Yes	Yes	Yes	Yes	Yes	No	Yes
psd	Yes	Yes	Yes	No	Yes	No	Yes
eps	Yes	No	No	Yes	Yes	No	No

4. Discussion

The objective of the paper is to determine if a system was used for counterfeiting. However, based on possible offender deniability the questions are formulated to respond to such circumstances.

If it is recognised that the application was not installed, it becomes possible that another computer system was used to create the documents. From analysing the log files, such information can be derived from the counterfeit document itself, this is the log in name on the computer, which is obtained by analysing the suspect counterfeit document illustrated in table 2. This can lead to identifying the name of the system that the counterfeit documents were created on.

An application can be uninstalled after editing counterfeit documents. The registry entries illustrated in this paper are under normal circumstances left behind after installation and un-installation. If however the offender has used some tool or has manually deleted these entries, an investigator can use a tool called "reg-slack"[24], which is used to recover deleted registry entries.

Furthermore, other tools can be obtained to clean registry entries. It is thereby important to mention that the fight between forensics and anti-forensics is beyond the scope of this paper. The papers objective is to present work for digital forensic investigators to be able to find and interpret evidence related to document counterfeiting.

Recalling that computer evidence is defined as any hardware, software or any data that can be used to prove one or more of the "who, what, when, where, why and how" of a security incident. The registry analysis proves the "who, when, where and how" of the digital evidence definition. The registry analysis also answers all three queries: (1) was the application installed, (2) was the application actually used, and (3) is there any link to the digital crime? Application log files prove the "where, who, and when" of a piece of digital evidence and respond to all three queries. Prefetch files prove the "when and how" part and answer the queries; was the application installed and was the application used? By following these three queries an investigator is able to conduct an investigation in a step-by-step uniform manner.

For content identification, the digital forensic investigator can use the recognized file signatures and the corresponding ASCII text representation to determine the file type of the graphic design applications in question. The file signatures can also be used when searching files from a formatted hard drive. Also an in-depth analysis of user-generated files can assist an investigator in knowing which particular metadata to acquire from graphic design file types and at what offset address.

By reviewing all the artifacts gathered the definition of digital evidence can be confirmed. This is so because all the six questions, "who, what, when, where, why and how" of the digital evidence definition are validated from the results acquired. Briefly clarifying the results: the "who" was specified by an artifact with the user name, the "what", specified by identifying the particular files types from the application, the "when", specified with a registry artifact indicating time of incident, the "where" specified with an artifact showing the file location, the "why" specified with a file metadata extraction revealing the file contents and the "how" with an artifact indicating which application was used for document editing. These results are essential for a digital forensic investigator to know where to look for digital forensic information, guided by knowing what information to find at a named particular location. This speeds up the process of an investigation where graphic design applications were used.

This approach is appreciated in addressing cases where document editing is largely associated with a particular application. The approach only addresses case studies involving Adobe products but the same can be done for similar graphic design applications. However, the approach doesn't tackle issues where the user only edits a hard copy, scans and prints without using any pre-installed application. The techniques discussed can be incorporated in bigger digital forensic tools like FTK and Encase or possibly the design of a crime specific tool similar to a porn detection stick created by Parabens software [21], which is a thumb drive device that will scan and detect pornographic content on a computer.

5. Conclusion

Registry keys, log files and prefetch files each reveal information that can be of digital forensic value. All this digital information can be correlated to constitute the digital evidence related to graphic design applications. Overall the three queries - was the application installed, was the application used, and is there any link between the crimes being investigated - have been responded to. By responding to all the three queries, the investigator eliminates doubts about whether an application was installed or used before establishing a possible link to the crime in question.

More-over, it is possible for a digital forensic investigator to conduct an in-depth analysis of files generated from graphic design applications. For user generated file examination the investigator is able to verify the identity of a file type through content identification using file signatures. Also an investigator is able to know which metadata can be extracted from user generated files from graphic design applications.

Revisiting the problem "graphic design applications can be used to create fraudulent documents" and having acquired the necessary digital forensic artifacts, a digital forensic investigator is able to deduce activities associated with the creating of fraudulent documents.

The experiments were conducted using the most used graphic design applications, so that the evidence illustrated can be of use to most digital forensic investigations. The work presented is suitable in cases were digital document counterfeiting has been exercised. The work does not cover cases in which hard copy documents have been counterfeited.

Aside from the five techniques, registry analysis, application log file analysis, system prefetch analysis, content identification (signature verification) and content examination (metadata extraction) discussed above, more techniques can be tested for future work to gather digital forensic information

related to the use of graphic design applications. The work contained in this paper can be incorporated into OpenCV [29] for use in detecting inserted images for example fingerprints, bar codes in counterfeit documents. Also, future work can be conducted by carrying out this exercise on other graphic design applications.

6. REFERENCES

[1] Bloomberg News, "Stocks weaken after Fed Statements, The New York Times", 12 June 2011.

[2] M. G Solomon, D. Barett, and N. Broom, Computer Forensics Jumpstart, Sybex, London, 2005.pp. 51.

[3] E. Casey, Digital evidence and computer crime, London, Academic Press, 2000, pp. 10.

[4] Gartner Research, "Which operating system will be 2011's bestseller", Accessed 11 August 2011.

[5] D. Jones, "Adobe 2Q Net Up 54% On Broad Sales Gains, Higher Margins", The Wall Street Journal, Accessed 21 June 2011.

[6] A. Jones and C. Valli, Building a digital forensic laboratory, Burlington, Elsevier, 2008, pp. 285.

[7] F. Cohan, "Towards a science of digital forensic investigation", IFIP Advances Digital Forensics VI, China, 2010, pp. 17-35.

[8] P. Jones, Practical forensic digital imaging: application and techniques, CRC press, Indiana, 2009, pp 147-157.

[9] M. V. Zelkowitz, Advances in computers; information security.Academic Press-Elsevier, 2009.

[10] Tech Specs, www.adobe.com, Accessed 22 June 2011.

[11] "A roadmap for Digital Forensic Research", Digital Forensic Research Workshop, 2001, pp 16.

[12] U.S National Institute of Justice, Electronic Crime Scene Investigation Guide: A guide for First Responders, 2001.

[13] Top Tech News, "Windows 7, Office Drive Record Microsoft Revenue". Accessed 23 July 2010.

[14] H. Carvey, Windows Forensic Analysis Dvd Toolkit, 2nd Ed, Elsevier. 2009, pp.296.

[15] I. Rawoot, "Terrorists favour 'easy' fake SA passports", Mail and Guardian, 17 June 2011.

[16] H. Carvey, *Windows Registry Analysis*, 2nd Ed, Elsevier. 2009, pp 194.

[17] Reglite software, www.resplendence.com/reglite Accessed, 14-July-2011.

[18] Regview, www.accessdat.com/support, Accessed 14 July-2011.

[19] T. Padova, Adobe Acrobat 9 PDF Bible" Indianapolis, Wiley, 2008.

[20] Winhex, www.x-ways.net/forensics, Accessed 13 June 2011.

[21] Porn detection stick, www.paraben-sticks. com/porn-detection-stick, Accessed 9 August 2011.

[22] C. Altheide and H. Carvey, Digital Forensics with Open Source tools, Elsevier, MA USA.2011.pp 2

[23] PDF recovery, "Advanced PDF Password Recovery Bundle", www.elcomsoft.com/products, Accessed 2 September 2011.

[24] Regslack, Downloads, www.regripper.net, Accessed September 2011.

[25] G. Kesler, File signatures, http://www. garykessler.net/library/file_sigs.html, Accessed 19 December, 2012.

[26] M. Reddy, Graphic design file format database, http://www.martinreddy.net/gfx/2d-hi.html Accessed 19 December, 2012.

[27] Adobe XMP. http://www.adobe.com/ products/ xmp/index.html, Accessed 19 December 2012.

[28] E. K. Mabuto and H. S. Venter, "User –generated evidence from graphic design applications", International conference on cyber security, cyber warfare and digital forensics, CyberSec2012, pp. 195-200.

[29] Open Source Computer Vision, (OpenCV), www.opencv.org, Accessed 11 September 2012.

[30] Metadata Extraction Tool, www.extractmetadata.com, Accessed 11 July 2012.

[31] J. Bargas, "Brazilian man attempted to open a bank account using a fake jack Nicholson ID", International Business Times, http://au.ibtimes.com/, 2 March, 2012.

[32] C. C. Lien, "Fast forgery detection with the intrinsic resampling properties", Journal of information security, Vol 1 no1, 2010, pp. 11-22.

[33] M. C. Stamm. "Forensic detection of image tampering using intrinsic statistical fingerprints in histograms". APSIPA Annual summit and conference, Japan 2009, pp. 563-572.

[34] K. Cohen. "Digital Still Camera Forensics", Small scale digital device forensics Journal, vol1, no 1, June 2007. pp. 2-8.

[35] H. Farid. "Image forgery detection", IEEE Signal Processing Magn, 2009, pp. 16-25.

[36] S. Bayram, I. Avcibas, B. Sankur and N. Memon, "Image manipulation detection," J. Electron. Imaging, vol. 15, no. 4, pp. 41-52, 2006.

[37] J. Wang, "Image forensics based on manual blurred edge detection", Multimedia information networking and security (MINES), 2010, pp. 907-911.

[38] N. Memon, "Photo Forensics". International workshop on information security, NYU, 2012.pp. 1-27.

(2012d4) SYSTEM-GENERATED DIGITAL FORENSIC EVIDENCE IN GRAPHIC DESIGN APPLICATIONS

The 2013 **ADFSL** Conference on. Digital Forensics, Security and Law. June 10-12, 2013. Richmond, Virginia USA

Enos K. Mabuto[1], H. S Venter[2]
Department of Computer Science
University of Pretoria, Pretoria, 0002, South Africa
Tel: +27 12 420 3654
Email: nasbutos@yahoo.co.uk[1], hsventer@cs.up.ac.za[2]

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.

II. 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,* terrorists 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.

III. BACKGROUND

In part 2.1, the authors discuss the studied literature on digital forensics, followed by an explanation of digital evidence and a definition of digital forensic artifacts. Part 2.2 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 of graphic design software 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 derivates 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! (Factorial)= 24) on eleven examined file types. The combination for all file types equalled two hundred and sixty four (24 x 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 scanner's 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	*Johnstone_passport_final_edit.psd*
Name of printer	shd	0X88	*HP Laserjet 2605_2605dnPCL*
Name of printer (repeat)	shd	0X3B0	*HP Laserjet 2605_2605dnPCL*
Name of the application that generated the print request	shd	0X2120	*Adobe Photoshop CS5*
Username and name of file	shd	0X2400	*Robert_graphics_editor. Johnstone_passport_final_edit.psd*

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.

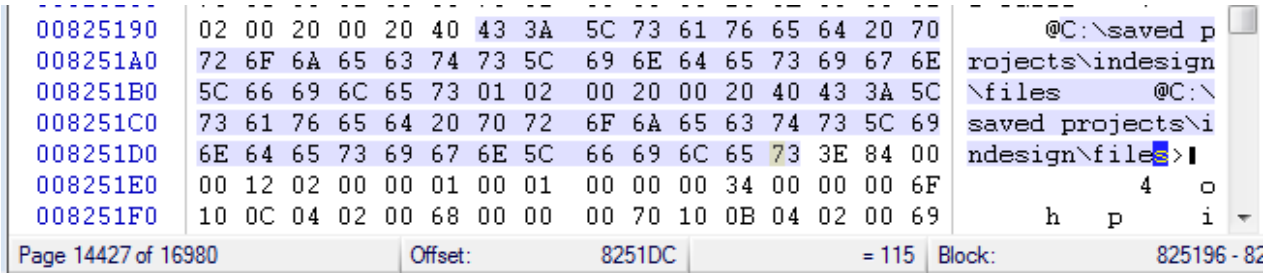

Figure 1: graphic design application log file containing 16980 pages

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 guides 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 a radar chart** in order to pinpoint where the evidence can be found within the log file. A radar chart is a graphical chart used to illustrate the distribution of data in a circular form. Figure 2 illustrates the distribution of the digital forensic artifacts within the Photoshop *psp* log file.

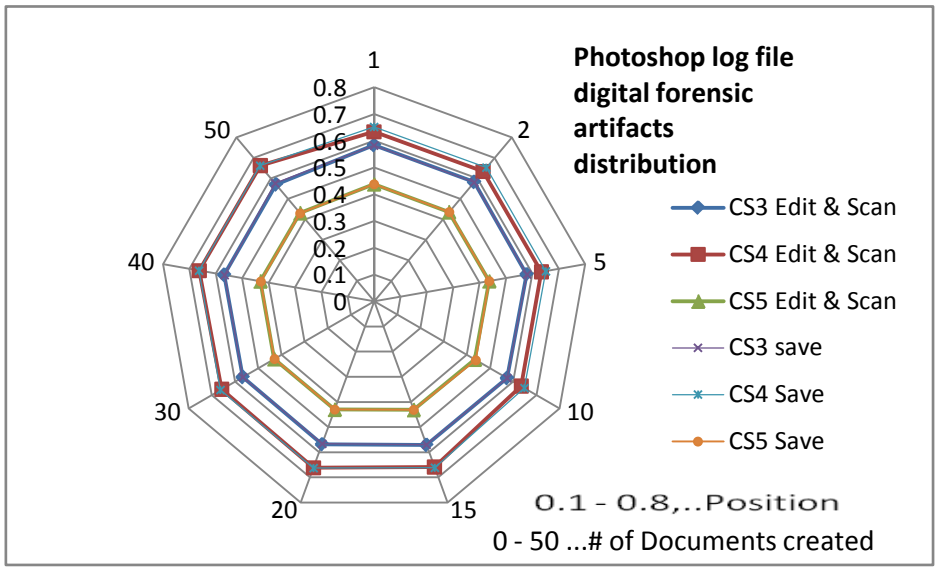

Figure 2: A graphic illustration of digital artifacts distribution in a Photoshop log file

The radar chart in Figure 2 shows that the digital forensic artifacts are located mostly in the middle of the log file for any action. If more than one document was created, the artifacts would include a list of the created documents, also in the middle of the log file. This region is

determined proportionally to the size of the log file. For example, the middle of a log file sized 1.5 MB would be 16E360 (1 500 000 bytes in hex number)/ 2 = B71B0. The result would be the address offset of the middle of the log file. The following equation can be used.

Size of log file (convert to hex number)/2 = hexadecimal Address offset of artifacts

In this chart (Figure 2), 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*.

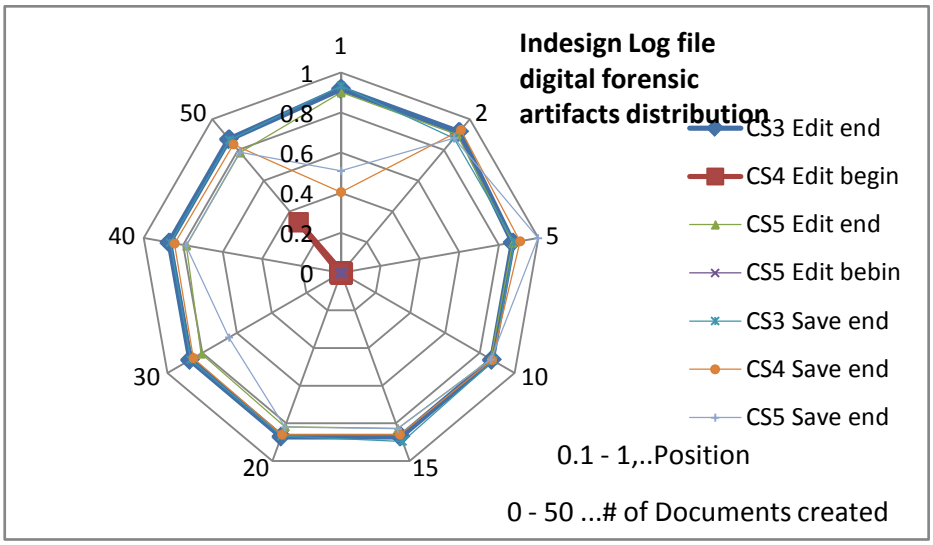

Figure 3: A graphic illustration of digital artifacts distribution in an Adobe In-Design log file

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. With respect to 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, can 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. 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. A similar counterfeit detection tool can be used in detecting how and who counterfeited a document or determining which system was used to counterfeit such documents.

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 pp 16.

[4] Jones, A. Valli, C (2008), Building a digital forensic laboratory, Burlington, Elsevier pp 285.

[5] Solomon, M.G. Barrett, D. Broom N. (2005), Computer Forensics Jumpstart, Sybex, London pp 51

[6] Casey, E (2000), Digital evidence and computer crime, London, Academic Press pp10.

[7] Cohan, F (2010), Towards a science of digital forensic investigation, IFIP Advances Digital Forensics VI, China pp 17-35

[8] Altheide, C. Carvey, H (2011), Digital Forensics with Open Source tools, Elsevier, MA USA pp 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), www.adobe.com (Accessed 15 January 2013).

[12]Gartner Research (2013), 'Which operating system will be 2011's bestseller', http://www.gartner.com/technology/research (Accessed 15 January 2013).

[13] Top Tech News (2010), 'Windows 7, Office Drive Record Microsoft Revenue'. http://www.toptechnews.com/story.xhtml?story_id=11300CM9DYVG (Accessed 15 January 2013).

[14] Winhex (2012), www.x-ways.net/forensics (Accessed 13 June 2012).

[15] Porn detection stick (2012), www.paraben-sticks.com/porn-detection-stick (Accessed 9 August 2012).

[16] Mabuto, E.K. Venter, H.S (2012), "User –generated evidence from graphic design applications", International conference on cyber security, cyber warfare and digital forensics, CyberSec2012, pp. 195-200.

[17] Mabuto, E.K. Venter, H.S (2012), "Finding digital evidence in graphic design applications", 7th International Workshop on digital forensics and Incident Analysis, WDFIA 2012, pp12-26

[18] Lien, C.C (2010), "Fast forgery detection with the intrinsic resampling properties", Journal of information security, Vol 1 no1, 2010, pp. 11-22.

[19] Stamm, M.C (2009). "Forensic detection of image tampering using intrinsic statistical fingerprints in histograms". APSIPA Annual summit and conference, Japan 2009, pp. 563-572.

[20] Cohen, K (2007). "Digital Still Camera Forensics", Small scale digital device forensics Journal, vol1, no 1, June 2007. pp. 2-8.

[21] Farid, H (2009). "Image forgery detection", IEEE Signal Processing Magn, 2009, pp. 16-25.

[22] Bayram, S. Avcibas, I. Sankur, B and Memon, N (2006), "Image manipulation detection," J. Electron. Imaging, vol. 15, no. 4, pp. 41-52, 2006.

[23] Wang, J (2010), "Image forensics based on manual blurred edge detection", Multimedia information networking and security (MINES), 2010, pp. 907-911.

[24] Memon, N (2012), "Photo Forensics". International workshop on information security, NYU, 2012.pp. 1-27.

THE END