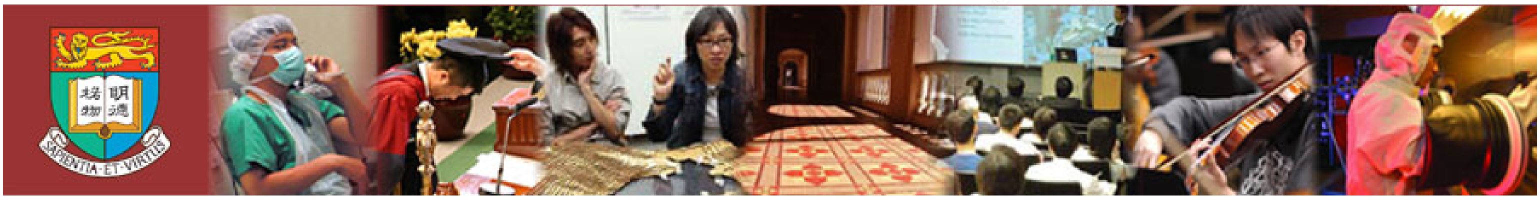


A Forensic Analysis Approach to Smartphones from a criminal investigation perspective

by

KONG, Yu Cho
江以藻

A thesis submitted in partial fulfillment
of requirements for the degree of
Master of Philosophy
in
Computer Science
at
The University of Hong Kong

October 2015

Abstract of thesis entitled

"A Forensic Analysis Approach to Smartphones from a criminal investigation perspective"

submitted by

KONG, Yu Cho

for the degree of Master of Philosophy

at the University of Hong Kong

in October 2015

Ever since the introduction of new functionalities like social networking and instant messaging, there has been a remarkably growth in the number of smartphone users. This innovative communication method also increases the likelihood for deploying smartphones, in view of its diversity and anonymous nature, as portable devices used in criminal activities. Thus, the objective of this research is to identify and review proper technical approaches in conducting forensic examinations on smartphones. The term, mobile device forensics denotes the recovery of digital evidence or data stored on a mobile device by any method or scheme that is forensically sound. This is a two-stage process which comprised of data extraction and analysis. Most of the forensic toolkits being used to gain access to a phone's internal memory are developed by forensic companies who design their own programs and acquisition methods. So far these toolkits have not been independently verified or tested for full memory acquisition. Accordingly, in the first part of this thesis, research experiments will be carried out to evaluate if the smartphone backup option, physical extraction using custom boot loader or the equipment specifically build to facilitate the invasive task of JTAG (Joint Task Action Group) acquisition can be used to acquire data and at the same time preserve the integrity of such digital evidence. The latter half of the thesis will examine the acquired data by means of various decoding software to determine their relevancy to forensic investigations. Test results are also cross-evaluated by commercial forensic tools so as to make a comparison on their effectiveness and completeness in analyzing the extracted data. The ultimate goal is to ensure digital data so recovered by mobile forensic tools can be adduced as reliable evidence in court proceedings. Some drawbacks of the mobile forensic toolkits and procedures will also be highlighted. For instance, it is considered that there is no single tool or method which is capable of acquiring all necessary evidence from various smartphone models. Lastly, this thesis will conclude with a synopsis of findings and the future work planned in this area.

(343 words)

DECLARATION

I declare that this thesis represents my own work, except where due acknowledgement is made, and that it has not been previously included in a thesis, dissertation or report submitted to this University or to any other institution for a degree, diploma or other qualifications.

Signed ……………………………
KONG Yu Cho

Acknowledgement

It is not an easy task for me, being a part-time student, to continue studying a higher degree with the university and carrying out research works in the realm of computer forensics. This program has enriched my vision on academic research studies and facilitated my self-reflection of the practical work that I discharge everyday as a computer forensic examiner.

I am grateful to Dr. K P Chow, my supervisor who patiently guides my study and allows me to take part in his research team. I would like to thank my team members Michael, Hayson, Ricci, Raymond, Vivien, Pierre, Kenneth, Xiaoxi and Frank for the support, suggestions, comments and assistance that they have offered me in my thesis and research projects.

I would like to thank Anna from Belkasoft Support, Paul from Sanderson Forensics Limited and the Magnet Forensics Team for letting me test run on their products, namely Belkasoft Evidence Centre, Forensic Browser for SQLite and Internet Evidence Finder respectively.

I would also like to thank my fellow worker, Zeeman, who assisted me in my office administration work while I was heavily engaged in the research project presentation and thesis writing. He also inspired me to think differently when conducting the research experiment. Besides, I am in gratitude to my wife Cherry who had patiently managed and arranged for the house work in the past two months. Also, my sister Irene had helped me in proof-reading my work papers to make sure they were comprehensive and readable on my research topics.

There is a long list of friends and fellows who have enlightened me to pursuit in lifelong learning and reminded me of the significance in self-upgrading to fulfill my goal in life. I wish to take this opportunity to thank them wholeheartedly. Without their encouragement and support, I will not be able to work on a degree of Master in Philosophy with the University of Hong Kong.

Table of Contents

List of Figures

List of Tables

Abbreviations

4G	Fourth Generation
ADB	Android Debug Bridge
AES	Advanced Encryption Standard
ARM	Advanced RISC Machine
ASCII	American Standard Code for Information Interchange
BIOS	Basic Input/Output System
BROM_DLL	Boot ROM Kernel Library
BYOD	Bring-Your-Own-Device
DA	Download DA
DCC	Digital Command Control
DLL	Dynamic Link Library
DMD	Droid Memory Dumper
e-Banking	Internet Banking
EMF	Data Partition / Media Encryption
eMMC	Embedded Multi Media Card
ESE	Extensible Storage Engine
GB	Gigabyte
GPS	Global Positioning System
HFSX	Hierarchical File System Plus
HFS+	Hierarchical File System Plus
ID	Identification
IDC	International Data Corporation
IEF	Internet Evidence Finder
IMEI	International Mobile Equipment Identity
iOS	iPhone Operating System
JTAG	Joint Test Action Group
KB	Kilobyte
LiME	Linux Memory Extractor
MAC	Macintosh
MIPS	Microprocessor without Interlocked Pipeline Stages
MTK	MediaTek Chipsets
NIST	National Institute of Standards and Technology
NT	Windows NT (New Technology)

NVRAM	Non-Volatile Random Access Memory
OEM	Original Equipment Manufacturer
OS	Operating System
PC	Personal Computer
PCB	Printed Circuit Board
Plist	Property List
RAPI	Remote Application Programming Interface
RAM	Random Access Memory
ROM	Read-only memory
SHA	Secure Hash Algorithm
SIM	Subscriber identity module
SMS	Short Message Service
SoC	System-on-a-Chip
SSH	Secure Shell
TAPs	Test Access Ports
TCP	Transmission Control Protocol
UEFI	Unified Extensible Firmware Interface
UFED	Universal Forensic Extraction Device
UID	Unique Identification
US	United States
USB	Universal Serial Bus
UTC	Coordinated Universal Time
UTF-8	8-bit Unicode Transformation Format
WMS	Windows Mobile Operating System
X86	Intel x86 Architecture
XML	Extensible Markup Language

Chapter 1 Introduction

1.1. Background on mobile forensics

Nowadays mobile data has formed part of crucial evidence in criminal investigations. A drug dealer can use smartphone to keep a customer list and make contact with them for trafficking drugs or a corrupt government official may record revenue of bribe moneys in his mobile device for easy access. As such, the capability of law enforcers in retrieving digital evidence from these devices has been thrust into the spotlight. In the past, standard extraction tools were able to capture data from traditional mobile phones since they were solely used for verbal or text communication with limited amount of information stored on the SIM card. But in recent years as technology advances, in particular there is a wide array of applications in connectivity function and given the device's ability to communicate via Internet, interaction between people can take place across vast geographical areas round the clock.

The latest mobile phones are often called "Smartphones". They are not only telephones but also incorporate application-based options such as cameras, calendars, alarm clocks, compass or can even act as a health coach. Different from last decade when storage capability of a mobile device was almost nonexistent, the majority of smartphones can now provide over 16GB internal capacity to users for storing data and serving as a personal convenient device to incorporate such functionalities like social network, portable office or entertainment centre, etc. Considering its evolving nature, forensic examiners require more sophisticated techniques to acquire digital data from smartphones.

The penetration rate of smartphones in the market has been rapid and inevitable. The Comscore 2014 Statistics revealed that in the US, smartphones had overtaken desktop computers in terms of total digital media engagement [1]. In 1983, Steve Jobs publicly announced that “Apple's strategy is really simple. What we want to do is we want to put an incredibly great computer in a book that you can carry around with you and learn how to use in 20 minutes. That's what we want to do and we want to do it this decade. And we really want to do it with a radio link in it so you don't have to hook up to anything and you're in communication with all of these larger databases and other computers.” [2] This truly reflects what is happening on the recent developments of mobile device. Indeed many computer software and applications have now been plotted or migrated into mobile platform because the industry believes the personal computer (PC) market will be dominated by mobile device in the near future. In the US, mobile applications are the fuel for driving the growth of over 80% of consumption activities whenever Internet browsers are used on these devices [1]. Such activities include checking the weather, conducting e-Banking, instant messaging with a friend, posting a status on social networking site, listening to video and audio streaming, making seat reservation, etc.

Needless to say mobile devices can provide valuable sources for crime investigations in view of the volume of user contents they generated through social networks and messaging applications every day. Besides, the diversity, anonymous nature and portability of the device could have increased its usage in criminal activities. While analyzing the call records, contact lists, text messages, images and geo-location found on a smartphone can help establishing critical connections between victims and suspects, the checking process may give the crime investigator possible clue to further the investigation. Mobile applications of a smartphone are often considered an important source for securing material evidence to prove an illegal act: a mobile phone is always tied up to an individual, unlike a computer which may be used among employees or family members. For this reason, in analyzing the data stored in a mobile phone, it will give an investigator access to plenty of personal information belonged to a particular person which may not be easily available elsewhere. The value of such data can be so important for prosecutors to adduce as evidence in a court proceeding and without this may become a bar to find out the facts of a crime. For example, the instant messaging between two suspects and GPS locations of their whereabouts can provide circumstantial evidence to convict a defendant whereas the social identity of a suspect on an online platform may alert an investigator to explore into his personal profile for understanding his modus operandi and if possible, build up connections with other syndicate members.

Another major advantage is the data retrieved may contain deleted information even after the phone user intends to destroy or make it unrecoverable. Flash memory of a mobile device, by its physical nature, is more resistant to high temperature and pressure, thus making the data stored therein more difficult to destroy. Moreover, mobile devices generally use wear levelling algorithms to write new data to less used memory area and deleted data will only be wiped off until a block is full. Although there is possibility to recover these deleted data within specific time frame, a full copy of the phone memory has to be acquired to facilitate such recovery.

1.2 Principles of mobile phone forensics

Mobile phone forensics is viewed as a division of digital forensics which includes the recovery and analysis of digital data that is of investigative interest. This can be done by any method or scheme provided that they are forensically sound so that the integrity of evidence is maintained and the findings can be admissible in a court proceeding. The term “Forensically Sound” implies the methods and techniques used in the preservation, acquisition, analysis and documentation of digital evidence adhered to the principle of non-contamination of the original data. Failing this the modification made on the device configuration during acquisition may risk in invalidating digital evidence in court proceedings even if all handling procedures are well documented.

The operating system (OS) of a smartphone is no different from a computer system. But conventional computer forensic techniques may be inadequate, incapable or completely infeasible to obtain “smoking-gun” information from a mobile device due to its embedded system architecture and the volatile nature of phone memory. Communication agents or boot loaders which are loaded in the phone memory will be used to establish communication between the target phone and computer workstation set up for the acquisition.

Mobile forensics has been further sub-categorized by referring to the underlying OSs of different phones. Subject to system constraints, each category adopts different scientific approach in data acquisition. Thus, they are named as

Android forensics, iOS forensics, Windows Mobile forensics, BlackBerry forensics and so on. In this document, several selection principles are set down to help identifying an appropriate extraction methodology for use. In short, they include ease of use, completeness of information extracted and the accuracy in presenting test results.

1.3 A brief introduction to various smartphone systems

According to the International Data Corporation (IDC)'s report [3], in the second quarter of 2015, the global shipments of smartphones were up to 341.5 million units with Android and iOS devices took up 96.7% of the market share. The same report recorded the shipment of Windows Phone and BlackBerry OS only accounted for 2.6% and 0.3%, respectively.

1.3.1 Android OS

Since the release of Android OS in 2008, the number of smartphones running on its system has increased tremendously. For instance, in 2009, Canalys estimated Android smartphones only gained approximately 2.8% of the market share [4]. But in 2015, the IDC's report noted the sales as high up to 82.8% [3]. Its popularity is largely attributed to the open standard adopted in the system kernel design which makes the Android phones compatible with a wide range of hardware platforms which use the same Linux kernel and the ARM architecture can officially provide support to x86 and MIPS. The open source licensing policy for program code of Android OS also plays a role. Hence, hundreds of hardware and software vendors are willing to invest on this highly fragmented platform and nearly every month, a dozen of new Android smartphones are launched worldwide.

In 2014, the August report of OpenSignal remarked a record high number of 18,769 distinct devices to have downloaded Android applications in a few months' time [5]. Starting from the use of OS named KitKat, Android has fully utilized the security model of Security Enhanced Linux which enforces the sandboxed area to allow its user running applications with minimum permission and adopts a general "deny" principle to other applications unless authorized [6].

1.3.2 Apple iOS

In 2007, Apple launched the first generation of iPhone which combines different functions of a mobile phone, digital camera, multimedia player, personal digital assistant and Internet connection. According to the IDC's report in 2015, iPhone gained a share of 13.9% in the global shipment [3]. iPhone uses HFSX as its file system, a modified version of HFS+. HFS+ is developed by Apple to replace MAC OS 8.1. It enables a user to handle larger files with Unicode being applied on the names of file system objects: the name of an individual file or folder can come up to 255 characters [7].

The main storage of iPhone is divided into two partitions. The first partition contains the system fundamental structure and corresponding applications whereas the latter stores data manipulated by end users.

iOS has enhanced its protection to user data by deploying a hierarchical encrypted file system with dedicated AES 256-bit cryptographic engine embedded between the system and data storage.

1.3.3 Windows Phone 8.1

Since the last decade, Windows OS has been a major system in the realm of computer industry. The current Windows OS 8 deploys a consistent look and feel amongst different models of desktops, laptops, tablets and smartphones. Windows Phone 8.1, which shares the same NT kernel as Windows 8 [8], is relatively new to the mobile device arena. It occupied a global shipment share of 2.6% in the 2nd quarter of 2015 [3] with a better adoption rate in European markets such as Spain, France, Germany, Italy and the United Kingdom [9].

The platform integrity is protected by the trusted boot and code signing against malware attacks by permitting only trusted or validated firmware images to load the OS. Windows Phone architecture deploys the design of System-on-a-Chip (SoC) to provide the Unified Extensible Firmware Interface (UEFI) boot loaders and environment. The latter implements the secure boot standard of which all codes in the OS should be signed by Microsoft, including the OEM drivers and applications.

The Windows Phone OS has deployed Bitlocker technology to encrypt all user data files stored locally in internal partition. This prevents phone data from offline hardware attacks. However, a user cannot enable device encryption unless an Outlook account is installed into the phone with the Microsoft Exchange server configured to cater for device encryption [10]. Alternatively the target phone can be connected to the enterprise device management server which propagates a device encryption policy. The phone will automatically begin running the encryption of the system and user’s partitions. But unlike a computer, mobile phone users cannot use the OS to enable or disable data encryption since the device does not have recovery key backup. As soon as the device encryption has been turned on, even Microsoft Exchange servers and enterprise device management servers cannot disable it later on, unless a factory reset is conducted.

1.3.4 BlackBerry

The first generation of BlackBerry phone was launched in 1999 in Munich, Germany. In 2003 the more popular assembled BlackBerry smartphone was released with extended functionalities to support push email, text messaging and Internet browsing.

BlackBerry smartphones occupied 0.3% of the global shipment share (Q2 2015) [3] and is focused on pushing email services since it is always powered on to participate in synchronizing with BlackBerry email server. Different from other mobile devices, BlackBerry smartphone does not require desktop synchronization. Its OS is equipped with a password keeper program to store sensitive information.

Since Blackberry smartphone occupies a small share in the global mobile phone market, it will be excluded from subsequent research and experiment of this thesis.

Table 1-1 below shows a comparison summary of different smartphone system.

	Android	iOS	Windows Phone	Blackberry
2015 Global Shipment per OS	82.8%	13.9%	2.6%	0.3%

System Design	Open Standard	HFSX by Apple	Microsoft Mobile	Proprietary OS
Kernel	Light Weight Linux	Unix-like	Windows-based	Java-based
System Security	Security Enhanced Linux with sandboxed area	Hierarchical Encrypted File System with AES 256-bit cryptographic engine	UEFI Secure Boot Loaders and Bitlocker technology	Integrated with hardware and software encryption mechanisms
Applications Market	Open	Close	Close	Close

Table 1-1 Brief Summary of various Smartphone Systems

1.4 The challenges of acquisition and examination

The standard practice of forensic acquisition is to preserve all relevant electronic data or information, by utilizing the best available technological means, from a mobile phone without any alteration to the data itself. The scope of mobile device forensics is more complex than examining a PC as the number of major PC OS vendors is small and their data storage device is non-volatile in nature. Indeed there are far too many manufacturers willing to deploy their own proprietary technology and format in designing a mobile device. It is understandable that an authentic mobile phone manufacturer often has to consider customer experience and act accordingly to change the form factors, system design, storage size, functionalities, peripherals, security settings, bundled software and applications to maintain competitiveness. Considering those factors like the bursting rate on releases of new mobile devices, the rapidly evolving mobile device technology, the more advanced system functionalities and application encryption methods as well as huge volume of data collected and generated daily by the 4G mobile network, forensic examiners are facing with major challenge as to how these data can be analyzed in a thorough and effective manner.

A mobile phone contains an inbuilt communication system and in most cases, equipped with proprietary storage mechanisms. Traditionally when carrying out forensics examination on a "dead" computer, the examiner will simply remove the internal hard disk, connect it to the cloning device via a write blocker and create a forensics image of the whole disk for data analysis. During the course of action, the one-way hash function will be used to prove the image acquired is a bit-stream copy of the original hard disk. Mobile phone forensics, on the other hand, is performed on a device which is able to manipulate data by itself, without any human intervention, when it is powered up. The acquisition process is operated on the embedded OS for which data may be lost or modified as the background process is constantly running. In consequence the forensic hash on the device storage will be given a different value whenever it is generated. Therefore, it is unlikely that a bit-stream copy of mobile phone memory can be obtained if the extraction process runs on the default embedded OS.

Another obstacle for mobile forensics is that the majority of mobile devices are operating on different OS versions and hardware or possessing customized features. Within the same OS, data storage options and file structures can vary, making it more difficult to acquire forensic artifacts from the device. Thus it is crucial for crime investigators and forensic examiners to keep abreast of the latest developments and techniques on mobile forensic toolkits so that the most suitable methodology is applied to analyze phone data. Indeed some prominent vendors have created a number of toolkits which are capable of handling different phone models, but their costs are too high to maintain an up-to-date support phone list.

These commercial toolkits have been developed to extract data from proprietary mobile devices. They can automate much of the acquisition process, making it possible for minimally trained frontline investigators to operate. However, these toolkits are largely designed by third party companies who use self-developed programs and extraction methods to gain access to the internal phone memory. Forensic examiners consider this methodology to be “black box operation” since the vendors will be unwilling to disclose their methodologies, such as the architecture and source codes of the forensic software, so as to safeguard their interests and investments. Besides, these toolkits are often limited themselves to a small number of mobile devices made by selected manufacturers. Since 2008, the National Institute of Standards and Technology (NIST) had evaluated 44 forensic toolkits operated on phone models ranging from traditional mobile phones to smartphones [11]. The results indicated that none of them could support all the selected mobile phone devices under test.

To make the situation even worse, the same methodology previously used on the original mobile device or system can become useless upon the release of a new generation or version. Furthermore, updates on firmware or ROMs are not only system specific but they are hardware dependency since some manufacturers are accustomed to add distinctive functionality and customized options to their systems. In summary a forensic method used effectively for a certain device or embedded OS may not work the same on smartphones produced by a different manufacturer.

Features concerning decoding data from mobile phones and third party applications have played no less significant role than the extraction capability in the forensic process. Irrespective of whether the data is extracted via logical or physical methods, examiners have to reconstruct file system structure of the mobile device and decode application data into readable format. Given there is in existence of over 1.6 million and 1.5 million of mobile applications for Android and iOS respectively, which are found on their online store in July 2015 [11], the extraction and decoding of data of all those applications by a single mobile forensic toolkit seems impracticable. In fact examiners should be cautious about the selection of toolkit used for the extraction and decoding process. For instance, extraction process can be carried out by one tool whereas the decoding process is performed by some other forensic tools so as to ensure the data is being parsed properly without missing any information. The analyzed information can then be collaborated to give the best possible findings.

In real situation, the forensic examination process is to compete against time. The quicker the data is extracted, analyzed and acted upon, the sooner the offender can be apprehended, prosecuted and public confidence restored.

To sum up crime investigators and forensic examiners are facing the following challenges in common:

- There is no single forensic toolkit that supports data extraction from all mobile devices. The selected toolkit used for a particular task is normally the one which the examiner feels most comfortable to work with, or which gives the best display of visual results for investigation.
- There is a lack of forensic program supporting the analysis on logs and

artifacts from massive mobile applications.

- For traditional mobile phones, forensic examiners used to obtain call records, SMS and contact lists for data analysis. However, examiners will now focus on the internet browsing history, social communication data, application databases or deleted data kept by smartphones.

- Offenders often delete files from their mobile devices with a view to concealing their illegal activities.

- Commercial and specialized forensic toolkits are expensive. Hardware and software update on the toolkits has to be constantly conducted to cope with the latest phone models.

- As the public have enhanced awareness on the concept of “privacy”, manufacturers are inclined to implement robust security controls over mobile devices. If the device is passcode protected, the passcode has to be bypassed before acquisition. Similarly, whole disk encryption mechanisms implemented by some latest models can be a barrier to access phone data or information.

1.5 Contributions of this research

This thesis investigates the types of information that can be recovered from data held by a smartphone. Based on different approaches, research experiments are conducted to acquire data and the data so extracted are then analyzed by decoding software with a view to retrieving useful information, files and artifacts. The competency and compatibility of selected methods among different OS, that is, Android, iOS and Windows mobile, will also be reviewed. Cross-evaluation of test results by commercial forensic toolkits is conducted to ensure the completeness and correctness of the extracted information.

The contribution of this thesis is two-fold: it demonstrates different information recovery techniques used on a variety of mobile devices and at the same time compares the test results among them.

The objectives set down for this project include:

- to investigate and review the literature which relates to mobile forensics on smartphones;
- to identify and review proper technical approaches in conducting forensic examinations on smartphones; and
- to evaluate the competency of selected toolkits and methodologies in recovering and presenting evidential data stored in the smartphones.

1.6 Organization of document

The other parts of this thesis are organized as follows:

Chapter 2 reviews the existing researches on smartphone forensics

Chapter 3 identifies and confirms proper approach in examining smartphones

Chapter 4 describes research experiments and evaluates the result of different methodologies used for data acquisition and analysis on iOS, Android and Windows Phone systems.

Chapter 5 concludes the thesis and outlines the future landscape.

Chapter 2 Literature review

2.1 Different extraction methods

Early researches on mobile phone forensics only focused on acquiring data from the SIM card [13]. As phone models are equipped with added features, new forensic analysis tools are necessary to acquire potential evidence from these new functionalities. Research has to be conducted to explore a methodology which is capable of acquiring complete memory dumps and translating the captured data into readable format.

Given there are various forensic tools available for acquiring memory and translating the captured data into readable format, the acquisition methods used are in general categorized into either physical or logical acquisition.

Physical acquisition tools can extract all binary data from the internal memory and save them to binary files. However, the de-soldering of memory chips from the printed circuit board (PCB) is rather invasive and may carry the risk of damaging the mobile device [14]. Instead, a less invasive approach is to acquire data via JTAG (Joint Task Action Group) ports attached to the PCB but this method is only available for limited phone models [15]. Notwithstanding the above, the two methods are able to bypass passcodes and allow an examiner to recover data from a “locked” phone [16].

Klaver used the methodology of ‘pseudo physical acquisition’ to perform data extraction by loading dedicated software to phone system [16]. The same method can also be performed by the RAPI tools developed by Hengeveld [17].

Since 1990, the methodology of JTAG has been used to test mobile devices. The JTAG port can access raw data stored in the connected device to obtain a physical image. By communicating through the Test Access Ports (TAPs), an examiner can extract the entire flash memory contents from compatible devices by making use of a boundary scan method to push data to the forensic computer [18].

Logical acquisition tools can recover logical files stored in the file system rather than obtaining a raw image of the memory chip. The extraction software is stored on an external card for execution in the target phone. After completion, the memory card containing the acquired files will be removed for analysis. Nevertheless, no information can be recovered from deleted blocks as a physical memory dump cannot be made since the OS of the device has “withheld” these blocks from being acquired [19].

Jahankhani described several logical acquisition tools, which could be used to examine a smartphone, but he did not perform any testing on these tools: they include BitPim, Oxygen Phone Manager, Paraban Cell Seizure and MOBILedit [20].

Regarding the test standards, the NIST and a number of law enforcement staffs had worked together to evaluate mobile forensic toolkits and issued guidance to determine their efficiency and functionality in data acquisition and reporting of forensic results. The document can serve as a baseline in assisting a forensic examiner to choose the correct toolkits to use [11].

According to the Good Practice Guide for Computer-Based Electronic Evidence, no modification of data is allowed after the target device has been seized and every examination step must be documented by certified professionals [21]. But in the case of smartphone most forensic procedures can only be conducted when the device is powered on. Data modification is inevitable as being a direct consequence of the acquisition and data analysis process. Thus, an examiner should have the expertise to the underlying technology so as to fully explain and defend those alterations that have been made to the device.

As mentioned earlier, mobile forensics conducted on a particular device is primarily dependent on the underlying OSs. Previous researches on Android, iOS and Windows Mobile forensics are discussed below.

2.2 Android Forensics

The OS of Android is based on Linux. Memory analysis tools previously targeted on Linux were therefore used by researchers to perform the tests. In 2010, three research projects, Volatilitux [22], fmem [23], and memfetch [24] were conducted on memory acquisition but they could not dump all data from the device memory since the process might potentially trigger contamination to the data in the user partition. The unsuccessful results indicated the research tools could not cope with the large family of Android kernel versions and OSs.

In 2011, Sylve introduced Droid Memory Dumper (DMD), a memory acquisition tool [25]. It involves the capture of memory data and address translation of each memory page. The result will be saved via a TCP socket. However, DMD has the following limitations:

- the USB debugging mode should be enabled to allow the DMD to tether data; and
- the tool requires root privileges to capture system data.

In 2012, Valenzuela stated that he had modified DMD to enable it to work with the latest Android developer toolkits [26]. He renamed the tool as LiME Forensics which claimed to be capable of dumping full contents of memory from

Android devices. In spite of this, there is still room for future study to explore whether direct access can be made to the phone memory without “rooting” the device to conduct memory dumping. “Rooting” is in fact changing the state of the device and if used, is likely to face legal challenge on the integrity of evidential data recovered from the target device.

Other published researches for Android platform and forensic methodologies are reviewed. One of the works was conducted by Lessard and Kessler [27]. The authors investigated Android smartphones and used a ‘dd’ command to acquire both logical and physical images. Cellebrite was later used to acquire the same images for comparison of results.

Vidas et al. also carried out research in this area [28]. He proposed using a custom recovery image which allowed the device to divert from booting its OS. The recovery image was designed to provide functionalities of dumping the flash memory, allowing the execution of super user account command to gain root access and adding some custom transfer binaries. So when the device was booted into the recovery mode, the adb tool could collect data and transfer them to a connecting computer via the USB port.

The thesis by Vijayan “Android Forensic Capability and Evaluation of Extraction Tools” has been reviewed [29]. The author evaluated three mobile forensic tools which extracted data from two HTC Android smartphones. The results were compared to assess their effectiveness. Only logical extractions were performed because of the limitation of the selected tools. As full memory dump of the tested phones had not been obtained, the relevancy of its final findings was thus reduced.

In Thing et al., the authors conducted research on an instant messaging scenario [30]. They recovered the content of messages from specific memory locations. The experiment concluded with 100% acquisition rate for outgoing messages while incoming messages gave an acquisition rate in the range between 75.6% and 100%.

Android system has two methods for logging. The first option is the internal SQLite database of individual application which stores data to enhance user experience such as recording web pages that have been visited by a user. The second one is the various debugging and system logs generated by the application or OS with alert message being delivered to the manufacturer when an incident of failure occurs. The timestamps for all logs are based on the internal clock of the device. Before extraction, the examiner will need to confirm the time on the device by checking the system clock. Temporary log buffers are stored in the directory /dev/log in RAM as a piece of virtual memory [31]. They are main buffers and by default will record application logs: the system buffer records system activities; the event buffer records system event information; and the radio buffer records cellular network information [32]. The examiner should aware that the content of buffer is volatile and can only be accessed when the device is running. Although the buffers do not have very high capacity, for instance, the size of event buffer is 256KB while the other three are 64KB each, they cannot be easily overwritten in a short period of time.

So far researches on log extraction and analysis on data retrieved from Android device as well as presenting these data in a timeline are rare. A review on research works shows that Jin has implemented a timeline framework that can analyze events within the Android OS in an orderly manner [32].

Some Android apps are found to be capable of collecting log files from the device. Grover presented an application called DroidWatch but the following shortcomings are identified: the datasets are collected without root privileges; the approach requires installation of software on the device; and no data on occurrences can be produced before the app is installed [33]. Therefore, post-incident investigation cannot be carried out if the application has not been installed on the device beforehand.

There have been other commercial forensic toolkits developed for criminal investigations in order to secure evidence for presenting in court proceedings. For example, the Mobile Phone Examiner created by AccessData [34], NowSecure Forensics by NowSecure [35], Cellebrite UFED Physical Analyzer by Cellebrite Limited [36] and XRY Office by Micro Systemation AB [37]. An examiner should note that the above tools can only collect data from selected phone models that are on the support list.

2.3 iOS Forensics

Zdziarski has implemented a physical extraction methodology for the iOS. His research revealed that data in system partition had been changed whilst the user data partition was left intact. The toolkit under test recovered a bit-by-bit copy of the memory image by using SSH to establish a tunnel between the target device and the workstation [38]. Results on data extraction from iTune backup files of an iPhone 3GS by some commercial forensic toolkits were also evaluated by Hoog [39]. Apart from forensic procedures, the test tools were compared as to their ease of installation and use, integrity of the acquired data, etc. The author concluded that depending on the functionalities, different forensic tools might lead to different quantity and quality of the acquired data.

In the research paper of Bader and Baggili, the authors investigated if it was possible to use Apple iTunes backup utility to perform logical forensic acquisition on iPhone 3GS [40]. During the acquisition process, the authors were able to identify all the files that were of interest to crime investigation such as email messages, text and multimedia messages, calendar events, browser history, call logs, contacts and locations. The iPhone 3GS was connected to a computer workstation and the iTunes backup procedure was initialized without triggering the synchronizing option. Unfortunately some modifications to the phone data were traced as the activation of the write-blocker at the computer’s USB port had failed during the backup process.

Husain et al. were dissatisfied with the performance of some commercial forensic toolkits and gave an account of the disadvantages which they found from the acquisition techniques involving jailbreaking procedure. They proposed the use of iTunes backup utility and the extracted data could be decoded by means of file parsers, SQLite database browsers and plist (property list) editors [41]. Jung et al. split the acquisition procedures into two groups [42]. The first group concerned the usage of jailbreaking technique to bypass the security mechanism of iOS while the other did not make any change. After the iTunes backup was put on, a survey was conducted to compare the extracted results derived from the social networking applications. It was found that files extracted in the normal procedure from non-jailbroken devices were stored with a hash file name while those extracted from jailbroken devices were saved with the original filename and type. Besides, temporary files from Facebook could only be retrieved through jailbroken devices. Similar experiment was conducted by Tso et al. on an iPhone 4 without using the jailbreaking technique [43]. Except for Facebook, additional files from social network and chatting applications (WhatsApp Messenger, Windows Live Messenger, Skype, Viber, etc.) were found storing in iTunes backup folders and could easily be viewed by using plist editors and SQLite browsers.

Arrifin et al. had successfully extracted deleted image files by interacting with the iOS journalizing system from the HFSX volume [44]. They highlighted the complexity of unallocated data retrieval procedures and the importance of data encryption. The technique was applied on iPhone 3GS and iPhone 4 and succeeded in extracting the encrypted copies of files.

2.4 Windows Phone analysis

The Windows Mobile OS (WMS) and Windows desktop OS are similar in the file system structure and directory layout as well as the common presence of many system files and applications. SMS messages, contact lists and phone call records are stored in the database files named as cemail.vol and pim.vol in the file system. Casey et al. had attempted to review some files that were of evidential value to an investigation and proposed the use of Windows Mobile emulator to extract the cemail.vol and examine the contents of it [45]. Moreover, he used the Microsoft Remote Registry Editor to examine the Windows Mobile Registry and its hive files.

Klaver also evaluated some physical acquisition methods conducted on Windows mobile devices [16]. He developed the xpdumpcedb.exe and wmdumped.exe to retrieve user information from the cemail.vol and pim.vol files respectively and exported the results to XML files after they had been extracted from the binary image. Furthermore, Klaver developed a Python script, cedbexplorer.py, to extract both active and deleted data from the cemail.vol file. Klaver continued his research on Windows Phone 7 and succeeded in reverse-engineering the Microsoft Embedded Database volume to recover additional records [46].

In his research, Rehault found that specific boot loaders could be used to extract data from Windows Phones while preserving data integrity. Like Klaver, he wrote a Python script, MsgCarving.py, to recover the directory structure and message content, including the deleted data [47]. Rehault also developed a custom-built tool to extract registry keys and configuration information of the device from the Registry.

In 2012, Canlar introduced an on-device forensic tool, LiveSD Forensics by installing a memory card to perform live data acquisition of the flash memory of WMS [48]. Its startup procedure called for a shell script to automate the extraction. The tool claimed to have smaller footprint and would not be interfered by background processes. In the same year, Schaefer explained the characteristics of Windows Phone 7 from a forensic perspective [49]. He briefly described the data acquisition methods, the extracted file system, the registry and other phone applications.

In 2014, Leong and a group of US police officers had conducted reverse-engineering to explore the useful artifact locations from Windows Phone 8 device for the investigation of a real crime case [50]. Having successfully acquired a physical memory dump from a Nokia Lumia phone by JTAG method, they obtained the user passcode from an incoming SMS. The live data could then be compared with the artifacts parsed from the store.vol and similar storage mechanism.

2.5 Summary

There were only a few mobile phone forensic researches conducted for the sake of criminal investigations. This thesis is expected to provide some useful information and directions to develop a framework on mobile phone forensics.

Chapter 3 Approach for examination

The mobile forensic process is divided into four main segments: acquisition, decoding, analyzing and presentation of information from mobile device. Due to the inherent security features of a mobile device, isolation of the memory segment before recovery to extract data may not be feasible as it is usually integrated into the device. Furthermore, as manufacturers normally make use of diverse file systems and formats, the decoding and presentation process can be complex. It is unclear to what extent the forensic tools so designed can present the information which is held on a mobile device, or whether the results produced by different tools are consistent.

Albeit there are in existence of commercial and open source toolkits specifically designed for smartphone forensics, the extraction method has yet to be standardized. For example, some toolkits use the boot loader option to conduct physical extraction while others will use backup mode for logical acquisition. Moreover, most of the forensic toolkits in conducting acquisition will require the USB debugging mode in the phone’s system menu be enabled to carry out the process. To resolve this problem, it is necessary to identify a suitable extraction toolkit that can capture maximum data from a smartphone in forensically sound manner and deal with such situation in case the phone is password locked and the USB debugging mode cannot be enabled.

3.1 Discussion on using different acquisition methods

The NIST introduced a mobile device forensic pyramid showing five levels of extraction methods after evaluating their degree of difficulty to operate [11]. The simplest method, manual acquisition, is operated by a keypad manually and the target mobile device is displayed with data retrieved from its internal memory. A camera is mounted above the phone to record the display of the results. The next level is logical extraction process where the device's OS is in control of what data can be accessed and provided information on timestamps and file locations. The phone system may record the acquisition activities and update the system data. Generally speaking, deleted data may not be extracted as it has been removed from the file system. But in exceptional cases, individual records that are kept in the database files and marked as deleted can still be recovered by examining these database files.

The third level is physical extraction which involves making a bit-by-bit copy of the entire flash memory in binary format, from the storage medium of a smartphone without relying on the OS to access data. Custom boot loader will be used to bypass the security measure and start up the phone. After physical extraction, the binary image file including deleted or privileged data needs to be parsed and decoded to obtain file system information and readable data. This type of acquisition is likely to obtain a more complete capture of memory dump which include accessing those data or file systems which are marked as deleted or obsoleted but in fact they have not been erased unless required.

The fourth level is to make use of JTAG interface to read internal memory and create an image. The phone has to be disassembled for connecting the ports and thereafter issuing commands to the processor units to acquire physical memory dump from the chips. JTAG acquisition is a type of physical acquisition method and only binary data of the memory chips are recovered. The running process takes considerable time and the examiner is required to possess technical knowledge on JTAG for the specific model of phone. Apparently the reconstruction work from the resulting binary data will be complicated and challenging. If the mobile phone has deployed whole disk data encryption, JTAG acquisition will produce an encrypted image accordingly. In order to decrypt the data, an examiner will need to access the respective applications on the phone and thus passcodes are required. Furthermore, the JTAG method can read and write to the memory area i.e. flash new partitions to repair a damaged boot partition etc. By flashing new data to the mobile phone can be destructive to those existing data. Besides, the examiner will require knowledge of the integrated circuit, which is generally only known to phone manufacturers. JTAG hardware manufacturers only identify the JTAG pinout assignment of a list of supported phone models. For unlisted models, the examiner needs to determine the unknown pinout by removing the chips from the board and trace the connections or if test device is not available, he needs to google for personal experience via the global JTAG community. Theoretically the JTAG approach provides a forensically sound method to gain direct access to the memory clips without the need to interact with the OS of the phone, even it is password-locked.

Chip-off is a more invasive method which involves physical de-soldering of memory chips from the circuit board of the mobile device and the content being read with dedicated reader to carry out data acquisition. As far as digital forensics is concerned, it should be used as the last resort for reason that the original device is normally unable to revert back to its normal functioning state. Direct extraction is always associated with a failure rate and may be impractical in many situations when the owner requests for the return of the device. This method carries high risk in that the electronic parts may be damaged and the data so extracted may be encrypted or unreadable. To give an example, it will not be feasible on the latest encryption methodologies used in iPhone which have combined the key with the hardware ID of the phone to prevent from decryption of data stored when offline.

Moreover, not every mobile device is suitable for this process. The best scenarios to use JTAG forensics or even Chip-off acquisition include:

- Damaged devices
- Password locked devices with no bypass support
- Devices for which debugging mode is not enabled
- Examinations where non-invasive physical acquisitions are not supported and/or logical extraction of data is not sufficient.

The last method is called Micro-read when high-power microscope is used to provide a physical view of memory cells. This is however a time consuming method which renders a full extraction almost impractical.

As far as the legal requirement for admissibility of digital evidence relies on the integrity of the original data, physical acquisition utilizing a custom boot loader option, to dump the binary data from the mobile device, is preferred in mobile phone forensic investigations. However, in shortlisting an appropriate extraction method, examiners will need to consider those factors such as the OS, the making and model of the mobile device as well as the type of data or information to be captured.

3.2 Android Forensics

The Android Debug Bridge (ADB) provides a communication interface between an Android system and a computer. But for security reason, apart from making personal data backup, users do not have permission to access system reserved areas. This feature is designed to protect the stability and reliability of Android OS from malicious or poorly developed applications. Nonetheless, it may hinder the extraction of data as it is necessary to obtain full image dump inclusive of all system partitions and reserved areas.

A less-used method is by forcing an Android backup, the built in “adb backup” command to copy databases and user files from the smartphone. Android backup is an implementation by Android which utilizes the support of an application to facilitate data backup [51]. When an application has been developed, there is a flag shown on the screen to indicate its inclusion in the Android backup. The new version of Android OS has excluded Google applications from Android backup. There is also a decrease in the volume of data recovered by this solution as the Android backup is blocked by more applications.

Logical extraction is on top of the Android Backup method. Examiners should take notice that the extraction method is making use of the adb daemon running on the device but the recursive copy can only do it with shell permissions. Some of the more forensically relevant files will not be accessible unless the device is rooted so as to gain full and unhindered access to the system. In the absence of root privilege, an adb pull can still access such files like unencrypted applications, user's browser history and system information as well as other files contained in the readable directories. However, this method only allows logical data to be recovered and normally will not produce any deleted file.

The Android OS provides a comprehensive logging system to include information about installation, launching and removal of each particular application and most user actions. As the number of new applications increases, this will add to the amount of logs and data stored on an Android device and become an important source to obtain evidence for forensic investigations. Forensic examiners should keep track of all data sources in the Android systems and be able to analyze them effectively within reasonable timeline. Examiners should also ensure the evidence they collected will not be tampered with by the applications. To tackle the problem, cross-referencing with timeline entries and analyzing logs that are more difficult to modify is helpful in assuring the trustworthiness of the data. So unless the suspect actively manipulated the logs, this data might uncover the identity of person to whom he had communicated with and the time when that communication took place. The log created for internet browsing is also useful to build a profile of user’s behavior. The event log that records application execution is able to verify the integrity of other logs. For example, it can record a user had launched the camera and took two pictures. Where the timeline is near to perfection, this type of integrity checking will be much easier and can increase the degree of accuracy.

Normally user settings and interactions in SQLite databases are stored in Android applications for future use. Information that can be extracted from those databases is the same as information normally viewed by a user when interacting with the application. Except on some occasions, extra metadata will be included. To give an example, in the SQLite databases, the web browser history can show a timestamp relating to the access to websites [27].

As such, the data for which a forensic examiner is of great interest are probably stored in SQLite database, because this data is easy to interpret since most of them are presented to the user and that the data mostly created is non-volatile in nature and can persist for a long time until they are deleted by the user. For the sake of mobile forensics, the following data and information should be acquired from the application databases:

- Communication patterns - for instance, to whom the user had communicated with and when did it happen;
- Networking history - what the user had been doing on the Internet; and
- Internal events - actions that the user had performed on the phone such as updating his calendar or writing his notepad.

3.3 iOS Forensics

iOS has extended protection to user data even in cases when security components have been compromised. Since iPhone 3GS, phone contents are encrypted by default and part of the key is baked into the hardware. This is achieved by the implementation of a hierarchical encrypted file system, HFSX which adds the case-sensitive function to its ancestor HFS+. Apple mobile devices have only one disk and are divided into two partitions, the firmware partition is about 900 – 2,700 MB and the rest is utilized as user data partition [52]. The firmware partition hosts the OS but does not contain any user data that is of interest to forensic examiners. The data partition has different storage size depending on the device model. This partition allows a user to read and write application data. Since the launching of iOS 4.1, this partition is encrypted with an AES-256 algorithm, using a particular key named EMF key which is computed to apply a key generated on boot time (called 0x89B key) [53]. Depending upon a unique ID (UID) key associated with an individual device, data can be cryptographically tied to that device as an additional level of protection. Although this cryptographic key does not directly encrypt user data blocks, it can derive the encryption key to protect user data. The purpose of this process is two-fold:

- Acceleration of encryption and decryption operations so as not to affect power efficiency.
- Protection of user data so that it always remains encrypted on the device's flash memory.

The procedure for generation of the EMF key is:

0x89B = AES (UID, 183e99676bb03c546fa468f51c0cbd49)

EMF key = AES(EMF_random, key0x89B), where EMF_random is a random value generated from the device.

The EMF key is saved into a memory zone from where it can be quickly deleted to fasten the wipe device procedures. The strength of the encryption for

EMF-key generation lies with the fact that the UID key is not extractable from the device. It is fused into the microprocessor of the phone during manufacture so as to avoid a direct reading of its value.

The Touch ID sensor is built into the home button which is surrounded by a steel detection ring to wake up the sensor. After the scanned fingerprint data are converted to a string of numbers through encryption method and one way hashing, the string is then stored on the Secure Enclave of the Apple processor. So to unlock the phone, a user will scan his finger on the sensor to encrypt the data for matching with the saved string stored on the Secure Enclave [54]. Apple will not allow third party developer to gain access to the Touch ID component in order to eliminate the possibility of tampering.

The iOS uses SQLite database format in order to store such information on the device like Address Book, SMS, Call History, Calendar, Notes, Photos, Videos and Music. These databases can be cross referencing from one database to another and is able to display on the user’s interface connectively. Most forensic software consists of file viewers to interpret those data acquired from SQlite, plist and XLM database files. There are other tools such as hex viewers, converters for time/date, image and video viewers, can be used to conduct an extensive analysis of the data.

Apple does not permit installing apps that are not approved by Apple Application Process. This is why iDevice is defined as jailed devices because they are self-contained. Apple justifies the “close the door” policy by asserting that it can protect security and reliability of the system.

Prior to iOS 4, passcodes are stored in a file which is accessible via SSH. After the release of iPhone 4S, no forensic tool can acquire a non-invasive physical acquisition unless it is a jailbroken 32-bit device and the password is made known. By October 2015, more than 90% of iOS devices have been running iOS 8 or above [55], so the chance of accessing an older version device is slim.

Logical extraction can only recover a limited amount of application data or deleted text messages located in the database files. Furthermore, as Apple uses whole disk encryption, it makes JTAG and chip-off acquisition not applicable. In the past it may be a viable strategy for law enforcement agencies to send the target devices directly to Apple for acquisition, but it can no longer be used now. Subsequent to the release of iOS 8, Apple announces in its Privacy Policy that unless a correct passcode is available, even Apple support team cannot retrieve information stored in the internal memory of the phone because iOS 8 encrypts the data with a key linked to the passcode which Apple do not have access [54].

If an iPhone is connected to a computer, the iTunes backup is another source to uncover valuable data even when they had been overwritten for a certain period of time. The default behavior of iTunes, if not altered, is to make an unencrypted backup, without seeking permission, whenever it is connected to an iPhone which has already paired with the computer. If the computer which has synchronized with the iPhone is also seized, the lockdown certificate saved in iTunes will help producing an offline backup via iTunes to the "locked" mobile phones with the following restrictions:

- Items stored in the keychain will be available only if the backup is password-protected. More information can be assessed when compared to the analysis by non-protected backup.
- Cached data such as downloaded email are not available.
- Commencing from iOS 7 version, the locked iPhone needs to entrust a forensic workstation before it can push data via iTunes to the designated backup directory.

Data is exchanged in both ways disregard whether individual item has a newer version on the phone or on the computer. It is therefore recommended to perform the forensic extraction with a sterile version of iTunes to avoid updating of the device under analysis. Besides, USB write blocker will prevent the backup utility to mount the iPhone file system.

The backup folder in Windows 7 is here:

\users\name\AppData\Roaming\Apple\Computer\MobileSync\Backup\

Or the following folder in the event of Mac OS X:

/Library/Application Support/MobileSync/Backup/

Backup files are stored in different folders with a string of 40 hexadecimal digits which appears to be a hashed value and labels as unique identifier for each set of data copied from the iPhone memory.

There are five categories of files found in the backup directory:

- SQLite 3 database files
- Plain text plist files
- Binary plist files
- Multimedia and text files
- Non-standard data files

If the backup is located on the host computer and found encrypted, a brute force or dictionary attack can be attempted to recover the passcode.

3.4 Windows Phone Forensics

Windows Phone is designed to give productivity and at the same time provides security to its users and business organizations. Windows phones are equipped with on-board eMMC storage and are difficult to capture an image by normal forensic procedures because the phone will not allow running any unsigned applications or third party boot loader. This security measure has been designed to prevent unauthorized access to the content in case a device is lost or stolen. Besides, Windows Phone are designed with optional whole-disk encryption (with Bitlocker) and Secure Boot, an option to prevent booting into a non-recognized OS such as a Linux-based bootable drive, which is frequently used for digital forensics. Notably, Secure Boot does not prevent booting from an external media with a bootable recovery image of Windows 8.1 bearing the required signatures as a must [56]. It is important to note that the Secure Boot has been permanently activated on all the Windows Phones 8.1.

Unified Extensible Firmware Interface (UEFI) is a firmware interface for devices to replace BIOS with additional security features to ensure the boot loader is secured. Only the hardware manufacturer of the device can have access to the digital certificate to create a valid firmware signature. When a Windows Phone device is turned on, its firmware will start the boot loader only if it is digitally signed by a trusted authority registered in the UEFI database.

Thereafter the Trusted Boot comes in place and monitors the startup process by validating all Windows boot components. If a file has been altered, Trust Boot will prevent it from running. After the startup process, system files and applications will be loaded automatically. Again they must be properly signed in the Windows Store or they should bear the organization’s enterprise development signature. AppContainer is a sandboxing mechanism so that each application runs inside its own sandbox with the least privilege given for execution.

The Windows Phone OS supports Bitlocker technology and makes use of it to encrypt all user data stored in the data partitions. As soon as device encryption is enabled on a phone, the phone will automatically begin encrypting the main OS and internal user data store partitions. Applications that run outside the main OS cannot write any more data to the encrypted partitions. Only applications in the main OS can write data to these partitions. Once the device encryption has been enabled, a user cannot disable device encryption on the phone.

Although Windows Phone 8.1 deploys UEFI and Secure Boot to prevent any custom boot image from either loading up the system or performing any forensic extraction, Cellebrite has created boot images with required signatures for specific phone models in the recovery mode. The custom boot will be executed to allow for a bitstream copy of the phone memory. Thereafter, the phone will restart its normal state as if the data has not been altered on the phone.

Due to the tight security approach imposed by Windows Phone 8.1, the experiment will be carried out using JTAG acquisition method to extract data from the device under test without contaminating the user data or destroying the phone during physical disassembly as in the case of other intrusive chip-off methods.

3.5 Summary

Table 3-1 below shows a summary of the usage and limitation of different extraction methods for different phone operating systems. A preferred extraction method for each phone operating system is also enlisted at the bottom.

	Android	iOS	Windows Phone 8
Manual	Usage Simplest, operated by user and data on screen captured by camera. Limitation Examine unlocked device and user viewable data only.	Usage Simplest, operated by user and data on screen captured by camera. Limitation Examine unlocked device and user viewable data only.	Usage Simplest, operated by user and data on screen captured by camera. Limitation Examine unlocked device and user viewable data only.

Logical	Usage Use ADB Backup to provide logical views of files and application data on timestamps and file locations; need root privilege to view all data. Limitation OS in control of data accessed and depends on the backup setting of individual applications; not including deleted data; need password to access.	Usage Use iTunes backup or examine backup files stored on synchronized computer to provide logical views of files and application data. Limitation OS in control of data accessed and depends on the backup setting of individual applications; not including deleted data; need password to access.	Usage Extract multimedia files via USB using media transfer protocol. Limitation OS in control of data accessed; need Microsoft account name and password to extract contact, calendar.
Physical	Usage Use custom boot loader to capture complete memory dump; bypass security measure and settings. Limitation Custom Boot Loader may not be available.	Limitation No forensic tool can acquire a non-invasive physical acquisition unless it is jailbroken because of the secure boot policy prohibited any custom boot loader.	Usage Only commercial tool Cellebrite can capture physical memory from limited models with signed bootable image. Limitation No other forensic tool can acquire a non-invasive physical acquisition.
JTAG / Chip-Off	Usage Direct access to the internal memory and create image. Limitation Knowledge of the circuit board; risk to damage the device or revert back to original functioning status; fail	Limitation Data partition is encrypted with AES-256 algorithm using unique key tied to the hardware which render off-line browsing of data unrealistic.	Usage Direct access to the internal memory and create image; bitlocker encryption is turned off by default. Limitation Knowledge of the circuit board; risk to damage the device or

	to interpret encrypted disk volume.		revert back to original functioning status.
Microread	Usage Use microscope to view the physical memory cells. Limitation Expert skill required; very costly and time-consuming – impractical for full memory extraction.	Limitation Data partition is encrypted with AES-256 algorithm using unique key tied to the hardware which render off-line browsing of data unrealistic.	Usage Use microscope to view the physical memory cells. Limitation Expert skill required; very costly and time-consuming – impractical for full memory extraction.

Method preferred	Physical Extraction if custom Boot Loader is available because of:
	non-invasive;
	most data yielded;
	data integrity preserved.
		iTunes backup located at synchronized computer because of:
		non-invasive
		data integrity preserved.
			JTAG if the required ports on the circuit board are known because of:
			tools in hand
			most data yielded;
			data integrity preserved.

Table 3-1 Evaluation of different Extraction Methods

Chapter 4 Experiments conducted

4.1 General principles

As described above, there is no standardized method of how data can be extracted from a smartphone. Unlike conducting computer forensics on a desktop computer, the forensic examiner will simply remove the hard drive, attach it to a disk cloner via a write blocker device and make a forensics image of the whole hard drive for data analysis. Mobile phone forensics, on the contrary, operates on the mobile device which can manipulate data by itself when powered up. Furthermore, the extraction method can be much more complicated due to its use of proprietary hardware, OS and security setting.

It is not disputed that physical access can yield a larger amount of information but it also increases the risk of damaging the device and digital evidence contained therein. As discussed before, there are different approaches that can acquire a forensic duplicate of mobile device at a physical level. Some forensic tools transfer and run a software agent on the target device. The trustworthiness of digital evidence acquired has to be proved as the alien software (the source codes of these tools are usually unavailable) may be accused of having contaminated user data's integrity since examiners are not certain about the side effects of the extraction agent when collecting data from the phone. In some smartphones, an examiner can divert the boot process via a custom boot loader to gain root access to the phone memory. The more advanced methodology on acquisition is to gain access at a hardware level, either through the JTAG interface or by reading the Flash memory chips direct.

The experiments are carried out in two stages. The first stage is to identify the best acquisition method for different smartphone OSs which include iOS, Android and Windows Mobile system. The second stage is to examine the extracted data and compare the test results obtained by commercial tools with regard to the completeness and relevancy of acquired data.

In order to achieve the defined objective, general principles adopted for the experiments include:

- Physical extraction of full memory data from the experiment device is preferred if not possible.
- The integrity of the data contained in the experiment device must remain intact after the forensics process.

4.2 Experiment on iTunes Backup Analysis

So far as iPhone 4S is released, its security mechanisms do not permit a forensic examiner to extract physical image by means of a stock device without first gaining privileged access. The iPhone has a standard directory structure in which various files are stored internally on the device. Through SSH, it is possible to image a jailbroken device to communicate with the file system and extract the data therein. Indeed the ability to image the iPhone can pull off both allocated and unallocated data to become crucial evidence in an investigation. Deleted text messages, photos, or videos can often make or break a case, thus making a physical acquisition of the user data partition an ideal solution to the forensic process. However, from the perspective of law enforcement agencies, injecting program code to change the current status of mobile phone exhibit is regarded as intrusive to the phone content and by modifying the device configuration for acquisition, there is a possible risk to invalidate the evidence in court even though all the processes are always well-documented. After due consideration, the prevailing iTunes backup acquisition on non-jailbroken iPhone is selected for experiment.

4.2.1 The purpose of this experiment

- Evaluate the extraction result between iTunes backup and Cellebrite UFED (Universal Forensic Extraction Device) Touch to find out if the commercial tool produces better result in the extraction and analysis of data.

- Examine information from iTunes backup manually and check if there is any data relevant to the forensic process which has not been picked up by the Cellebrite Physical Analyzer.

Cellebrite UFED Touch is an expansive and well-known forensic tool used in more than 60 countries. It is a hardware-based device for extracting data from mobile phones. It supports physical, file system and logical extraction of data. If the targeted device is on the supported device list, the software has an auto-detection mechanism to provide a step-by-step guide for the extraction process. If not, Cellebrite UFED Touch may still have a generic profile to support unlisted devices. The phone memory is captured into the binary files that are later read and decoded using Cellebrite UFED Physical Analyzer [36]. A comprehensive and consolidated analysis report will then be generated automatically. Basically, the report displays all phone records, multimedia files and applications data which the manufacturer sees fit to meet the expectation by most forensic examiners in general circumstances. The vendor also launches the software's updates almost every month to catch up with the latest development on mobile phone forensics.

4.2.2 Execution of the experiment

Hardware: iPhone 5S 32GB memory running iOS 8.4.1 without jailbreaking

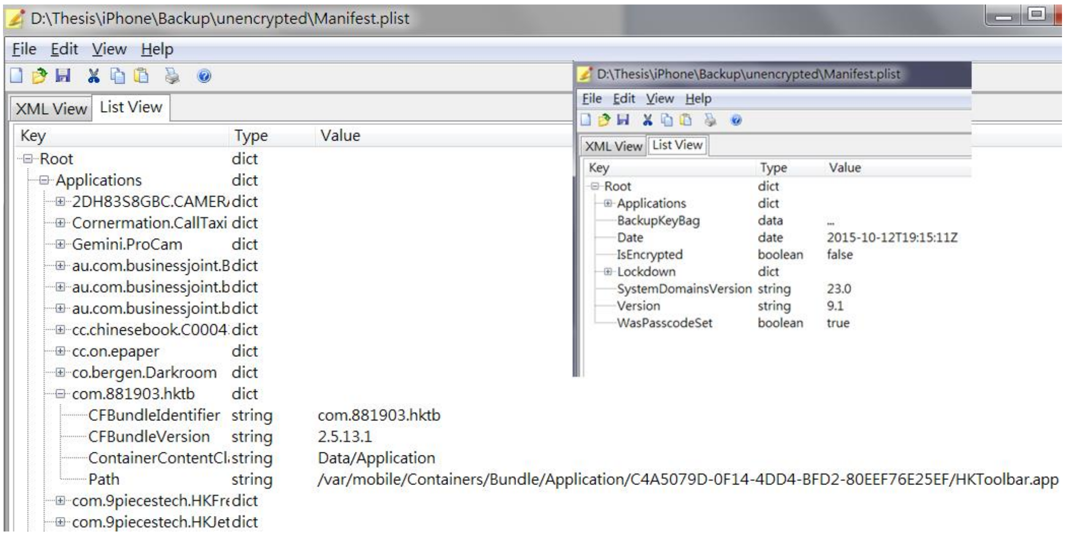
Software: iTunes 12.3.0.44 and Cellebrite UFED Physical Analyzer 4.2.6.4

As discussed earlier, physical acquisition of data from newer version of iPhone is technically difficult unless jailbreaking or alteration of part of the operation system is carried out. Therefore, the best chance of obtaining a forensically sound image is to look for a computer workstation that has synchronized with the targeted iPhone using iTunes backup. A user has the option to create backup files once the device is connected to a computer. The backup is automatically initiated during the synchronizing process or when an update or restore is performed. The backup files are stored in a specific location, depending on the type of OS running on the computer workstation. For the sake of experiment, a workstation installed with Microsoft Windows 7 was used and the location of the backup files was found in here:

C:\Users\userA\AppData\Roaming\Apple Computer\MobileSync\Backup\

After the backup process, all applications and related data files were stored in the designated location. Having reviewed these files, it was found that their filenames were unusual in that they were made out by a combination of SHA-1 hash value with the original filename, together with the path and home domain. The followings were exceptions:

- Status.plist provided the status of the last backup.
- Info.plist contained basic information about the device in general, including device name, build version, Internation Mobile Equipment Identity (IMEI), phone number, etc.
- Manifest.plist provided a list of all the applications, backup key bag and lockdown certificate.
- Manifest.mbdb was a database of filenames stored in the backup folder.



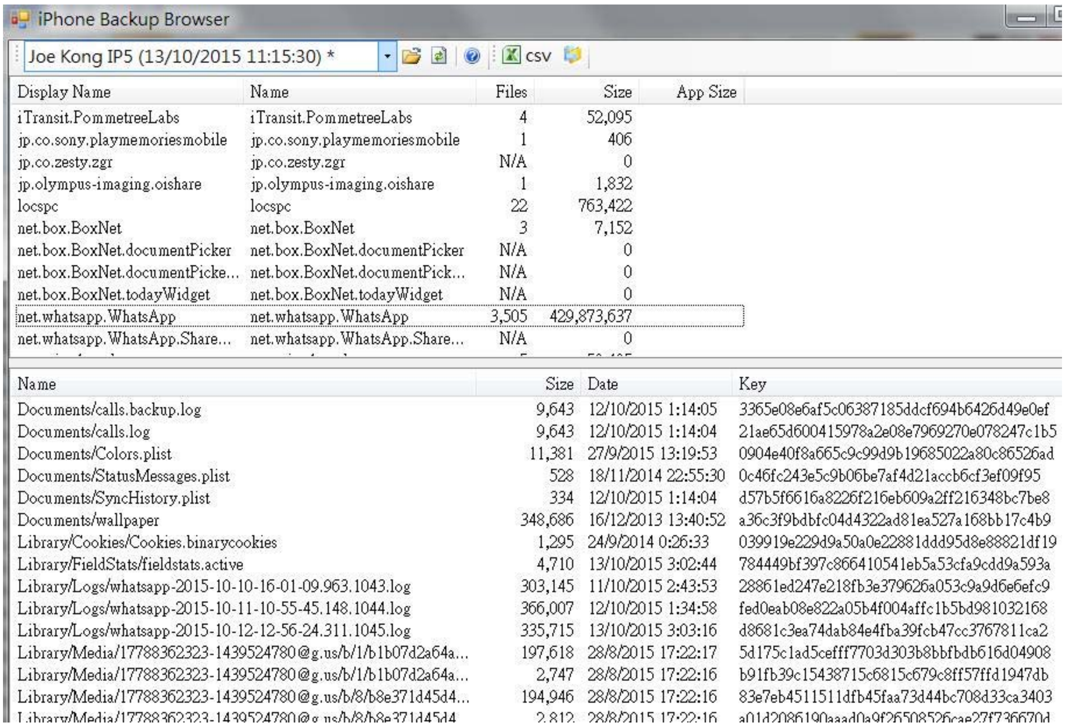
Key	Type	Value
Root	dict	
Applications	dict	
2DH83S8GBC.CAMER	dict	
Cornermation.CallTaxi	dict	
Gemini.ProCam	dict	
au.com.businessjoint.B	dict	
au.com.businessjoint.b	dict	
au.com.businessjoint.b	dict	
cc.chinesebook.C0004	dict	
cc.on.epaper	dict	
co.bergen.Darkroom	dict	
com.881903.hktb	dict	
CFBundleIdentifier	string	com.881903.hktb
CFBundleVersion	string	2.5.13.1
ContainerContentCl	string	Data/Application
Path	string	/var/mobile/Containers/Bundle/Application/C4A5079D-0F14-4DD4-BFD2-80EEF76E25EF/HKToolbar.app
com.9piecestech.HKFr	dict	
com.9piecestech.HKJet	dict	

Key	Type	Value
Root	dict	
Applications	dict	
BackupKeyBag	data	...
Date	date	2015-10-12T19:15:11Z
IsEncrypted	boolean	false
Lockdown	dict	
SystemDomainsVersion	string	23.0
Version	string	9.1
WasPasscodeSet	boolean	true

Figure 4-1. Screen capture of Manifest.plist

Property List (plist) file is encoded using Unicode UTF-8 Encoding and structured as XML. It contains essential information on configuration and holds key values for a bundle of applications. A plist editor was used directly to examine the content of the first three files (Figure 4-1 illustrates an example of Manifest.plist file). In respect of other application executable, images and documents such as databases or logs and other temporary files were contained in the backup folder. Appropriate software such as SQLite database browser, WordPad or other image viewers was used to conduct the examination.

As all the applications and data files in the backup folder were renamed in SHA1 hash value of that particular file. In order to examine the file content, the examiner needed to use appropriate software to view the database or log file. Firstly he had to find out the original file names from those mysterious names in SHA1 hash value. One of the methods was to use an executable “mbdbdump.exe” to convert the manifest.mbdb into ASCII text file (command to be executed: mbdbdump.exe > backupfilelist.txt.) To make the task easier, another open source program “iPhone Backup Browser” was used to inspect the iTunes backup folder and interpret the filenames contained therein [57]. After running the program, the file explorer had translated the filenames automatically and grouped together the segmented data according to the package name. An organized tree was then displayed. Each file was appended with an appropriate file extension and a link to the physical file in the backup folder. This process was set up to facilitate the navigation by a forensic examiner so that data could be retrieved accurately and effectively. An example showing the WhatsApp package was exhibited in Figure 4-2.



iPhone Backup Browser

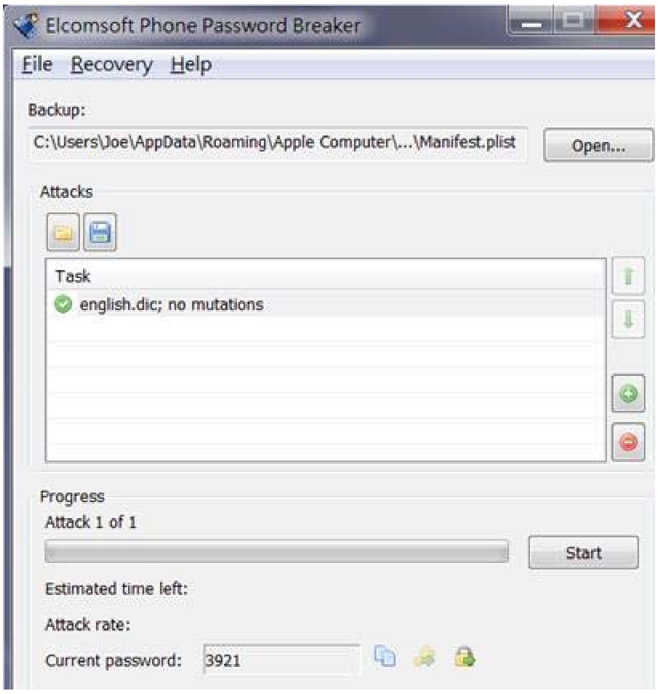
Joe Kong IP5 (13/10/2015 11:15:30) *

Display Name	Name	Files	Size	App Size
iTransit.PommetreeLabs	iTransit.PommetreeLabs	4	52,095	
jp.co.sony.playmemoriesmobile	jp.co.sony.playmemoriesmobile	1	406	
jp.co.zesty.zgr	jp.co.zesty.zgr	N/A	0	
jp.olympus-imaging.oishare	jp.olympus-imaging.oishare	1	1,832	
locspc	locspc	22	763,422	
net.box.BoxNet	net.box.BoxNet	3	7,152	
net.box.BoxNet.documentPicker	net.box.BoxNet.documentPicker	N/A	0	
net.box.BoxNet.documentPicke...	net.box.BoxNet.documentPick...	N/A	0	
net.box.BoxNet.todayWidget	net.box.BoxNet.todayWidget	N/A	0	
net.whatsapp.WhatsApp	net.whatsapp.WhatsApp	3,505	429,873,637	
net.whatsapp.WhatsApp.Share...	net.whatsapp.WhatsApp.Share...	N/A	0	

Name	Size	Date	Key
Documents/calls.backup.log	9,643	12/10/2015 1:14:05	3365e08e6af5c06387185ddcf694b6426d49e0ef
Documents/calls.log	9,643	12/10/2015 1:14:04	21ae65d600415978a2e08e7969270e078247c1b5
Documents/Colors.plist	11,381	27/9/2015 13:19:53	0904e40f8a665c9c99d9b19685022a80c86526ad
Documents/StatusMessages.plist	528	18/11/2014 22:55:30	0c46fc243e5c9b06be7af4d21accb6cf3ef09f95
Documents/SyncHistory.plist	334	12/10/2015 1:14:04	d57b5f6616a8226f216eb609a2ff216348bc7be8
Documents/wallpaper	348,686	16/12/2013 13:40:52	a36c3f9bdbfc04d4322ad81ea527a168bb17c4b9
Library/Cookies/Cookies.binarycookies	1,295	24/9/2014 0:26:33	039919e229d9a50a0e22881ddd95d8e88821df19
Library/FieldStats/fieldstats.active	4,710	13/10/2015 3:02:44	784449bf397c866410541eb5a53cfa9cdd9a593a
Library/Logs/whatsapp-2015-10-10-16-01-09.963.1043.log	303,145	11/10/2015 2:43:53	28861ed247e218fb3e379626a053c9a9d6e6efc9
Library/Logs/whatsapp-2015-10-11-10-55-45.148.1044.log	366,007	12/10/2015 1:34:58	fed0eab08e822a05b4f004affc1b5bd981032168
Library/Logs/whatsapp-2015-10-12-12-56-24.311.1045.log	335,715	13/10/2015 3:03:16	d8681c3ea74dab84e4fba39fcb47cc3767811ca2
Library/Media/17788362323-1439524780@g.us/b/1/b1b07d2a64a...	197,618	28/8/2015 17:22:17	5d175c1ad5cefff7703d303b8bbfbdb616d04908
Library/Media/17788362323-1439524780@g.us/b/1/b1b07d2a64a...	2,747	28/8/2015 17:22:16	b91fb39c15438715c6815c679c8ff57ffd1947db
Library/Media/17788362323-1439524780@g.us/b/8/b8e371d45d4...	194,946	28/8/2015 17:22:16	83e7eb4511511dfb45faa73d44bc708d33ca3403
Library/Media/17788362323-1439524780@g.us/b/8/b8e371d45d4	2,812	28/8/2015 17:22:16	a01d2086190aaad0a9f26508526cae27f736670d

Figure 4-2. Display of WhatsApp package using iPhone Backup Browser

A user can also initiate an encrypted backup by entering a password prior to synchronizing the backup files. Since the release of iOS 4, the keychain file that contains user name and password is encrypted by hardware keys stored on the iPhone if the backup is not password-protected. Otherwise, the keychain file will be encrypted by employing software keys generated from the backup password and there is possibility to retrieve encrypted data stored in the keychain file. Currently there is no software available to read an encrypted backup. As an alternative method, the Elcomsoft Phone Password Breaker had recovered the password by running brute-force attack to the backup file "manifest.plist" as illustrated in Figure 4-3 [58].



Backup:
C:\Users\Joe\AppData\Roaming\Apple Computer\...\Manifest.plist

Task
english.dic; no mutations

Attack 1 of 1

Estimated time left:

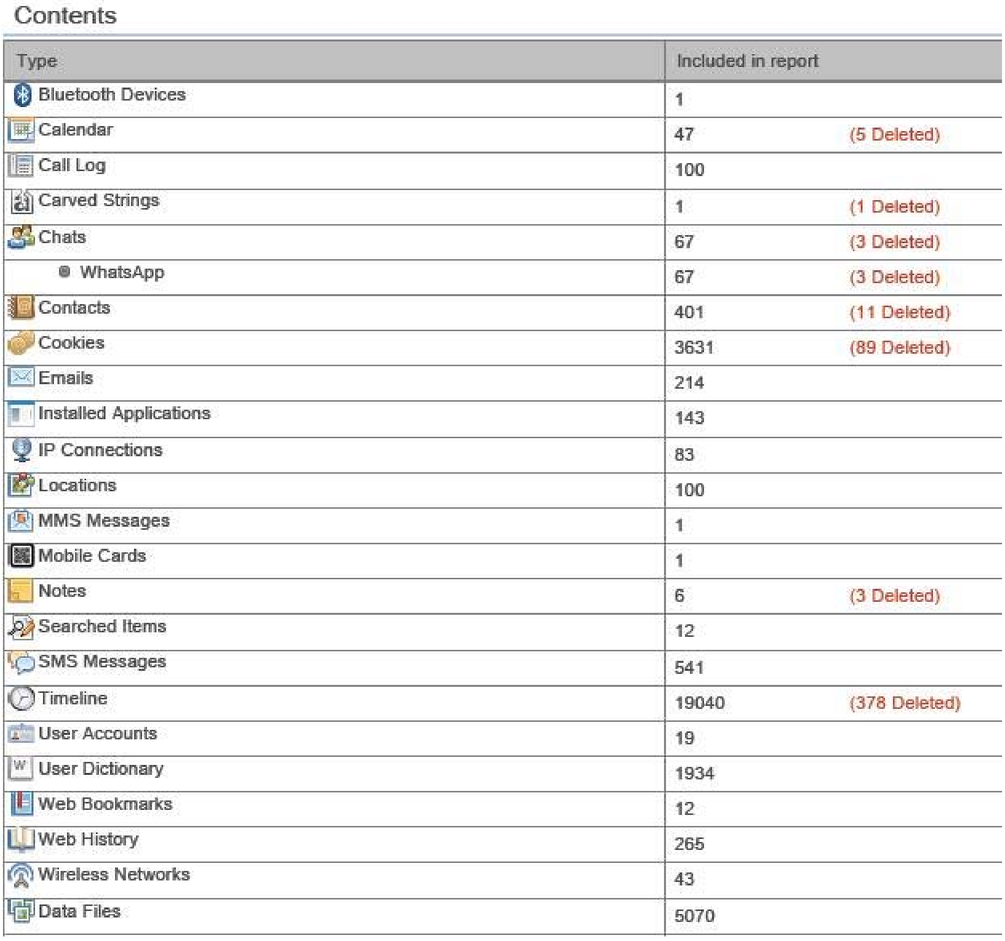
Attack rate:

Current password: 3921

Figure 4-3. Use of Elcomsoft Phone Password Breaker to recover password

The aforementioned software used the retrieved password automatically to decrypt files contained in the backup folder and copied all decrypted files to another folder as specified by the user for storage and further examination.

Meanwhile, a file system acquisition was conducted by Cellebrite UFED Touch which operated the supported profile listed in the software. A comprehensive report was generated automatically and shown in Figure 4-4 after the Cellebrite Physical Analyzer had completed the decoding process from the extracted data.



Contents

Type	Included in report	
Bluetooth Devices	1	
Calendar	47	(5 Deleted)
Call Log	100	
Carved Strings	1	(1 Deleted)
Chats	67	(3 Deleted)
● WhatsApp	67	(3 Deleted)
Contacts	401	(11 Deleted)
Cookies	3631	(89 Deleted)
Emails	214	
Installed Applications	143	
IP Connections	83	
Locations	100	
MMS Messages	1	
Mobile Cards	1	
Notes	6	(3 Deleted)
Searched Items	12	
SMS Messages	541	
Timeline	19040	(378 Deleted)
User Accounts	19	
User Dictionary	1934	
Web Bookmarks	12	
Web History	265	
Wireless Networks	43	
Data Files	5070	

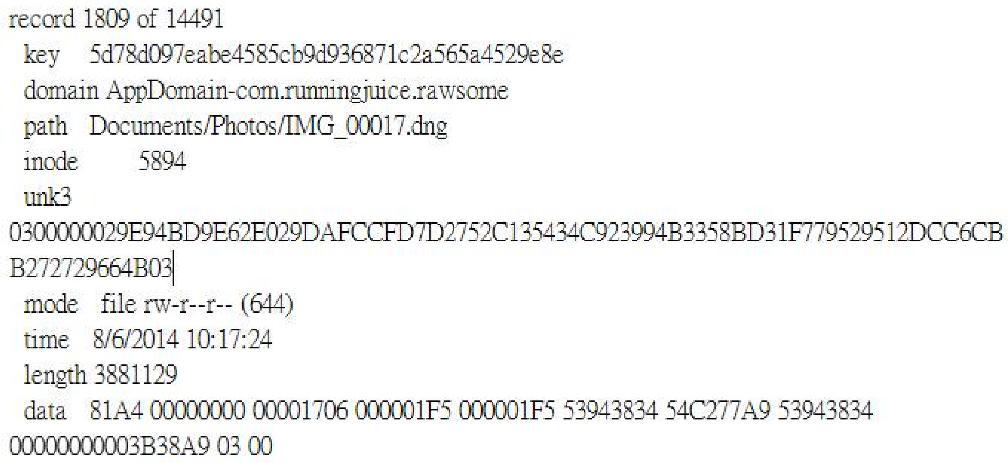
Figure 4-4. Report generated by Cellebrite Physical Analyzer

4.2.3 Evaluation for the experiment

Many of the mobile forensics tools that support iPhone logical acquisitions will also export the information of extracted data into a report. An examiner can see the reported data but cannot view the source of that data. For example, a report may show a website being visited, but does not display the data and time of visit. In the absence of this crucial information, the examiner will need to use another tool to view the original source of information and recover any metadata associated with that file. It is more desirable if the report also allows the examiner to view the raw files from which it was derived.

Upon the provision of original iTunes backup by Apple, the acquisition process of data was completed smoothly and authenticated. Nevertheless, the synchronize process had exerted an influence on the content of the targeted phone because the transfer of application and data was bi-directional so information had been uploaded from the workstation to the mobile phone. Nevertheless, this experiment only carries a conceptual meaning. In reality a forensic examiner should search for and seize the computer which the suspect had used it to synchronize with his iPhone and capture a forensic image of the computer for examination of the backup files contained within the dedicated iTunes folder.

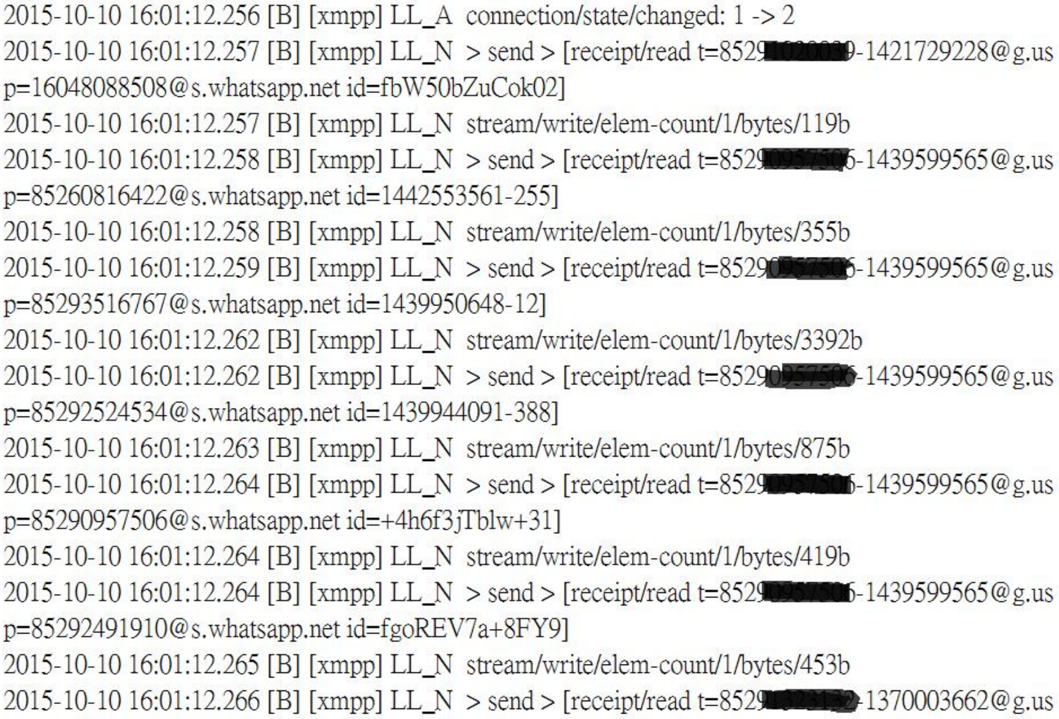
The files in the decrypted iTunes backup folder were examined and revealed that the content or records contained therein had matched with the report generated by the Cellebrite Physical Analyzer. But as the original directory structure did not exist, considerable time had been taken to consolidate the files and records to carry out the comparison. It can be concluded that the backup files produced by iTunes have provided sufficient data and information for further analysis. When reviewing the report created by Cellebrite, it was noted that some information had not been interpreted and presented. Also in examining the converted manifest.mbdb, some of the domain fields contained information of the parent application which generated the existing data file. To give an example below in Figure 4-5, the IMG_00017.dng file was generated by a software package application, com.runningjuice.rawsome, which was named as Digital Negative listed in Apple Store.



```
record 1809 of 14491
  key    5d78d097eabe4585cb9d936871c2a565a4529e8e
  domain AppDomain-com.runningjuice.rawsome
  path   Documents/Photos/IMG_00017.dng
  inode        5894
  unk3
0300000029E94BD9E62E029DAFCCFD7D2752C135434C923994B3358BD31F779529512DCC6CB
B272729664B03
  mode   file rw-r--r-- (644)
  time   8/6/2014 10:17:24
  length 3881129
  data   81A4 00000000 00001706 000001F5 000001F5 53943834 54C277A9 53943834
0000000003B38A9 03 00
```

Figure 4-5. The domain field of a graphic dng in manifest.mbdb

Besides, additional information was found when viewing each individual file manually. For example, as shown in Figure 4-6, different Whatsapp log files had stored user's activities including the time stamp e.g. sending a message, adding a new picture and exiting from a conversation. This activity log collaborated on the Whatsapp chat messages and gave a more detailed picture of incidents that had taken place.



```
2015-10-10 16:01:12.256 [B] [xmpp] LL_A  connection/state/changed: 1 -> 2
2015-10-10 16:01:12.257 [B] [xmpp] LL_N  > send > [receipt/read t=8529[redacted]-1421729228@g.us
p=16048088508@s.whatsapp.net id=fbW50bZuCok02]
2015-10-10 16:01:12.257 [B] [xmpp] LL_N  stream/write/elem-count/1/bytes/119b
2015-10-10 16:01:12.258 [B] [xmpp] LL_N  > send > [receipt/read t=8529[redacted]-1439599565@g.us
p=85260816422@s.whatsapp.net id=1442553561-255]
2015-10-10 16:01:12.258 [B] [xmpp] LL_N  stream/write/elem-count/1/bytes/355b
2015-10-10 16:01:12.259 [B] [xmpp] LL_N  > send > [receipt/read t=85290[redacted]-1439599565@g.us
p=85293516767@s.whatsapp.net id=1439950648-12]
2015-10-10 16:01:12.262 [B] [xmpp] LL_N  stream/write/elem-count/1/bytes/3392b
2015-10-10 16:01:12.262 [B] [xmpp] LL_N  > send > [receipt/read t=85290[redacted]-1439599565@g.us
p=85292524534@s.whatsapp.net id=1439944091-388]
2015-10-10 16:01:12.263 [B] [xmpp] LL_N  stream/write/elem-count/1/bytes/875b
2015-10-10 16:01:12.264 [B] [xmpp] LL_N  > send > [receipt/read t=8529[redacted]-1439599565@g.us
p=85290957506@s.whatsapp.net id=+4h6f3jTblw+31]
2015-10-10 16:01:12.264 [B] [xmpp] LL_N  stream/write/elem-count/1/bytes/419b
2015-10-10 16:01:12.264 [B] [xmpp] LL_N  > send > [receipt/read t=8529[redacted]-1439599565@g.us
p=85292491910@s.whatsapp.net id=fgoREV7a+8FY9]
2015-10-10 16:01:12.265 [B] [xmpp] LL_N  stream/write/elem-count/1/bytes/453b
2015-10-10 16:01:12.266 [B] [xmpp] LL_N  > send > [receipt/read t=8529[redacted]-1370003662@g.us
```

Figure 4-6. WhatsApp log file showing timestamp of user activities

4.2.4 Conclusion

Having gone through the well-organized and presentable report generated by Cellebrite Physical Analyzer, all the information was actually found in the iTunes backup folder. This commercial tool has the benefit of collecting related information in a timely manner and presenting them in a comprehensive and standardized way. However, the conducting of in-depth manual examination to individual file in the backup folder of an iPhone yields a little more information which the commercial tool has not covered. The commercial tool just shows the data recovered and related metadata such as creation or modification of date and time with no further exploration or linkage to other sources of information.

Indeed the extra piece of information obtained from manual examination is useful to prove the chain of occurrence of things that happened.

The second information recovered is the individual application log file which is in most circumstances will not be analyzed by commercial tools. Different from human intelligence, it is difficult to set up taxonomy to link up background operations of the application with the test result. Again, this information is helpful in proving user's activities and can explain how the final result is attained within the defined time frame. Despite this background information provides good grounds for inclusion in the final report to collaborate the findings, it has long been omitted by the commercial tools. The key message to convey here: there are a lot of information kept in the applications and related database files. Although crime investigators or forensic examiners may use them to crack down the root of the criminal event, they should understand that commercial tools can only assist in drawing out a quick, usually the last picture of relevant data for the crime case. It is strongly recommended that investigators or examiners should examine entries in the logs and contents of database files as well as property list files in order to gain a better understanding of the events that occurred. To give an example, in some occasions, when the activity logs of the FaceBook Messagers are reviewed, information may find missing in the chat messages because the user has logged out from Facebook.

In-depth analysis of file content in the mobile device is in fact similar to the examination of artifacts in the Windows registry. Many fundamental or underlying situations will tell the true story by themselves.

Having said the above, when time and manpower resources are limited as well as backlogs are accumulating, commercial tools can be utilized to provide a more effective and simple way to consolidate all important elements gathered from the device to enable the examiner in preparing a comprehensive report.

4.3 Experiment on physical acquisition of Android phone

As described in the earlier chapter, physical acquisition is able to retrieve maximum data from a smartphone. A number of forensic tools tend to use boot loader or ADB options for physical extraction. Indeed ADB has been used by a quantity of forensic tools to be a communication interface for accessing the Android phone via a computer installed with extraction software. By default, forensic examiners have no access right to the system reserved areas when first connected via the USB cable. Prior to the data extraction phase, the Android device must be made available for ‘super user’ privilege of access to modify system files, also known as “rooting the device”. After the “rooting” process, the examiner can acquire a copy of all system partitions including those that are not originally accessible. However, this method is considered invasive because the Android phone has to be powered up normally and the USB Debugging mode being turned on manually in the system menu of the phone. Therefore, if the mobile device is protected by power-on password or pattern lock, the extraction process cannot be executed.

Alternatively the mobile phone can be put into the Download mode, a state in which the Flash Memory can be formatted and reprogrammed. The Flash Memory holds all binary information which includes internal memory of the device, drivers, applications and other types of data in memory structure like Read Only Memory (ROM) and Non-Volatile Random Access Memory (NVRAM)] required for the device to boot up and function. With an unlocked boot loader, the Flash Memory can be temporarily reprogrammed in a way to establish connection between the target phone and any storage media. The above procedure is similar to computer forensics where a forensic boot disk is used to operate the cloning process for acquiring data from the target computer without affecting the content of the original hard disk. The uniqueness of this method is that there is no need to "rooting the device" or enabling the USB Debugging mode before extraction, thereby resolve the difficulty in accessing a password-protected phone. The entire process is forensically sound as it will not interfere with the internal storage of the device.

SP Flash Tools is an example of applying the Download mode [59]. It is an application that can capture memory images or binary data from the phone. Apart from erasing the phone data, it can also modify the code or data and write it back to the phone. The program employs the boot ROM kernel library ("BROM_DLL") and Download Agent ("DA") program to download, read or erase files on the target phone's flash memory via a USB port connection. In practice, SP Flash Tool reads a length of memory from the phone according to a scatter file (Figure 4-7 gives a view of it), beginning at a start address with a given length. Multiple blocks starting at different addresses can be read and copied as image files to the forensic workstation for storage.

4.3.1 The purpose of this experiment

- Evaluate the extraction result between physical acquisition method using customized bootloader and Cellebrite UFED Touch with regard to the completeness and effectiveness of the tools.
- Examine the extracted information manually and check if there is any data relevant to the crime investigation which has not been picked up by the

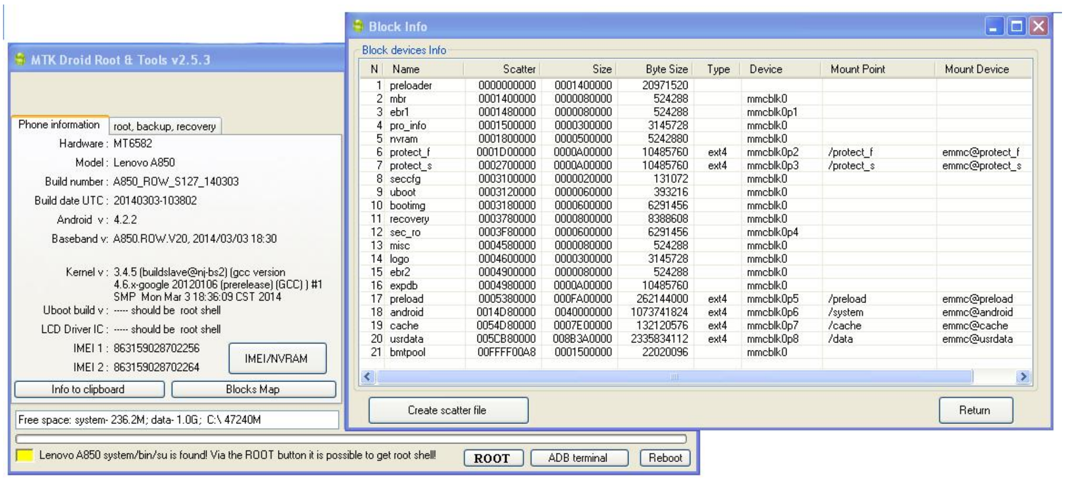
Cellebrite UFED Physical Analyzer.

4.3.2 Execution of the experiment

Hardware: Lenovo A850 mobile phone 4GB memory running Android 4.2.2 unrooted, Cellebrite UFED Touch

Software: All MTK USB Driver 2014, MTK_AllinOne_DA.bin, SP Flash Tool and Cellebrite Physical Analyser

A computer workstation installed with the required software had to be set up. The phone was powered off and connected to the workstation via a USB connection cable while the SP Flash Tool was running. A boot ROM handshake was performed and the DA was then downloaded to the targeted phone. The DA detected the target’s flash type and was in charge of transferring files from the device. The scatter file held the internal memory map of the phone. By inputting the corresponding memory address, it read back data from the loaded regions of the phone and saved as a binary file in the workstation.



Hardware : MT6582
Model : Lenovo A850
Build number : A850_ROW_S127_140303
Build date UTC : 20140303-103802
Android v : 4.2.2
Baseband v: A850.ROW.V20, 2014/03/03 18:30
Kernel v : 3.4.5 (buildslave@nj-bs2) (gcc version 4.6.x-google 20120106 (prerelease) (GCC)) #1 SMP Mon Mar 3 18:36:09 CST 2014
Uboot build v : ----- should be root shell
LCD Driver IC : ----- should be root shell
IMEI 1 : 863159028702256
IMEI 2 : 863159028702264

Free space: system- 236.2M; data- 1.0G; C:\ 47240M

Lenovo A850 system/bin/su is found! Via the ROOT button it is possible to get root shell!

N	Name	Scatter	Size	Byte Size	Type	Device	Mount Point	Mount Device
1	preloader	0000000000	0001400000	20971520				
2	mbr	0001400000	0000080000	524288		mmcblk0		
3	ebr1	0001480000	0000080000	524288		mmcblk0p1		
4	pro_info	0001500000	0000300000	3145728		mmcblk0		
5	nvram	0001800000	0000500000	5242880		mmcblk0		
6	protect_f	0001D00000	0000A00000	10485760	ext4	mmcblk0p2	/protect_f	emmc@protect_f
7	protect_s	0002700000	0000A00000	10485760	ext4	mmcblk0p3	/protect_s	emmc@protect_s
8	seccfg	0003100000	0000020000	131072		mmcblk0		
9	uboot	0003120000	0000060000	393216		mmcblk0		
10	bootimg	0003180000	0000600000	6291456		mmcblk0		
11	recovery	0003780000	0000800000	8388608		mmcblk0		
12	sec_ro	0003F80000	0000600000	6291456		mmcblk0p4		
13	misc	0004580000	0000080000	524288		mmcblk0		
14	logo	0004600000	0000300000	3145728		mmcblk0		
15	ebr2	0004900000	0000080000	524288		mmcblk0		
16	expdb	0004980000	0000A00000	10485760		mmcblk0		
17	preload	0005380000	000FA00000	262144000	ext4	mmcblk0p5	/preload	emmc@preload
18	android	0014D80000	0040000000	1073741824	ext4	mmcblk0p6	/system	emmc@android
19	cache	0054D80000	007E00000	132120576	ext4	mmcblk0p7	/cache	emmc@cache
20	usrdata	005CB80000	008B3A0000	2335834112	ext4	mmcblk0p8	/data	emmc@usrdata
21	bmtpool	00FFFF00A8	0001500000	22020096		mmcblk0		

Figure 4-7. The memory address listed in a scatter file for MT6582

The Cellebrite UFED Touch was also used to perform a physical extraction from the same Android phone. Having reviewed the phone support lists, it was found that Cellebrite UFED Touch only provided options for logical extraction to Lenovo A850. Therefore, the extraction process deployed the generic physical extraction profile for Chinese Andriod MTK device instead. This method is using ADB as a communication interface to access the Android system. The Android phone was powered up normally and the USB Debugging mode was turned on manually from the phone. The flash memory was captured into a single extraction file which included unallocated memory space and deleted data such as SMS, call logs, phonebook entries, pictures, videos and user passwords. The extraction file was later processed by Cellebrite UFED Physical Analyzer which automatically decoded all records and displayed them comprehensively in viewable formats like graphic photos, video/audio clips, call logs, contacts, messages, emails etc.

4.3.3 Evaluation of the experiment

The SP Flash Tool enables a forensic examiner to gain physical access to a target mobile device without the need to obtain a passcode, enable the USB debugging mode and gain root privilege beforehand. It takes on a boot loader approach by injecting a secure boot image to start up the mobile device which is similar to the process for acquiring an image from a computer hard disk. Besides, the correct DA for Android MTK is conveniently included in the SP Flash Tool. But as the tool is an open source application, the examiner should have obtained a scatter file for the target phone from another open source software i.e. MtkDroidTools [60]. In proving the merit of using a controlled boot program, the SP Flash Tool was tested three times consecutively to acquire image files from the target phone. Their hash values are verified and proved to be identical. It is fair to conclude that no file creation or modification has been made to the internal memory when the phone is booted up for data acquisition.

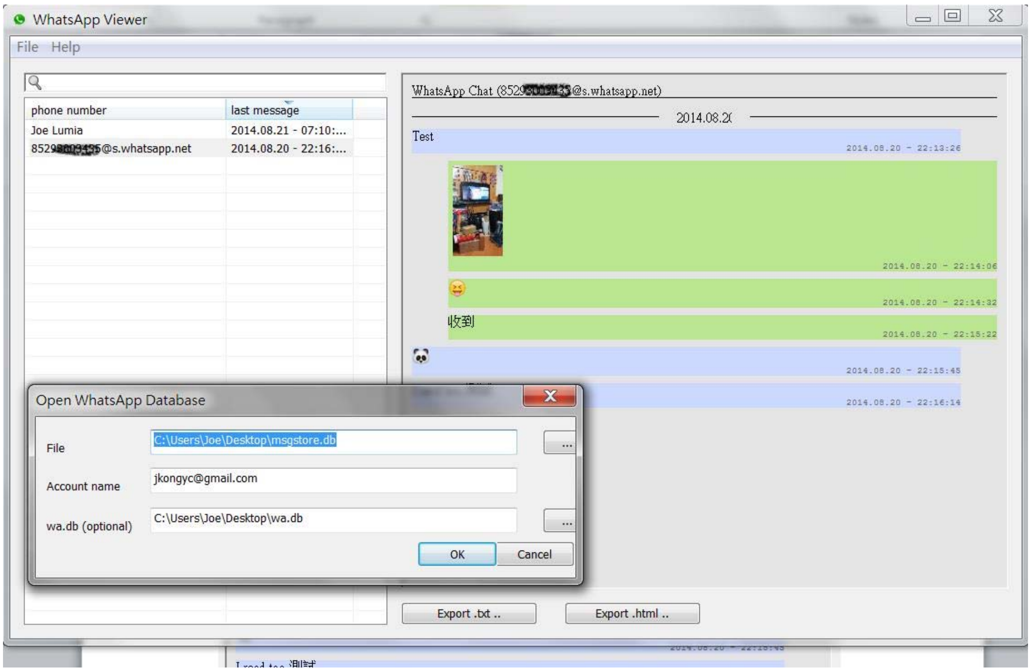
During the acquisition process, the Cellebrite UFED Touch completed the task using ADB mode on physical acquisition. However, it was found that system and application log files were created or modified whenever the phone was switched on for the extraction. To take a closer look, these files included newly created system start-up event files, modified system logs or application library files. All these files were activated as part of the system boot up process without intervention from the user.

The image files acquired using the SP Flash Tool had been mounted as a logical partition using FTK Imager, a well-known computer forensic tool [61]. File content examined by SQLite database browser, WordPad and other image viewers had revealed that the information matched with the report generated by Cellebrite Physical Analyzer. However, the Cellebrite report did not cover the backup copies of the chat messages of Whatsapp. Unlike the iPhone version, the Android version of Whatsapp has turned on the auto backup option by default which means the application will produce backup and save data in the media folder for a period of 7 days before the backup file will be purged. This file, though encrypted by default, is not examined by the majority of forensic tools which lead to a significant volume of information remains untouched.

To illustrate the above, the following steps had been performed on the Android version of Whatsapp. First of all, current chat messages were stored in plain text in the following folder:

Root/data/databases/msgstore.db

Apart from the ‘msgstore.db’, the ‘wa.db’ file displayed and linked the contact name with the phone number. To open the ‘msgstore.db’, an open source tool “Whatsapp Viewer” was used as shown in Figure 4-8 [62].



phone number	last message
Joe Lumia	2014.08.21 - 07:10:...
8529[redacted]@s.whatsapp.net	2014.08.20 - 22:16:...

WhatsApp Chat (8529[redacted]@s.whatsapp.net)

2014.08.2(

Test
2014.08.20 - 22:13:26

2014.08.20 - 22:14:06

2014.08.20 - 22:14:32

收到
2014.08.20 - 22:15:22

2014.08.20 - 22:15:45

2014.08.20 - 22:16:14

Open WhatsApp Database

File: C:\Users\Joe\Desktop\msgstore.db

Account name: jkongyc@gmail.com

wa.db (optional): C:\Users\Joe\Desktop\wa.db

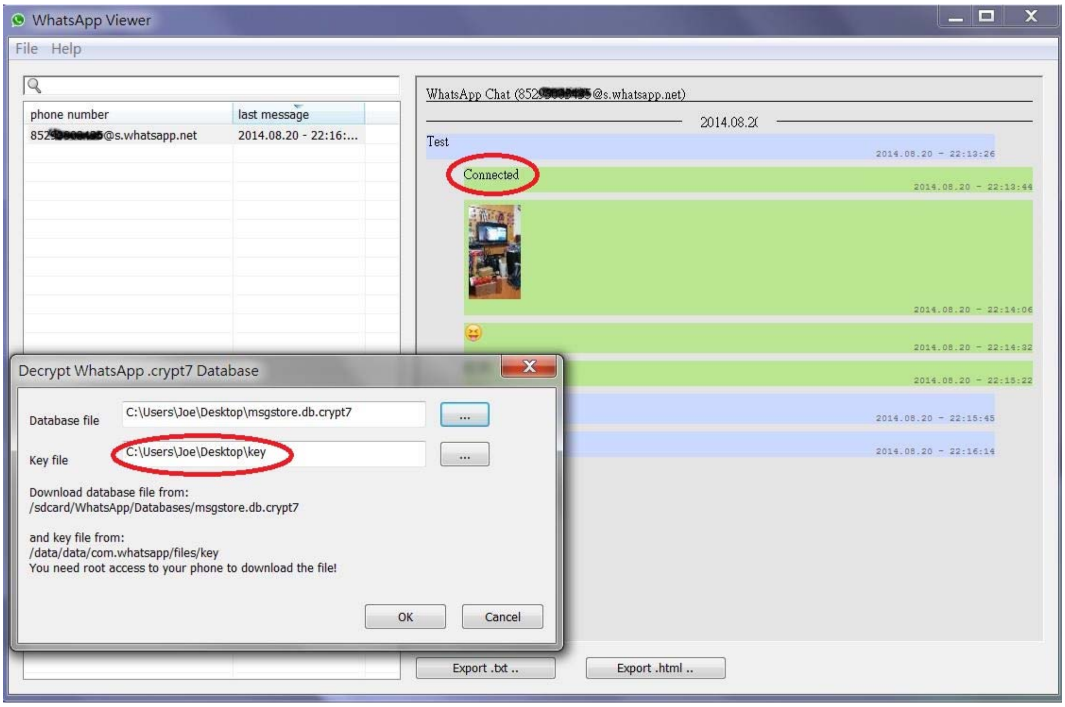
Figure 4-8. Reading WhatsApp messages using WhatsApp Viewer software

As mentioned above, the backup file for the Whatsapp Chat message was encrypted and found in the following folder:

Root/media/0/WhatsApp/Databases/msgstore.db.crypt7

The most challenging part was to acquire a “key” file to decrypt the database. The key file was located in the folder which was generally not accessible unless the user gained root privilege. But with the boot loader method, the DA was bundled with root privilege and the key file was readily available in the extracted image. The backup file was decrypted using the same Whatsapp

Viewer and generated the msgtore.decrypted.db file for further viewing.



phone number	last message
852[redacted]@s.whatsapp.net	2014.08.20 - 22:16:...

WhatsApp Chat (852[redacted]@s.whatsapp.net)

2014.08.2(

Test
2014.08.20 - 22:13:26

Connected
2014.08.20 - 22:13:44

2014.08.20 - 22:14:06

😝
2014.08.20 - 22:14:32

2014.08.20 - 22:15:22

2014.08.20 - 22:15:45

2014.08.20 - 22:16:14

Decrypt WhatsApp .crypt7 Database

Database file: C:\Users\Joe\Desktop\msgstore.db.crypt7

Key file: C:\Users\Joe\Desktop\key

Download database file from:
/sdcard/WhatsApp/Databases/msgstore.db.crypt7

and key file from:
/data/data/com.whatsapp/files/key
You need root access to your phone to download the file!

Figure 4-9. Decrypted backup file using WhatsApp Viewer and Crypt 7 key

When comparing the two pictures, it was found that the backup file had revealed one more message in the second line as shown in Figure 4-9. It had been deleted intentionally on the next day but the Cellebrite report had not covered this backup file in the decoding process.

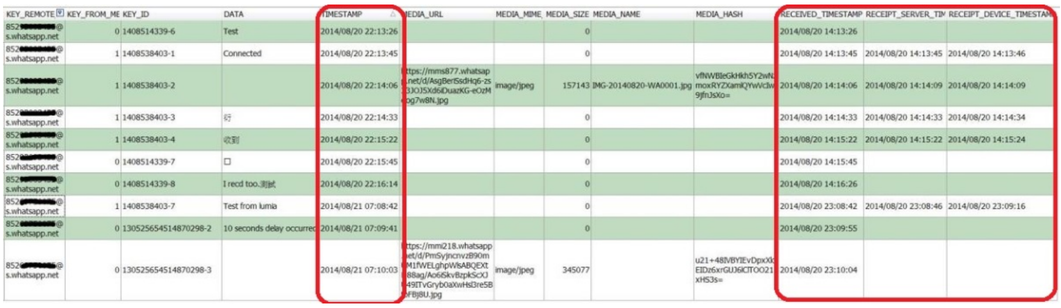
Besides, further examination on the msgstore.db file with another forensic tool, the SQLite Forensic Browser by Sanderson Forensics (this browser allows a customized view of the whole database and bundles with hex convertor for individual column) [63], it was revealed that four different fields on timestamp had been stored on the database file in Figure 4-10, namely:

- timestamp – mobile phone timestamp when sending or receiving messages

- received timestamp – the server time connected to the local side of the mobile phone

- receipt server timestamp – the server time connected to the opposite side of the mobile phone

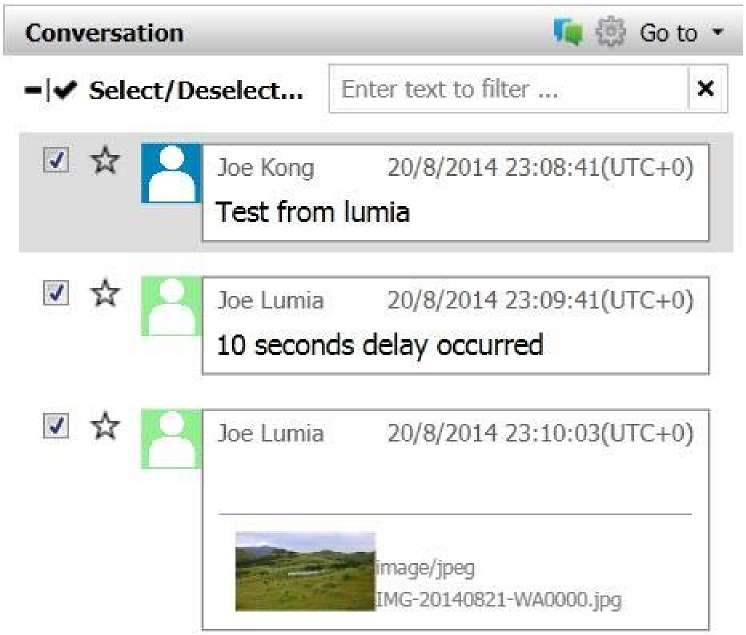
- receipt device timestamp – mobile phone timestamp on the opposite side receiving messages



KEY_REMOTE	KEY_FROM_ME	KEY_ID	DATA	TIMESTAMP	MEDIA_URL	MEDIA_MIME	MEDIA_SIZE	MEDIA_NAME	MEDIA_HASH	RECEIVED_TIMESTAMP	RECEIPT_SERVER_TIM	RECEIPT_DEVICE_TIMESTAM
852[redacted]@ s.whatsapp.net	0	1408514339-6	Test	2014/08/20 22:13:26			0			2014/08/20 14:13:26		
852[redacted]@ s.whatsapp.net	1	1408538403-1	Connected	2014/08/20 22:13:45			0			2014/08/20 14:13:45	2014/08/20 14:13:45	2014/08/20 14:13:46
852[redacted]@ s.whatsapp.net	1	1408538403-2		2014/08/20 22:14:06	ttps://mms877.whatsap .net/d/AsgBerl5sdHq6-zs 3JOJ5Xd6iDuazKG-eOzM og7w8N.jpg	image/jpeg	157143	IMG-20140820-WA0001.jpg	vfNWBIeGkHkh5Y2wN moxRYZXamIQYwVclw 9jfnJsXo=	2014/08/20 14:14:06	2014/08/20 14:14:09	2014/08/20 14:14:09
852[redacted]@ s.whatsapp.net	1	1408538403-3	行	2014/08/20 22:14:33			0			2014/08/20 14:14:33	2014/08/20 14:14:33	2014/08/20 14:14:34
852[redacted]@ s.whatsapp.net	1	1408538403-4	收到	2014/08/20 22:15:22			0			2014/08/20 14:15:22	2014/08/20 14:15:22	2014/08/20 14:15:24
852[redacted]@ s.whatsapp.net	0	1408514339-7	□	2014/08/20 22:15:45			0			2014/08/20 14:15:45		
852[redacted]@ s.whatsapp.net	0	1408514339-8	I recd too.測試	2014/08/20 22:16:14			0			2014/08/20 14:16:26		
852[redacted]@ s.whatsapp.net	1	1408538403-7	Test from lumia	2014/08/21 07:08:42			0			2014/08/20 23:08:42	2014/08/20 23:08:46	2014/08/20 23:09:16
852[redacted]@ s.whatsapp.net	0	130525654514870298-2	10 seconds delay occurre	2014/08/21 07:09:41			0			2014/08/20 23:09:55		
852[redacted]@ s.whatsapp.net	0	130525654514870298-3		2014/08/21 07:10:03	ttps://mmi218.whatsapp et/d/PmSyjncnvzB90m M1fWELghpWsABQEXt 88ag/Ao6iSkvBzpkScXJ 49ITvGryb0aXwHsl3re5B FBj8U.jpg	image/jpeg	345077		u21+48IVBYIEvDpxXk EIDz6xrGUJ6iCITOO21 xHS3s=	2014/08/20 23:10:04		

Figure 4-10. Viewing msgstore.db file using SQLite Forensic Browser

When examining the Cellebrite report in Figure 4-11, only the timestamps of receiving and sending message in the local machine were displayed.



Conversation

Go to

Select/Deselect...

Enter text to filter ...

☑ ☆ Joe Kong 20/8/2014 23:08:41(UTC+0)

Test from lumia

☑ ☆ Joe Lumia 20/8/2014 23:09:41(UTC+0)

10 seconds delay occurred

☑ ☆ Joe Lumia 20/8/2014 23:10:03(UTC+0)

image/jpeg
IMG-20140821-WA0000.jpg

Figure 4-11. WhatsApp Timestamp showing in Cellebrite report

4.3.4 Conclusion

A French scientist, Locard introduced an eminent analysis concept (Locard’s exchange principle) [64] which showed that an interaction between two objects (one being the mobile device OS) could result in the transfer or creation of data. So when a user logs into the OS, artifacts of the login as well as the user’s activities are created. There will be traces of transfer or creation of data of a program which runs within the system. The majority of these artifacts or records only exist for a very short period of time while some may persist until the system is rebooted.

The test result revealed in the above experiment is crucial for crime investigations. To give an example, the police conducts a raid to a gambling den and before breaking into the premises, the operator has deleted Whatsapp messages which contained bets submitted by customers. Since the backup file for previous Whatsapp messages cannot be easily altered or destroyed in short moments, it is possible for the examiner, by using appropriate forensic tool, to retrieve deleted file as evidence to show the gambling activities do take place.

As to different timestamp records contained in the Whatsapp database, they are useful for cross referencing with other mobile phone seizures which belonged to the opposite party or participants in the chat group. The record of the receipt device timestamp is a piece of good evidence to prove and confirm the timeline that such communication has taken place.

The development and support to cater for bundled applications such as contact list, call record, browser history, email and SMS has reached maturity in Cellebrite Physical Analyzer since there are less structural change in the newly launched Android OS. Nevertheless, with regard to the huge number of third party applications (over 1.6 million since July 2015 [11]), it is apparent that even the most reputable commercial tool will have a hard time to trace after the new applications for decoding the data contained therein.

4.4 Experiment on JTAG acquisition of Windows smartphone

In the last chapter, it is mentioned that Windows Phone 8.1 deploys secure boot to ensure the integrity of the entire OS. This security feature refers to the provision of keys to enforce the trusted boot loader and also the digital signatures signed by Microsoft to prevent unauthorized applications from installing on the phone. As every layer is in control, it becomes very complex to integrate a non-certified process or a custom boot loader to extract physical memory. Under the circumstances, this experiment is to carry out physical acquisition of data by using JTAG method instead.

4.4.1 The purpose of this experiment

- Evaluate the extraction result between physical acquisition methods conducted by JTAG and Cellebrite UFED Touch respectively and compare the completeness and effectiveness of the two tools.
- Examine the extracted information manually and check if there is any data relevant to the crime investigation that has not been picked up by the Cellebrite Physical Analyzer.

4.4.2 Execution of the experiment

Hardware: Nokia Lumia 520 mobile phone, 8GB memory running Windows Phone OS 8.1, RIFF Box, Cellebrite UFED Touch

Software: RIFF Box JTAG Manager with a dedicated DLL file for Nokia Lumia 520 (resurrector file) and Cellebrite Physical Analyser

The phone model was chosen because of its compatibility with the acquisition toolkit used in the experiment. Firstly, the phone was disassembled to gain access to the JTAG pads on the printed circuit board as shown in Figure 4-12.

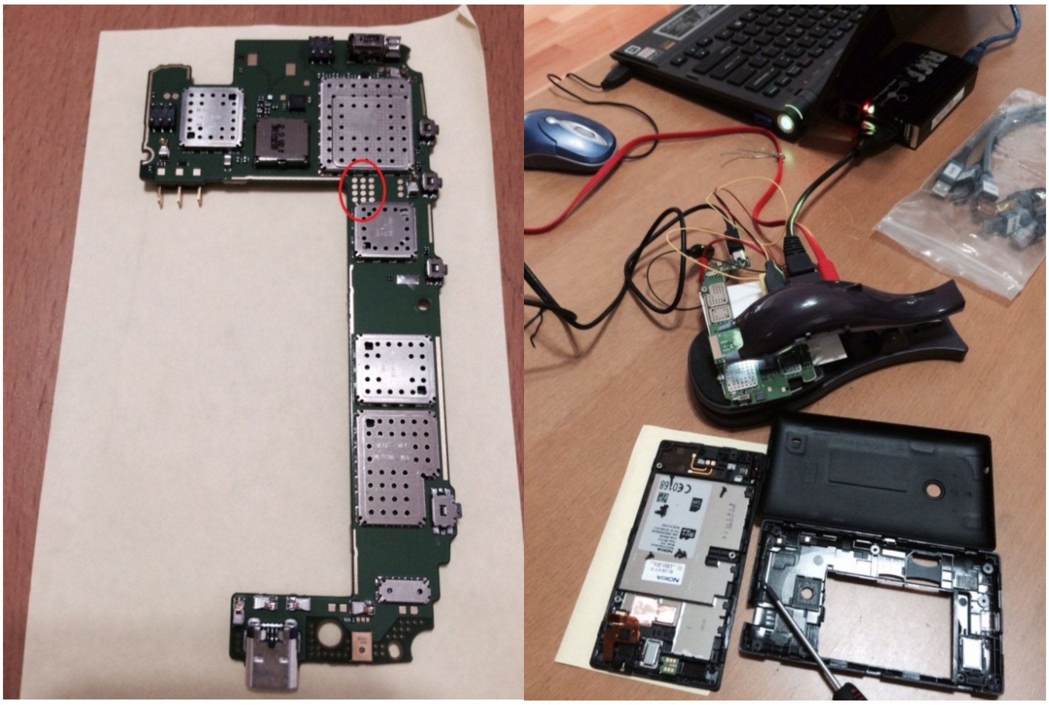


Figure 4-12. Use of RIFF Box to JTAG Nokia Lumia 520

Then, a special JTAG adaptor or "jig" was connected to the target phone and the JTAG hardware. In the experiment, the “Dolpin” JTAG adaptor was equipped with pre-soldered pogoes that matched up with the JTAG pads (Figure 4-13). No soldering was required and the risk of damaging the circuit board became null, especially when the work was conducted by a forensic examiner but not an electronic engineer.

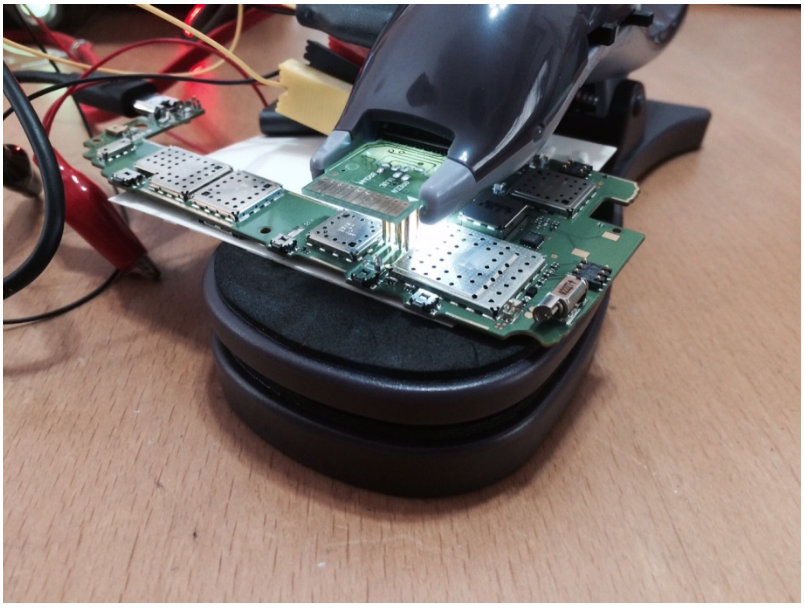
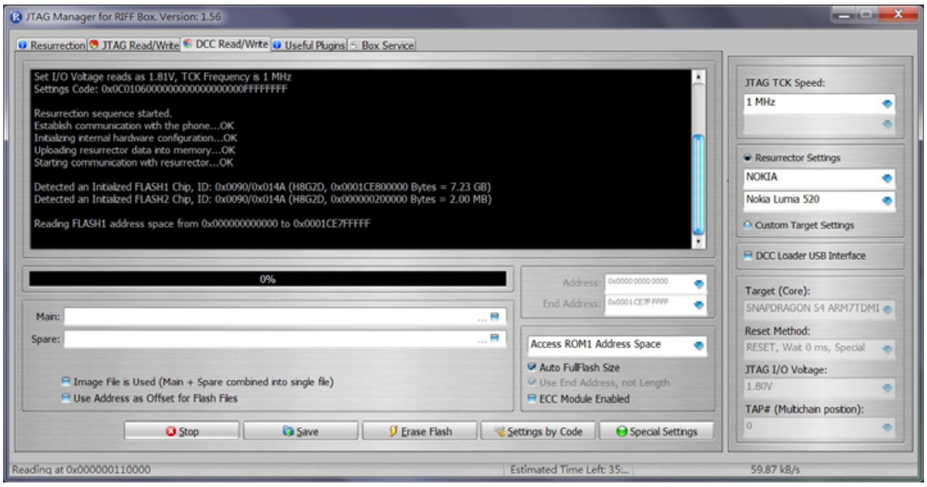


Figure 4-13. A closer look to view the "jig" adapter

The JTAG software used was RIFF Box which had a circuit board diagram to help identifying the location of JTAG pads and the specific pinout assignment for data extraction [65]. The phone model Nokia 520 Lumia was selected in the Resurrector settings of the RIFF Box software and the custom DLL file for configuration was loaded as illustrated in Figure 4-14. A DCC Read option was chosen to extract a binary image of the physical memory of the phone.



```
Set I/O Voltage reads as 1.81V, TCK Frequency is 1 MHz
Settings Code: 0x0C0106000000000000000000FFFFFFFF

Resurrection sequence started.
Establish communication with the phone...OK
Initializing internal hardware configuration...OK
Uploading resurrector data into memory...OK
Starting communication with resurrector...OK

Detected an Initialized FLASH1 Chip, ID: 0x0090/0x014A (H8G2D, 0x0001CE800000 Bytes = 7.23 GB)
Detected an Initialized FLASH2 Chip, ID: 0x0090/0x014A (H8G2D, 0x000000200000 Bytes = 2.00 MB)

Reading FLASH1 address space from 0x000000000000 to 0x0001CE7FFFFF
```

Main:
Spare:
□ Image File is Used (Main + Spare combined into single file)
□ Use Address as Offset for Flash Files

Address: 0x0000 0000 0000
End Address: 0x0001 CE7F FFFF
Access ROM1 Address Space
☑ Auto FullFlash Size
☑ Use End Address, not Length
□ ECC Module Enabled

JTAG TCK Speed: 1 MHz
● Resurrector Settings
NOKIA
Nokia Lumia 520
○ Custom Target Settings
□ DCC Loader USB Interface
Target (Core): SNAPDRAGON S4 ARM7TDMI
Reset Method: RESET, Wait 0 ms, Special
JTAG I/O Voltage: 1.80V
TAP# (Multichain position): 0

Reading at 0x000000110000 | Estimated Time Left: 35:... | 59.87 kB/s

Figure 4-14. The JTAG Manager is connecting to Nokia Lumia 520

The JTAG method was found effectively bypassing the security mechanism imposed by Microsoft and a full memory dump in the size of 7.23GB was extracted in 27 hours. The FTK imager was then used to mount the binary file for examination.

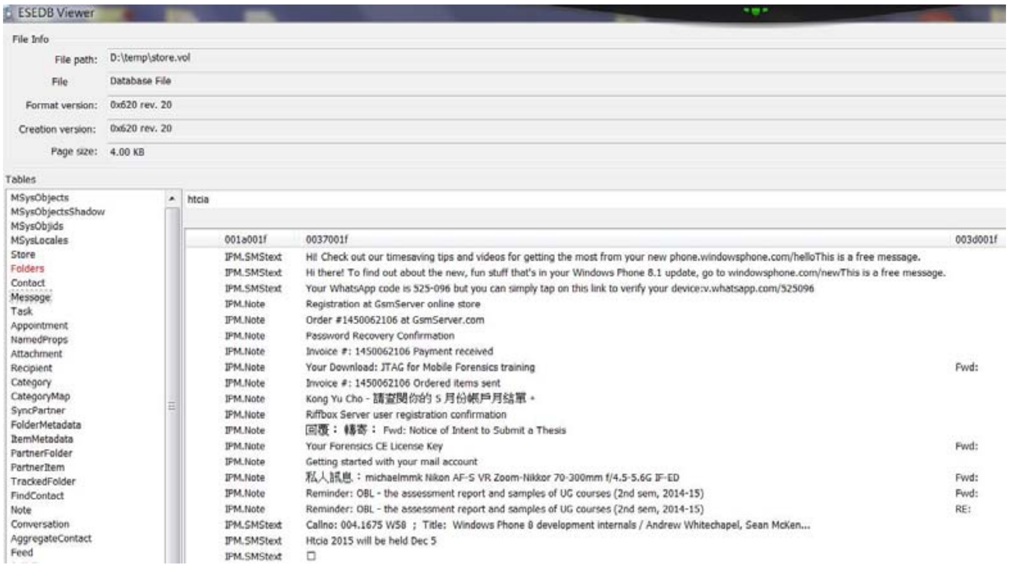
The Cellebrite UFED Touch was also used to perform a physical extraction from the same Windows phone. According to the Cellebrite's web page, Cellebrite UFED Touch is the first in the forensic industry to support physical extraction on Windows Phone 8.1 which runs on limited models of devices including Nokia Lumia 520. However, no information is provided as to how it can divert the device’s secure boot process, a technical development which Microsoft is proud of. According to the Cellebrite report generated, the information retrieved was relatively little when comparing the examination reports previously conducted on iPhones and Android phones.

4.4.3 Evaluation of the experiment

If soldering to the JTAG pads is required, the connecting of leads to the RIFF box can be technical and risky because it may damage the circuit board if excessive solders are used for bridging the pads. The correct voltage is also important when powering up the PCB. Too little voltage makes the board unstable while too much voltage will otherwise cause permanent damage. A good tip is to refer to the battery voltage specification listed on the battery label.

Image files acquired had been mounted as a logical partition using the FTK Imager with file content being reviewed by ESE database browser, WordPad and other image viewers. The information obtained had matched with the report generated by the Cellebrite Physical Analyzer. Call Logs were stored in the phone.vol while the SMS, contacts and emails were stored in the store.vol (Figure 4-15 refers) which is located at:

Users\WPCOMMSSERVICES\APPDATA\Local\Unistore\



ESEDB Viewer

File Info

File path: D:\temp\store.vol
File: Database File
Format version: 0x620 rev. 20
Creation version: 0x620 rev. 20
Page size: 4.00 KB

Tables

MSysObjects
MSysObjectsShadow
MSysObjids
MSysLocales
Store
Folders
Contact
Message
Task
Appointment
NamedProps
Attachment
Recipient
Category
CategoryMap
SyncPartner
FolderMetadata
ItemMetadata
PartnerFolder
PartnerItem
TrackedFolder
FindContact
Note
Conversation
AggregateContact
Feed

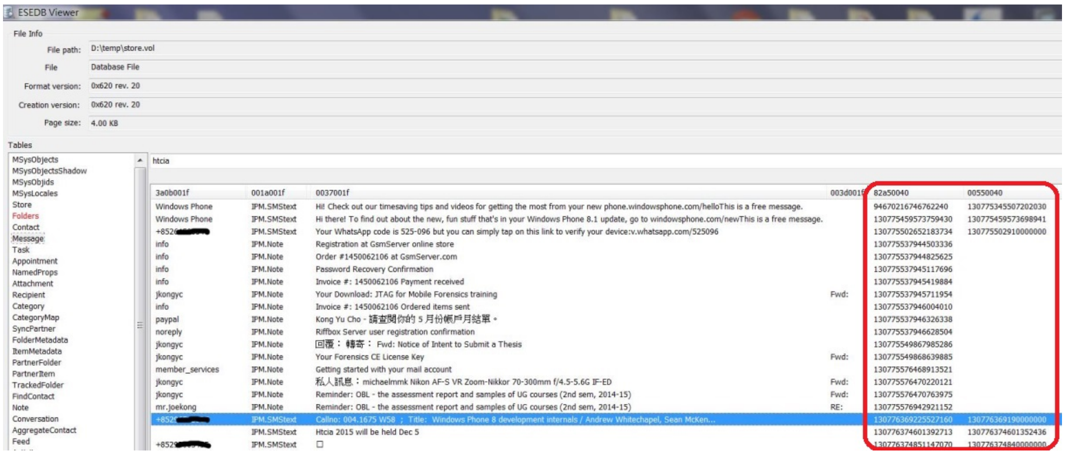
htcia

001a001f	0037001f	003d001f
IPM.SMStext	Hi! Check out our timesaving tips and videos for getting the most from your new phone.windowsphone.com/helloThis is a free message.	
IPM.SMStext	Hi there! To find out about the new, fun stuff that's in your Windows Phone 8.1 update, go to windowsphone.com/newThis is a free message.	
IPM.SMStext	Your WhatsApp code is 525-096 but you can simply tap on this link to verify your device:v.whatsapp.com/525096	
IPM.Note	Registration at GsmServer online store	
IPM.Note	Order #1450062106 at GsmServer.com	
IPM.Note	Password Recovery Confirmation	
IPM.Note	Invoice #: 1450062106 Payment received	
IPM.Note	Your Download: JTAG for Mobile Forensics training	Fwd:
IPM.Note	Invoice #: 1450062106 Ordered items sent	
IPM.Note	Kong Yu Cho - 請查閱你的 5 月份帳戶月結單。	
IPM.Note	Riffbox Server user registration confirmation	
IPM.Note	回覆： 轉寄： Fwd: Notice of Intent to Submit a Thesis	
IPM.Note	Your Forensics CE License Key	Fwd:
IPM.Note	Getting started with your mail account	
IPM.Note	私人訊息：michaelmmk Nikon AF-S VR Zoom-Nikkor 70-300mm f/4.5-5.6G IF-ED	Fwd:
IPM.Note	Reminder: OBL - the assessment report and samples of UG courses (2nd sem, 2014-15)	Fwd:
IPM.Note	Reminder: OBL - the assessment report and samples of UG courses (2nd sem, 2014-15)	RE:
IPM.SMStext	Callno: 004.1675 W58 ; Title: Windows Phone 8 development internals / Andrew Whitechapel, Sean McKen...	
IPM.SMStext	Htcia 2015 will be held Dec 5	
IPM.SMStext	□	

Figure 4-15. Viewing of Message data in store.vol using ESE database browser

Alternatively, same type of data can also be retrieved by Python script for further investigation.

The SMS records extracted from the store.vol indicate that there were two timestamps (Figure 4-16 refers) which was different from the Cellebrite report as the latter could only show one timestamp.



ESEDB Viewer

File Info

File path: D:\temp\store.vol
File: Database File
Format version: 0x620 rev. 20
Creation version: 0x620 rev. 20
Page size: 4.00 KB

Tables

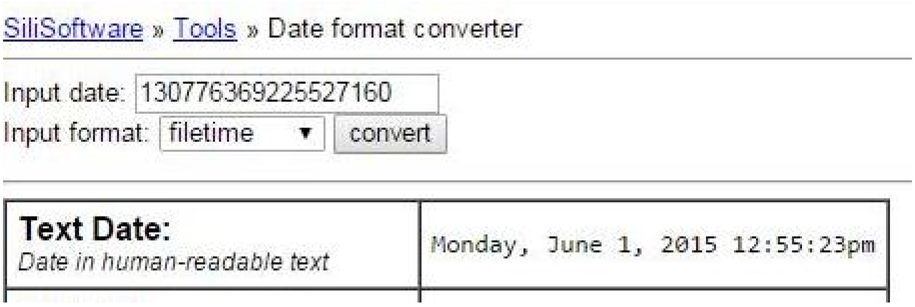
MSysObjects
MSysObjectsShadow
MSysObjids
MSysLocales
Store
Folders
Contact
Message
Task
Appointment
NamedProps
Attachment
Recipient
Category
CategoryMap
SyncPartner
FolderMetadata
ItemMetadata
PartnerFolder
PartnerItem
TrackedFolder
FindContact
Note
Conversation
AggregateContact
Feed

htcia

3a0b001f	001a001f	0037001f	003d001f	82a50040	00550040
Windows Phone	IPM.SMStext	Hi! Check out our timesaving tips and videos for getting the most from your new phone.windowsphone.com/helloThis is a free message.		94670216746762240	130775345507202030
Windows Phone	IPM.SMStext	Hi there! To find out about the new, fun stuff that's in your Windows Phone 8.1 update, go to windowsphone.com/newThis is a free message.		130775459573759430	130775459573698941
+852[redacted]	IPM.SMStext	Your WhatsApp code is 525-096 but you can simply tap on this link to verify your device:v.whatsapp.com/525096		130775502652183734	130775502910000000
info	IPM.Note	Registration at GsmServer online store		130775537944503336	
info	IPM.Note	Order #1450062106 at GsmServer.com		130775537944825625	
info	IPM.Note	Password Recovery Confirmation		130775537945117696	
info	IPM.Note	Invoice #: 1450062106 Payment received		130775537945419884	
jkongyc	IPM.Note	Your Download: JTAG for Mobile Forensics training	Fwd:	130775537945711954	
info	IPM.Note	Invoice #: 1450062106 Ordered items sent		130775537946004010	
paypal	IPM.Note	Kong Yu Cho - 請查閱你的 5 月份帳戶月結單。		130775537946326338	
noreply	IPM.Note	Riffbox Server user registration confirmation		130775537946628504	
jkongyc	IPM.Note	回覆： 轉寄： Fwd: Notice of Intent to Submit a Thesis		130775549867985286	
jkongyc	IPM.Note	Your Forensics CE License Key	Fwd:	130775549868639885	
member_services	IPM.Note	Getting started with your mail account		130775576468913521	
jkongyc	IPM.Note	私人訊息：michaelmmk Nikon AF-S VR Zoom-Nikkor 70-300mm f/4.5-5.6G IF-ED	Fwd:	130775576470220121	
jkongyc	IPM.Note	Reminder: OBL - the assessment report and samples of UG courses (2nd sem, 2014-15)	Fwd:	130775576470763975	
mr.joekong	IPM.Note	Reminder: OBL - the assessment report and samples of UG courses (2nd sem, 2014-15)	RE:	130775576942921152	
+852[redacted]	IPM.SMStext	Callno: 004.1675 W58 ; Title: Windows Phone 8 development internals / Andrew Whitechapel, Sean McKen...		130776369225527160	130776369190000000
	IPM.SMStext	Htcia 2015 will be held Dec 5		130776374601392713	130776374601352436
+852[redacted]	IPM.SMStext	□		130776374851147070	130776374840000000

Figure 4-16. Two timestamps found in the store.vol

They were in Windows filetime format containing a value of 64-bit and decoded as illustrated in Figure 4-17. The value is in fact the number of 100-nanosecond intervals (UTC) commencing on January 1, 1601. In the two columns circled in red in Figure 4-16, the first column was the server receipt time while the second column was the device receipt time after conversion.



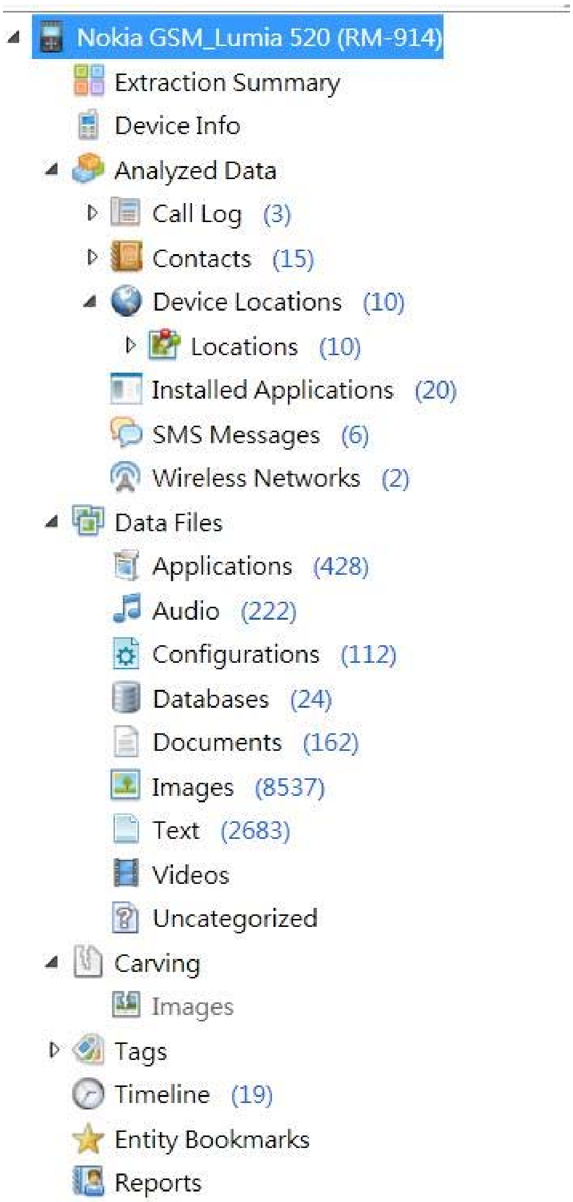
SiliSoftware » Tools » Date format converter

Input date: 130776369225527160
Input format: filetime

Text Date: *Date in human-readable text*	Monday, June 1, 2015 12:55:23pm

Figure 4-17. Conversion of SMS timestamp from Windows filetime

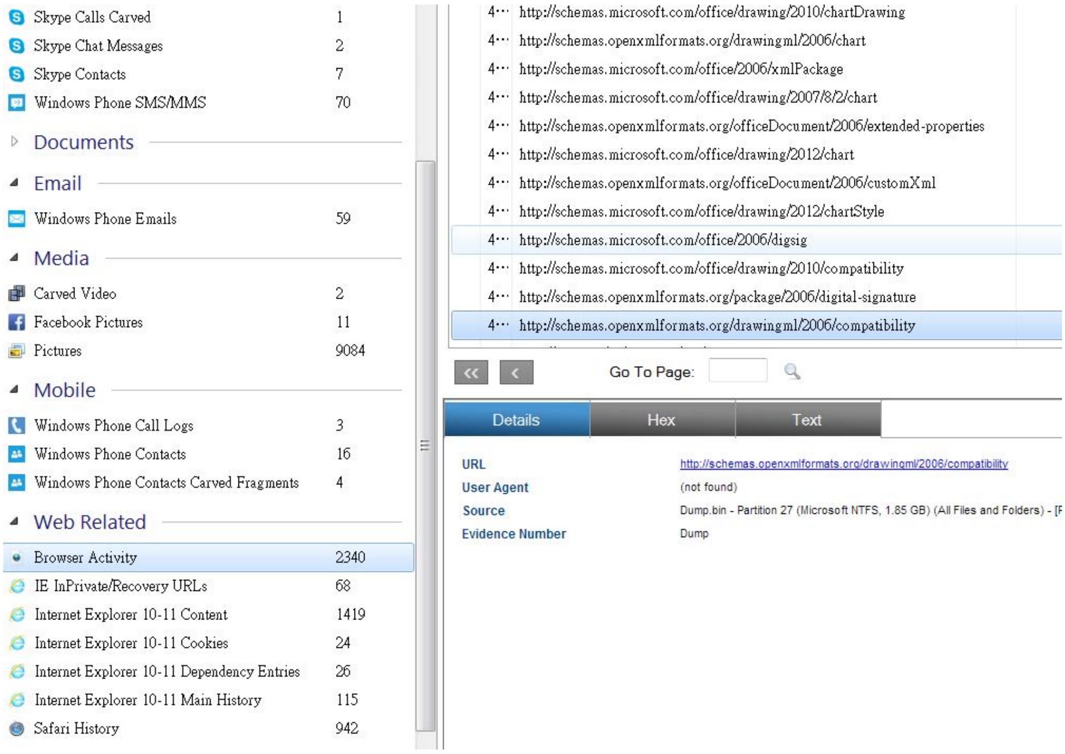
Another important issue was the missing of online activities. As shown in Figure 4-18, the analysis report created by Cellebrite Physical Analyzer did not include any findings on Internet artifacts.



- Nokia GSM_Lumia 520 (RM-914)
  - Extraction Summary
  - Device Info
  - Analyzed Data
    - Call Log (3)
    - Contacts (15)
    - Device Locations (10)
      - Locations (10)
    - Installed Applications (20)
    - SMS Messages (6)
    - Wireless Networks (2)
  - Data Files
    - Applications (428)
    - Audio (222)
    - Configurations (112)
    - Databases (24)
    - Documents (162)
    - Images (8537)
    - Text (2683)
    - Videos
    - Uncategorized
  - Carving
    - Images
  - Tags
  - Timeline (19)
  - Entity Bookmarks
  - Reports

Figure 4-18. No Internet artifacts presented in Cellebrite report

There are similarities between the Windows Phone as well as the Windows desktop OS and they include the file system structure, directory layout and the common presence of registry and system files such as pagefile.sys. Therefore, it is possible to perform same type of computer forensic examination to parse artifacts and user data files from the target phone. Accordingly a number of analysis tools, including open source and commercial tools have been evaluated with regard to their ability to carve artifacts from the pagefile.sys. It was found that the Internet Evidence Finder from Magnet Forensics could identify a substantial amount of evidence on visits to Web page, social network profile and open chat session [66]. Pictures or video footages could also be viewed and presented logically in the forensic report as displayed in Figure 4-19.



Skype Calls Carved	1
Skype Chat Messages	2
Skype Contacts	7
Windows Phone SMS/MMS	70
Documents	
Email	
Windows Phone Emails	59
Media	
Carved Video	2
Facebook Pictures	11
Pictures	9084
Mobile	
Windows Phone Call Logs	3
Windows Phone Contacts	16
Windows Phone Contacts Carved Fragments	4
Web Related	
Browser Activity	2340
IE InPrivate/Recovery URLs	68
Internet Explorer 10-11 Content	1419
Internet Explorer 10-11 Cookies	24
Internet Explorer 10-11 Dependency Entries	26
Internet Explorer 10-11 Main History	115
Safari History	942

```
4··· http://schemas.microsoft.com/office/drawing/2010/chartDrawing
4··· http://schemas.openxmlformats.org/drawingml/2006/chart
4··· http://schemas.microsoft.com/office/2006/xmlPackage
4··· http://schemas.microsoft.com/office/drawing/2007/8/2/chart
4··· http://schemas.openxmlformats.org/officeDocument/2006/extended-properties
4··· http://schemas.microsoft.com/office/drawing/2012/chart
4··· http://schemas.openxmlformats.org/officeDocument/2006/customXml
4··· http://schemas.microsoft.com/office/drawing/2012/chartStyle
4··· http://schemas.microsoft.com/office/2006/digsig
4··· http://schemas.microsoft.com/office/drawing/2010/compatibility
4··· http://schemas.openxmlformats.org/package/2006/digital-signature
4··· http://schemas.openxmlformats.org/drawingml/2006/compatibility
```

URL	http://schemas.openxmlformats.org/drawingml/2006/compatibility
User Agent	(not found)
Source	Dump.bin - Partition 27 (Microsoft NTFS, 1.85 GB) (All Files and Folders) - [F
Evidence Number	Dump

Figure 4-19. Internet artifacts in the report generated by Internet Evidence Finder

4.4.4 Conclusion

The Cellebrite UFED Touch is claimed to be the first and only forensic tool to conduct physical extraction on Nokia Lumia 520 running Windows Phone 8.1. Other commercial forensic software, e.g. Internet Evidence Finder (IEF), Belkasoft Evidence Centre have to rely on the binary dump acquired by Cellebrite and use the binary file for further decoding and analysis. It is quite unusual in the commercial market for a manufacturer to build a commercial product that relies on the output of other market opponents. This may suggest that Microsoft has been successful in implementing secure boot in the latest Windows Phone models.

Again it is important for forensic examiners to understand the characteristics of the tools that they are using. Open source tools which can perform an in-depth analysis play an important role in uncovering information relating to the timestamp of SMS and this may apply to other applications or dataset. It is always a doctrine to a forensic examiner that he should use multiple tools to validate his findings. Therefore, the result generated by standard forensic tools can be validated by another tool which is designed specifically to deal with certain type of data structure in order to ensure there is no missing information before consolidating all the results to give the best possible visual findings.

For instance, when IEF is used to validate the same binary dump acquired by the Cellebrite UFED Touch, it can yield the Internet browsing history which has not been provided in the Cellebrite report. The difference in result is because the Cellebrite Physical Analyzer will not carve any internet artifacts from the pagefile.sys in the system partition.

When examining a computer running Windows, a forensic examiner will always focus on the pagefile.sys as it can provide substantial information such as passwords, encrypted keys, typed commands, webpages, social networking chat session, shared and executable files, current running processes and terminated processes, open ports and active connections. However, as the Windows Phone uses ARM architecture, its format and storage of page files will be different from its desktop counterpart, Windows 8 despite they both run on the NT Kernel. Therefore, the pagefile.sys on Windows Phone has to be handled by dedicated data carving tools, which are mobile forensics tools mentioned in previous paragraphs, in order to search for known data types.

The data carvers recover files depending on the presence of file content signature [67]. Although the carved result is compromised due to the absence of certain metadata, the contents recovered can still play a role in the collaboration of digital evidence or may possibly give references to forensic investigations.

4.5 Overall observations

The selection criteria of the extraction tools used in the experiments is based on the principle that physical extraction can yield more information and is therefore preferable. Besides, data integrity to remain intact during the extraction process is of paramount importance. As regard to the process on data decoding or analyzing, open source and commercial tools are deployed based on their functionalities and popularity in analyzing specific data types. In addition, the easy-to-use operational procedures and availability of comprehensive manuals of the commercial tool will bring credits to the selection criteria. For the sake of this thesis, only those commercial tools providing free trial versions are being evaluated.

Table 4-1 below shows a summary of the evaluation result. The experiment of iTunes Backup is only a proof of concept and in real situation, instead of examining the phone, a crime investigator should seize the computer which has already synchronized with the target phone in order to analyze the backup files therein.

	Android	iOS	Windows Phone
Extracting data	SP Flash Tools physical extraction of full memory by custom boot loader and data integrity maintained.	iTunes Backup logical extraction of partial user data (e.g. email messages is excluded) and system files modified during synchronization.	RIFF Box Physical extraction of full memory by JTAG and data integrity maintained.
Decoding Application Data	Whatsapp Viewer backup files of Whatsapp are located and deleted messages are found.	iPhone Backup Browser file names and paths are correctly displayed; data are	Internet Evidence Finder Examine the pagefile.sys and find substantial amount

		organized under different application package mbdbdump.exe; convert the manifest.mbdb into text file and additional information are found.	information on visits to Web pages, social network profile and open chat session.
Analyzing Log and timestamp	SQLite Forensic Browser examine the msgstore.db and find extra timestamp recording Whatsapp chats.	mbdbdump.exe convert the manifest.mbdb into text file and user activities logs are found.	ESE Database Browser examine the store.vol and find extra timestamp recording SMS communication.

Table 4-1 Evaluation of different data decoding methods

Considering the selection criteria as described in the preceding paragraph and the experiments that have been conducted by using current extraction methodologies, it is recommended that in normal circumstances, physical extraction using custom boot loader will be the best approach for data extraction on Android smartphone while logical extraction using iTunes backup is more practical to extract data from non-Jailbroken iOS device. Regarding Windows Phone 8, where Cellebrite UFED, the only commercial tool which supports physical extraction of data by using boot loader, is not available, JTAG extraction should then be applied.

So far the information revealed in the above experiment is crucial for forensic examiners. It is not disputable that digital evidence is fragile in nature and can easily be tampered with during the course of examination. As such in conducting an investigation of crime, the investigator will need to have collaboration to support their findings. In most cases, an examiner will tackle the investigation in post-crime scenarios and digital evidence is usually the end result to prove the offender’s activities. It is therefore necessary to have other pieces of evidence showing the course of action to confirm or provide additional information to verify the examiner’s findings.

In the iPhone case, examination of the Whatsapp log files will help confirming the conversation between two parties did take place by recording the time that they invoked the Whatsapp application as well as the attachment of media files, if there is any. Meanwhile, the domain information contained in the manifest.mbdb may be useful in determining the source of information or parent application that generates the child data. For instance, if child pornography does exist in the mobile phone, the examiner needs to ascertain if the photos to be investigated are taken by the phone’s camera or they are in fact data files downloadable from the Internet. It is deemed necessary for examiners to have in-depth analysis on the phone application and associated data, such as artifacts, log files or any hidden information and background process so that those information can be reviewed to strength the proof of events when a crime occurs.

Most of the commercial tools are catered for the extraction and recovery of all files from the user partition so that the examiner can have easy access to conduct further analysis. They have consolidated a large number of user requirements to develop automated functions of their tools. The objective is to seamlessly assist the examiner to carve and analyze the data according to the defined workflow. However, the analytical functions created by different vendors are dependent on how they understand or interpret their clients' need and can be affected by the market value of these products. Besides, the timeline of an investigation does not normally align with the software development priorities and production target of commercial forensic tool vendors. In the Android and Windows Phone cases, some data or information was left behind without processing. It is unfortunate that their operational manuals do not include any reminder to the examiner that these data will not be processed. Besides, forensic examiners may work as if they are in black boxes since manufacturers will be reluctant to release the source code of these commercial tools. .

In introducing non-commercial alternatives and methodologies to review the test results, it provides opportunities to compare and determine the efficiency of the commercial tools. Any differences and the causes of these differences should be investigated so that they can be explained as part of the evidence. Another compelling reason to develop alternatives is to supplement the results whenever the commercial tools do not or only partially support data extraction and parsing from the targeted smartphone. When a suspect is alleged to have in possession of some child pornography in his mobile device, the investigator will like to track down how these materials have come into his possession. The Internet browsing history may give him a clue to ascertain whether a syndicate has been involved in treating this as a business trading. This searching process is somehow similar to investigating a physical crime like murder. The police will look for the lethal weapon, for example, a knife and will then trace for the source of this weapon. The investigation process is to strength the evidence proving that this is a premeditated murder so that the suspect cannot plead to manslaughter in court. Besides, it may also assist in identifying any co-accused involved in the crime case.

Forensic examiners who work with law enforcement agencies should be reminded that the existing commercial tools can only serve as a mean to expedite the examination process. An analysis on mobile forensics cannot solely rely on the results or findings obtained by a single forensic tool. Despite this, in real life situations, given the heavy caseload and emerging amount of mobile phone seizures, it is difficult for crime investigators or forensic examiners to carry out in-depth analysis on all applications to verify the forensic results in each investigation case. As mentioned before, the apprehension of offenders and bring them to court should be made in a timely manner. It is prudent to focus the analysis on a particular area of digital evidence for analysis so that the best forensic tool can be shortlisted to complete the task effectively and accurately. In order to achieve this purpose, the forensic examiner should:

- communicate with case officers and know exactly the scope of data or evidence that forms the subject of investigation; and
- acquaint himself with the latest forensic tools, either from commercial or open-source origin and also their functionalities.

Chapter 5 Conclusion

5.1 Discussion

Crime investigators and forensic examiners have put in longstanding efforts on the effective management of digital evidence derived from smartphones due to the following reasons:

- Smartphones integrate proprietary interface, storage media and hardware. Different approaches and toolkits may only extract limited data, or could have analyzed artifacts inaccurately or imperfectly.
- The variety of OSs embedded in the devices indicate that different manufacturers as well as mobile phone models can contain the same type of information using different file formats.
- Law enforcement agencies have difficulty to stay current with new technologies due to the short product cycles of mobile devices, which are a consequence of the unceasingly launching of new phone models and the enhanced security features of the devices.
- The forensic tool might have been evaluated and found to be reliable in the past but its reliability has become uncertain when coping with the

latest mobile technology advancement (even if a forensic tool claims to have supported a particular model, it does not imply that the tool can further support the updated firmware of that model).

- Extraction results from mainstream commercial tools may be accepted on basis of the vendor’s reputation, which in turn is frequently referred to the recognition by name.
- The absence of prominent mobile forensic tools and the decreasing budgets for acquiring suitable tools have caused ongoing problems to mobile forensics as there is no single tool which can possibly be recommended by the forensic community.

The practical consequence of the conclusions has prompted to the importance of developing a diverse range of forensic tools that can be utilized by examiners to carry out analysis of information derived from mobile devices. However, cross comparisons of artifacts in multiple results can also arouse concerns on the reliability of each of the tools used. Unfortunately, research conducted on product descriptions of existing tools on the developer’s sites have failed to give a clear understanding of the functions and features of these tools. Even when it claims to provide support for more than seven thousand profiles, in practice it can only support three thousand phone models after taking into account different extraction methods applied on the same phone, for example, physical vs file system or advanced logical vs logical [68].

5.2 Recommendation

Forensic examiners should aware that a complete disclosure of all evidential information stored in the mobile device is vital to an investigation. The experiments conducted for this thesis are not intended to identify all the issues arising from smartphone forensics nor provide all possible solutions to cope with those issues as new mobile devices and applications as well as forensic tools are evolved and launched rapidly. The objective of this thesis is to serve as a reminder to forensic practitioners that they should not rely on one or two tools or even simply wait for the launching of new releases of forensic tools to carry out a thorough examination of the target device.

Besides, examiners should also keep in mind that the ultimate goal to deploy forensic examination on smartphones is to recover digital information which can be adduced as reliable evidence in court proceedings. Under the rule of laws the prime concern will be the integrity of data to remain intact during the course of forensic examination. Based on this principle, physical extractions using a custom bootloader or a JTAG acquisition as discussed in the experiment seem to be the most preferable way. Furthermore, it is revealed that physical extractions can yield more information or even a full memory extraction through bypassing the operating system restriction and the security settings. This full memory extraction will provide better opportunities for forensic examiners to carve deleted data which may be of interest to an investigation. As regard to the phases for data recovery and analysis, giving the fast pace in the development of native and third party smartphone applications, the methodology of deploying both open source and commercial forensic tools will complement each other based on their specific functionalities in analyzing different types of data. In reality, due to the time constraint on conducting a crime investigation and the relatively short time interval in understanding the target device's architecture, commercial forensic tools that introduce easy-to-use operational procedures and supply with comprehensive manuals will undoubtedly get credits in the selection criteria.

The research work carried out in this thesis demonstrates that in the event of applying a diverse approach to information recovery, it can lead to a more complete result being produced.

The major problem which forensic examiners have to deal with is to ensure the reliability of digital evidence presented in court. The legal impacts are:

- whether the evidence presented in court represents the actual events that took place when the crime occurs; and

- whether it is possible to rely absolutely on that evidence.

It is therefore recommended that the mobile device should be investigated with deeper insights instead of simply examining the OS level with some automated tools. Examiners should look for a better understanding of the underlying data structure that the tools are parsing and they can determine the validity of the output that is produced. Besides, it is always advisable to obtain and examine additional hardware that a mobile device can be synchronized with, such as a laptop or workstation.

In reality, there is growing pressure on forensic examiners to extract, filter, analyze and share insights speedily so as to avoid building up backlogs of the device under investigation. The ability to utilize the right tools for examination and visualize key connections quickly will help unlocking the intelligence adduced from digital data to guide criminal investigative teams, speed up investigations and produce evidence for critical decision making. As such, further research work is necessary to assist forensics examiners in making comparison of test results obtained from different toolsets and methods.

Technological evolution is beyond imagination. Perhaps, if forensic examiners can develop an understanding of the fundamentals, they may be able to build up a foundation upon which new issues can be addressed effectively rather than responding to these challenges by making use of the “traditional” methodology, which may be unviable. That is to say, instead of attempting to research for an unrealistic automated forensic tool to cope with all smartphone models in the market, the fundamental need for designing an independent platform with dedicated carvers to support the recovery of artifacts from the memory dump is deemed necessary to help verifying the analyzed results produced by standard forensic toolkits.

5.3 Future landscape

Digital forensics is a reactive field. Thus, analysis on mobile devices is largely dependent on the trends in mobile phone industry. Given the explosion of the permissive policy, Bring-Your-Own-Device (BYOD) and coupled with the rapid acceptance of using smartphones in the commercial sector, mobile forensics has nonetheless changed its landscape on the detection, collection and use of digital evidence. It is not surprising that nowadays smartphones have been used universally to denote a mobile device. Although the proliferation of high-end smartphones in both consumer and business sectors has played an increasingly important role in forensic investigations, it is envisaged that low-end phones will continue to exist in significant numbers because they are inexpensive and easy to dispose. Prepaid SIM cards and low-priced mobile phones are welcome by criminals because it is difficult to track down the ownership of their users even if the device has been used for a prolonged period of time. Besides, these devices can be physically destroyed to prevent data recovery.

Mobile device forensics is not an easy specialty. The increased storage capacity of mobile phones requires more in-depth analysis to be conducted on per-device basis and examiners are required to have a better understanding of the phone's in-built architecture. It is also anticipated that forensic analysis on mobile devices will involve the reverse-engineering of third party applications to explore the data contained therein. Forensic examiners and analysts need to constantly conducting researches to get ahead of the new technology.

It is anticipated that commercial forensics applications will provide communication gateway to extract, preserve and analyze cloud-based account data by utilizing login information or user credentials extracted from the mobile device [69]. Mobile phones connected to computer workstations are acting like a satellite device. Examination of these connected systems can help to gaining a better picture on the incident and may also provide additional investigative leads.

The future landscapes of the three dominant smartphone systems are:

- The future of iOS forensics lies with over-the-air acquisition as the majority of users incline to configure their devices to maintain cloud backups. Data can be obtained either from iCloud or when requested by Apple. Except in cases when device passcodes or backup passwords are unavailable, logical acquisition can be conducted via offline iTunes backups. As for physical acquisition, it is no longer applicable for iOS forensics.

- Android has gradually become a secure platform. Not surprisingly more and more devices feature the whole-disk encryption out of the box. Indeed Android 5 could have brought about an obstacle to physical acquisition as many forensic tools do not support the physical acquisition of data from these devices unless the passcode is known. Physical acquisition remains the viable extraction option for use on current phone models subject to the launching of Android 6 devices in near future which will be encrypted out of the box [70].

- Since 2012, Microsoft has started to unify the Windows platform across device classes. Windows Phone 8 built the platform upon the NT kernel that shared much of the same architecture with its PC counterpart, Windows 8. In July 2014 Nadella pointed out that Microsoft was planning to "streamline the next version of Windows from three OSs into one single converged OS for screens of all sizes" [71]. In delivering the Build keynote in 2015 [72], Microsoft announced the porting of Android and iOS software to run on Windows 10 smartphone as the OS features the runtime environment "Astoria" for Android software and the Windows Bridge for iOS [73].

All these will pose further challenges in the realm of mobile phone forensics. Apart from constantly conducting researches to get ahead of the new technology, forensic examiners should have creativity and be intelligent along with the enthusiasm to stray beyond traditional digital forensics practices.

Bibliography

[1] ComScore, The U.S. Mobile App Report 2014, https://www.comscore.com/Insights/Presentations-and-Whitepapers/2014/The-US-Mobile-App-Report

[2] Chan C, published at Gizmodo on 10/02/2012, This Lost Speech from 1983 Will Make You Think That Steve Jobs Was from the Future http://gizmodo.com/5948434/this-lost-speech-from-1983-will-make-you-think-that-steve-jobs-was-from-the-future

[3] International Data Corporation, Smartphone OS Market Share, 2015 Q2 http://www.idc.com/prodserv/smartphone-os-market-share.jsp

[4] GSM Mobile World, Market Share, May 2015, http://gsm-mobileworld99.blogspot.hk/2015/05/market-share.html

[5] OpenSignal Inc., Android Fragmentation Visualized, August 2014, https://opensignal.com/reports/2014/android-fragmentation/

[6] Michael E, Procedure for gaining physical access to certain Android phone running Kit Kat, the International Association of Computer Investigative Specialists, 2015

[7] Fabio M, iOS and Android File System Overview, in Advanced Smartphone Forensics, eForensics Magazine, 2015, https://eforensicsmag.com/course/advanced-smartphone-forensics-w14/

[8] Windows Phone architecture overview, 10/07/2015, https://sysdev.microsoft.com/en-us/Hardware/oem/docs/Getting_Started/Windows_Phone_architecture_overview

[9] Kantar WorldPanel, Smartphone OS sales market share http://www.kantarworldpanel.com/global/smartphone-os-market-share/

[10] http://forums.windowscentral.com/windows-phone-8/269441-how-i-successfully-encrypted-windows-phone.html

[11] Ayers R, Sam Brothers, Wayne Jansen, Mobile Device Forensics Revision 1, National Institute of Standards and Technology, May 2014, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-101r1.pdf

[12] Statista , Number of apps available in leading app stores as of July 2015, http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

[13] Casadei F, Savoldi A, Gubian P, Forensics and SIM cards: an Overview, International Journal of Digital Evidence, Fall 2006, Vol.5, Issue 1,

https://www.utica.edu/academic/institutes/ecii/publications/articles/EFE3EDD5-0AD1-6086-28804D3C49D798A0.pdf

[14] Willassen S, Forensic Analysis of Mobile Phone Internal Memory, in Advances in Digital Forensics, Vol 194, The International Federation for Information Processing, 2005, Page 191-204

[15] Breeuwsma IM, Forensic imaging of embedded systems using JTAG (boundaryscan), Digital Investigation: The International Journal of Digital Forensics & Incident Response, Vol 3, Issue 1, March 2006, Page 32 - 42

[16] Klaver C, Windows Mobile advanced forensics, Digital Investigation 6 (2010) Page 147-167, http://www.sciencedirect.com/science/article/pii/S1742287610000095

[17] Hengeveld W, xda tools, 2009, www.xs4all.nl/witsme/projects/xda/tools.html

[18] JTAG in Mobile Forensics, NowSecure Inc, 2014, https://www.nowsecure.com/resources/jtag-forensics-training/

[19] Dellutri F, Ottaviani V, Me G, MIAT-WM5: Forensic acquisition for windows mobile pocketpc, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.187.2171&rep=rep1&type=pdf

[20] Jahankhani H, Criminal investigation and forensic tools for smartphones, International Journal of Electronic Security and Digital Forensics 2009, Vol 2 No.4 page 387-406

[21] Good Practice Guide for Digital Evidence Version 5, updated on March 2012, Association of Chief Police Officers http://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf

[22] Girault E, Volatilitux : Physical memory analysis of Linux systems, December 2010, http://www.segmentationfault.fr/projets/volatilitux-physical-memory-analysis-linux-systems/

[23] Kollar I, Master Thesis, Charles University in Prague, Forensic RAM dump image analyser, 2010, http://hysteria.sk/~niekt0/foriana/doc/foriana.pdf

[24] Zalewski M, Memfetch, 2002, http://lcamtuf.coredump.cx/soft/memfetch.tgz

[25] Sylve JT, Android Memory Capture and Applications for Security and Privacy, University of New Orleans Theses and Dissertations, 2011, http://scholarworks.uno.edu/cgi/viewcontent.cgi?article=2348&context=td

[26] Valenzuela I, Acquiring volatile memory from Android based devices with LiME Forensics Part I, April 2012, http://blog.opensecurityresearch.com/2012/04/acquiring-volatile-memory-fro

m-android.html

[27] Lessard J, Kessler G.C., Android Forensics: Simplifying Cell Phone Examinations, Small Scale Digital Device Forensics Journal 4(1), 2010, http://www.garykessler.net/library/SSDDFJ_V4_1_Lessard_Kessler.pdf

[28] Vidas T, Zhang C, Christin N, Toward a general collection methodology for Android devices, Digital Investigation 8 (2011) S14 - S24, http://www.dfrws.org/2011/proceedings/07-339.pdf

[29] Vijayan V, Android Forensic Capability and Evaluation of Extraction Tools, Edinburgh Napier University Thesis, 2012, http://www.academia.edu/1632597/Android_Forensic_Capability_and_Evaluation_of_Extraction_Tools

[30] Thing V.L.L., Ng K.Y., Chang E.C., Live memory forensics of mobile phones, in Digital Investigation, Volume 7, 2010, Pages 74-82

[31] Hoog A, Chapter 7 - Android Application and Forensic Analysis (YAFFS2 forensic analysis) in Android forensics : investigation, analysis and mobile security for Google Android, 2011

[32] Jin Y, Timeline analysis for Android-based systems, M.S. thesis, Technical University of Denmark, 2013, http://www2.imm.dtu.dk/pubdb/views/edoc_download.php/6582/pdf/imm6582.pdf

[33] Grover J, Android forensics: Automated data collection and reporting from a mobile device, Digital Investigation, Volume 10, Supplement, August 2013, Pages S12-S20, http://www.sciencedirect.com/science/article/pii/S1742287613000480

[34] Mobile Phone Examiner Plus, AccessData Corporation, http://accessdata.com/solutions/digital-forensics/mpe

[35] NowSecure Forensic, NowSecure Inc., https://www.nowsecure.com/forensics/

[36] Cellebrite UFED Touch & Physical Analyzer, Cellebrite Limited, http://www.cellebrite.com/Pages/Forensics-Device-Specific-Capabilities

[37] XRY Office, Micro Systemation AB, https://www.msab.com/products/office/

[38] Zdziarski JA, iPhone Forensics: Recovering Evidence, Personal Data, and Corporate Assets, O’Reilly Media, Inc., ISBN 978-0596153892, 2008

[39] Hoog A, Gaffaney K, iPhone Forensics, 2009 http://www.mandarino70.it/Documents/iPhone-Forensics-2009.pdf

[40] Bader M, Baggili I, iPhone 3GS Forensics: Logical Analysis using Apple iTunes Backup Utility, Small Scale Digital Device Forensics Journal Vol.4, No.1, September 2010,

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.185.4439&rep=rep1&type=pdf

[41] Husain MI, Baggili I, Sridhar R. A Simple Cost-Effective Framework for iPhone Forensic Analysis, in Digital Forensics and Cyber Crime, ICDF2C 2010 page 27–37

[42] Jung J, Jeong C, Byun K, Lee S, Sensitive privacy Data Acquisition in the iPhone for Digital Forensic Analysis, in Secure and Trust Computing, Data Management and Applications, 8th FIRA International Conference, 2011, page 172–186

[43] Tso YC, Wang SJ, Huang CT, Wang WJ, iPhone social networking for evidence investigations using iTunes forensics, in Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication, Article 62, 2012

[44] Arrifin A, D'Oorazio C, Choo KKR, Slay J, iOS Forensics: How Can We Recover Deleted Image Files with Timestamp in a Forensically Sound Manner? in Availability, Reliability and Security, Eighth International Conference, 2013

[45] Casey, E., Bann, M., & Doyle, J. (2010). Introduction to Windows mobile forensics. Digital investigation, 6(3), 136-146

[46] Klaver C, Kaart M, Baar RB van, Forensic access to Windows Mobile pim.vol and other Embedded Database (EDB) volumes, Digital Investigation 9 (2013) Page 170-192, http://www.sciencedirect.com/science/article/pii/S1742287612000874

[47] Rehault F, Windows mobile advanced forensics: An alternative to existing tools, Digital Investigation 7 (2010) Page 38-47, http://www.sciencedirect.com/science/article/pii/S1742287610000551

[48] Canlar ES, Conti M, Crispo B, Pietro RD, Windows Mobile LiveSD Forensics, Journal of Network and Computer Applications 36 (2013) Page 677-684, http://www.sciencedirect.com/science/article/pii/S1084804512002718

[49] Schaefer T, Hofken H, Schuba M, Windows Phone 7 from a Digital Forensics' Perspective in Digital Forensics and Cyber Crime 2012, Pages 62-76

[50] Leong A, Murphy C, Gaffncy M, Punja SG, Gibb JA, McGarry B, Windows Phone 8 Forensic Artifacts, SANS Institute InfoSec Reading Room, 2015, https://www.sans.org/reading-room/whitepapers/forensics/windows-phone-8-forensic-artifacts-35787

[51] Tindall D, Tamma R, Learning Android Forensics, Packt Publishing, 2015

[52] Proffitt T, Forensic Analysis on iOS Devices, SANS Institute InfoSec Reading Room, Nov 2012, http://www.sans.org/reading-room/whitepapers/forensics/forensic-analysis-ios-devices-34092

[53] B´edrune JB, Sigwald J, iPhone data protection in depth, Sogeti ESEC Lab, Hack in The Box Security Conference 2011, http://esec-lab.sogeti.com/static/publications/11-hitbamsterdam-iphonedataprotection.pdf

[54] Apple Privacy Policy, http://www.apple.com/privacy/manage-your-privacy/

[55] Apple developer website to show that iOS 9 is now running on 61 percent of iPhones and iPads, October 2015, https://developer.apple.com/support/app-store/

[56] Windows Phone 8 Security Guide, September 2013, http://download.microsoft.com/download/9/4/2/942B0F2C-3962-4B4D-B71A-CCC63649F8F3/Windows%20Phone%208%20Security%20Overview.pdf

[57] iPhone Backup Browser, https://code.google.com/p/iphonebackupbrowser/

[58] Elcomsoft Phone Breaker, Elcomsoft Proactive Software, https://www.elcomsoft.com/eppb.html

[59] SP Flash Tool + MediaTek MT65XX Drivers Download and Installation Guide including Bricked Devices, July 2014, http://laurentiumihet.ro/sp-flash-tool-mediatek-mt65xx-drivers-download-and-installation-guide-including-bricked-devices/

[60] MTK Droid Tools, How to create Scatter File for MTK Devices, http://androidmtk.com/create-scatter-file-for-mtk-devices

[61] FTK Imager User Manual, AccessData, https://ad-pdf.s3.amazonaws.com/Imager%203_1_4_UG.pdf

[62] Husen4u, Whatsapp Viewer for PC, April 2014, http://forum.xda-developers.com/showthread.php?t=2719741

[63] Sanderson P, Forensic Browser for SQLite, Sanderson Forensics, http://sandersonforensics.com/forum/content.php?198-Forensic-Browser-for-SQLite

[64] Zatyko K, Bay J, The Digital Forensics Cyber Exchange Principle in Forensic Magazine, Dec 2011, http://www.forensicmag.com/articles/2011/12/digital-forensics-cyber-exchange-principle

[65] RIFF Box JTAG, www.riffbox.org

[66] Internet Evidence Finder, Magnet Forensics, https://www.magnetforensics.com/computer-forensics/internet-evidence-find

er-adds-mac-os-x-file-system-support-new-timeline-feature-in-latest-upgrade-to-forensic-software/

[67] Grispos G, Storer T, Glisson WB, A comparison of forensic evidence recovery techniques for a windows mobile smart phone, Digital Investigation 8, Page 23-36, 2011, http://www.sciencedirect.com/science/article/pii/S1742287611000417

[68] Oxygen Software, Mobile Phone Forensic Challenge, Forensic Focus, May 2012, http://articles.forensicfocus.com/2012/05/17/mobile-phone-forensic-challenges/

[69] UFED Cloud Analyzer by Cellebrite Limited, Delivering Cloud Data Access and Insights to Accelerate Investigations, 2015 http://www.cellebrite.com/Mobile-Forensics/Applications/ufed-cloud-analyzer

[70] Cunningham A, Android 6.0 re-implements mandatory storage encryption for new devices, Oct 2015, http://arstechnica.com/gadgets/2015/10/android-6-0-re-implements-mandatory-device-encryption-for-new-devices/

[71] Kelion L, BBC News, Windows development set to be 'unified' by Microsoft, July 2014, http://www.bbc.com/news/technology-28440288

[72] Warren T, The Verge, Windows 10 can run reworked Android and iOS apps, April 2015, http://www.theverge.com/2015/4/29/8511439/microsoft-windows-10-android-ios-apps-bridges

[73] Denning A, Windows Bridge for iOS, August 2015, https://blogs.windows.com/buildingapps/2015/08/06/windows-bridge-for-ios-lets-open-this-up/