

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By William G. Harshbarger Jr.

Entitled ANDROID TABLET FORENSIC LOGICAL IMAGE TOOL TESTING

For the degree of Master of Science



Is approved by the final examining committee:

Marcus K. Rogers

Chair

J. Eric Dietz

Philip T. Rawles

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Marcus K. Rogers

Approved by: James L. Mohler

Head of the Graduate Program

07/18/2012

Date

**PURDUE UNIVERSITY
GRADUATE SCHOOL**

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

ANDROID TABLET FORENSIC LOGICAL IMAGE TOOL TESTING

For the degree of Master of Science



I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22*, September 6, 1991, *Policy on Integrity in Research*.*

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

William G. Harshbarger Jr.

Printed Name and Signature of Candidate

07/06/2012

Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

ANDROID TABLET FORENSIC LOGICAL IMAGE TOOL TESTING

A Thesis

Submitted to the Faculty

of

Purdue University

by

William G. Harshbarger Jr.

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science

August 2012

Purdue University

West Lafayette, Indiana

UMI Number: 1529682

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1529682

Published by ProQuest LLC (2012). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

For Mom and Dad

ACKNOWLEDGEMENTS

No one can truly achieve without a network of great people in support. First, I acknowledge my advisor, Dr. Marcus Rogers, a great educator who kept me on track and who has built a great program at Purdue. I also acknowledge my committee members Philip Rawles and Dr. Eric Dietz. Without their guidance this research would not have taken place. Greg Hedrick, my manager while working at Purdue who made it possible for me to attend graduate school while maintaining a full time job. My coworkers at ITSP Brad Graves, Doug Couch, Nathan Heck, and the rest of the team who helped cover for me when I had daytime classes or important deadlines to meet. Also, fellow Purdue grad students Taylor Owings, Eric Katz, Matt Levendoski and Rob Winkworth who gave me valuable help and advice on completing this research. I'd also like to acknowledge the CIT secretarial staff that has provided much help when needed. Finally, I thank Dr. Nicole Becker for supporting me, being a formatting whiz, master editor, and introducing me to PhD comics.

TABLE OF CONTENTS

	Page
LIST OF TABLES	vii
LIST OF FIGURES	viii
ABSTRACT	x
CHAPTER 1 INTRODUCTION.....	1
1.1 Background.....	1
1.2 Scope.....	3
1.3 Significance.....	4
1.4 Research Question	4
1.5 Assumptions.....	5
1.6 Limitations	5
1.7 Delimitations.....	6
1.8 Definition of Key Terms.....	6
1.9 Summary	7
CHAPTER 2 REVIEW OF RELEVANT LITERATURE	9
2.1 Background.....	9
2.2 A Brief History of Android to Present.....	9
2.3 Android Operating System Architecture	11
2.4 Android Tablet Nonvolatile Storage.....	13
2.4.1 NAND Flash Memory.....	13
2.4.2 File Systems	15
2.5 Digital Forensic Science and Tool Testing.....	17
2.6 The Nature of Errors in Science and Forensic Practice	19
2.7 A Survey of Android Device Acquisition Technologies	21
2.7.1 Hardware Based Data Acquisition.....	21
2.7.2 Methods That Require Rooting.....	22
2.7.3 Secure Digital (SD) Card Swap	23
2.7.4 dd Imaging	24
2.7.5 Android Debug Bridge (ADB) and SDK Tools.....	24
2.7.6 Android Device Backup Applications	25

	Page
2.7.7 Nandroid Backup Acquisition.....	25
2.7.8 Guidance Software.....	25
2.7.9 AccessData Software.....	26
2.7.10 Paraben Device Seizure.....	26
2.7.11 USB Copy via MTP.....	26
2.8 Comparison Between Mobile and Traditional Forensics.....	27
2.9 Connectivity to Device.....	30
CHAPTER 3 METHODOLOGY.....	31
3.1 Overview of Experimental Design.....	32
3.2 Criterion for Errors.....	33
3.3 Hypothesis.....	33
3.4 Data Collection.....	34
3.4.1 Tablet Nominal State.....	35
3.4.2 Forensic Workstation Setup.....	35
3.4.3 Evidence Corpus.....	37
3.4.4 Hashing Tools Used.....	38
3.5 Initial Compatibility Testing.....	38
3.6 Tool Specific Procedures.....	39
3.6.1 ADB Procedures.....	40
3.6.2 DDMS Procedures.....	42
3.6.3 Mobile Phone Examiner Plus Procedures.....	46
3.6.4 Oxygen Procedures.....	49
3.6.5 Windows Native MTP over USB Procedures.....	57
3.7 Limitations and Bias.....	57
CHAPTER 4 FINDINGS.....	59
4.1 Overview of Collected Data.....	59
4.2 Feature Error Rates.....	60
4.3 Description of Errors Encountered.....	61
4.4 Error Rates and Sample Size.....	62
4.5 Android Tablet Acquisition Technologies.....	62
4.6 Differences in Presentation of Acquisition Results.....	65
CHAPTER 5 CONCLUSIONS, DISCUSSION, AND RECOMMENDATIONS.....	67
5.1 Discussion.....	67
5.1.1 Summary of Relative Error Rates.....	68
5.1.2 Discussion of Hashing Issues.....	68
5.1.3 Lack of Physical Imaging.....	69
5.1.4 Meaningfulness of Error Rates.....	69
5.2 Conclusions.....	70
5.3 Limitations.....	70

	Page
5.4 Significance.....	74
5.5 Recommendations.....	72
5.5.1 Common Root Cause Investigation	72
5.5.2 Dealing With Software Updates	73
5.5.3 Automated Error Testing	73
5.6 Summary	73
APPENDICES	
Appendix A: Evidence Corpus Master List	80
Appendix B: Table of Error Conditions Per Trial	83

LIST OF TABLES

Table	Page
Table 1.1 Android Platforms (Android Developers, 2011b)	11
Table 2.1 Selected Android Imaging Technologies' Affordances and Constraints.....	29
Table 3.1 Tablet Device Characteristics	34
Table 3.2 SDK Options and Dependencies.....	36
Table 3.3 File Types in Corpus.....	37
Table 3.4 Evidence Corpus Creation	58
Table 4.1 Feature Error Rates Per Tool Tested.....	60
Table 4.2 List of Android Forensic Tools Discovered	62
Table 4.3 Android Tablet Forensic Tool Categories	64
Table 4.4 List of Reporting Elements Per Tool Tested	65

LIST OF FIGURES

Figure	Page
Figure 2.1 Android Operating System Layers (Chen, 2008)	12
Figure 3.1 ADB Package Options.....	36
Figure 3.2 Output of Mount Command on Tablet	41
Figure 3.3 Windows Shell Command Used.....	41
Figure 3.4 MD5 Hash Shell Code Used.....	42
Figure 3.5 DDMS GUI	43
Figure 3.6 Choosing DDMS File Explorer	44
Figure 3.7 DDMS File Explorer Dialog	45
Figure 3.8 MPE+ Device Selection	47
Figure 3.9 MPE+ Error Dialog	48
Figure 3.10 MPE+ File System Extraction Selection	49
Figure 3.11 Oxygen Device Connection.....	50
Figure 3.12 Oxygen Wizard.....	51
Figure 3.13 Oxygen Wizard with Device Information	52
Figure 3.14 Oxygen Connection Wizard	53

Figure	Page
Figure 3.15 Oxygen Device Selection	54
Figure 3.16 Oxygen Data Selection	55
Figure 3.17 Oxygen Progress Bar	56

ABSTRACT

Harshbarger, William G. M.S., Purdue University, August 2012. Android Tablet Forensic Tool Testing. Major Professor: Marcus Rogers.

In order to scientifically demonstrate a measurable error rate in the operation of forensic imaging tools, any technology used must be empirically verified. A methodology is introduced to provide for measuring software logical imaging error rates on an Android tablet. This study observed no error rate in four of five tools tested, yet some measurable rate did exist for one tool. Additionally it was found that forensic imaging tools may have error rates comparable to other file transfer tools.

CHAPTER 1 INTRODUCTION

1.1 Background

Mobile devices, namely cellphones, have had a growing adoption rate for at least the last two decades. In fact, at the time of this writing, the United States experienced over 100% saturation of mobile devices per population (CTIA, 2011). Although simpler, telephony-only cellular devices have existed for many years, smartphones are rapidly becoming the mobile device of choice for consumers. Specifically, the market share of Android-based smartphones had increased from 22.7% in 2010 to 38.5% in 2011 (Gartner, 2011). While the tablet adoption rate was far behind other mobile devices, it has experienced nearly double growth between May 2010 and May 2011 (International Telecommunications Union, 2011).

Tablet devices, which occupy a middle ground in physical size between handheld devices and laptops, share many characteristics with smartphones. For instance, tablets often have touch screen interfaces and wireless data capability. Additionally, tablets enjoy a similar convergence of features like geo-location, photographic capability, increased data storage, and additional wireless connectivity (Godwin-Jones, 2008).

Feenberg (2006) suggested that technology such as tablets comprise part of an interwoven system of designers, technology and users. In this system designers create technology that users can apply to many different and perhaps unforeseeable ends. Users

of tablets may make use of these devices towards beneficial or malicious ends. Such misuses of digital systems can generally be classified as incidental, sourced, or targeted to the device or data (Casey, 2004). Detecting and analyzing these malicious and criminal uses of digital systems are a primary focus of digital forensics.

Due to the nature and quantity of evidence that can be found on mobile devices, Android-based mobile devices are of great forensic value (Lessard & Kessler, 2010). Digital evidence such as personal emails, photos, audio, text, SMS, networking configurations, geo-location, network remnants, etc. have been important to many law enforcement investigations (Ahmed & Dharaskar, 2008; Cardwell, 2011; Casey, 2004; Hoog, 2011; Lessard & Kessler, 2010).

While much work has been done regarding mobile forensic analysis tools and methodologies, tablets have not yet received the same amount of attention. Factors contributing to this lag in research may include the newness of these devices (Thing, Ng, & Chang, 2010) and the currently lower adoption rate by consumers, which was estimated at around 8% of adults in the United States in May of 2011 (International Telecommunications Union, 2011), perhaps making tablets less prevalent in forensic cases.

Digital forensics, as a forensic science, must meet certain criteria and rigor in order to be considered a peer with the more established sciences like chemistry. These practices are necessary for forensic tradecraft as well as admissible evidence to have empirical backing. Evidentiary standards include the federal Daubert ("*Daubert v. Merrell Dow Pharmaceuticals*," 1993) standard and the Federal Rules of Evidence, even though judicial discretion may override these requirements. One component of ensuring

that evidentiary requirements for scientific evidence are met is testing the tools used in gathering such evidence for failure and error rates. Tool testing is simply good scientific practice and allows for empirical backing of statements on evidentiary considerations like error rates.

Due to the lack of research explicitly focused on mobile digital forensic tool error rates, this thesis concentrated on determining what technologies are available to practitioners for imaging an Android based tablet. By applying a testing methodology, error rates associated with these imaging tools related to evidence acquisition was described. It was hoped that the research provided for more rigorous standards for Android tablet evidence by discovering any errors associated with imaging tools.

1.2 Scope

The proposed research focused on five technologies for acquiring a physical or logical image from a single make and model of Android tablet, specifically the Samsung Galaxy Tab, 32 GB, model GT-P7510, running Android version 3.2 (Honeycomb), Linux kernel 2.6.36.3. By narrowing the research to a single make and model of device, additional confounding variables should have been reduced and tests for specific functionality such as a physical image can be more accurately measured. Various image acquisition tools were used in order to compare the relative success and failure rates and presentation of the results of the processes.

1.3 Significance

The significance of this research has to do with assessing the forensic “correctness”, or in other words accuracy and precision of current technologies for acquiring a logical image of an Android based tablet. While tool testing is common among the physical sciences, forensic sciences and specifically digital forensic science are undergoing a shift (National Academy of Sciences, 2009) towards the establishment of more rigorous scientific standards. A key component of this shift is the systematic testing of forensic tools to obtain error rates and ensure admissibility. The relevant standards of admissibility for federal evidence are set forth in precedents like Daubert (*"Daubert v. Merrell Dow Pharmaceuticals,"* 1993) and the Federal Rules of Evidence. The Supreme Court of the United States established this standard in order to assure that evidence purporting to be based in science can be tested and proven to be so.

1.4 Research Question

The proposed research is guided by the following question:

- What are the relative success and failure rates among selected Android tablet evidence acquisition technologies?

Additional sub-questions that further frame this research include:

- What technologies exist to acquire a logical or physical image?
- How do the technologies differ in their presentation of results?

The primary question goes to the heart of the problem of knowing tool error rates with respect to Android tablet devices. In other words, establishing a baseline error rate for different tools relative to each other was of primary concern in this study. Informative questions helped guide the descriptions of available tools, and how the data was presented

1.5 Assumptions

This research relies on the following assumptions: first, that there was in fact testable error rates due to intrinsic properties or the implementation of the tools used; second, it was assumed that the differing tools would be comparable enough to be able to take reliable evaluations. In other words, it is assumed that the outputs of the tools had enough commonality to provide suitable data to use as metrics across all tools. Third it was assumed that errors may be random in nature or deterministic based on coding bugs. Fourth it was assumed that while hash tools used were cross verified initially, that subsequent uses would not error.

1.6 Limitations

As with any research, there are limitations to the possible depth and breadth of study. The limitations of this project are as follows: First, due to time and resource constraints, an exhaustive test of all types of devices with the acquisition tools was not undertaken. Secondly, a sample of the possible range of functionality of the technologies was tested, again due to time constraints. A truly random sample of evidence is unlikely to be feasible in the context of this study, so pre-configured evidence was used as a

deterministic measure, meaning that the evidence and its properties and locations was known to the experimenter beforehand. Finally due to features available in the tools tested, only logical imaging was tested.

1.7 Delimitations

The delimitations of this research are that only one specific variant of a device running one version of Android was tested. Also, even though the technologies chosen may have many features, only the acquisition functionality was tested.

1.8 Definition of Key Terms

ADB/Android Debug Bridge: A tool included with the Android software development kit that allows console access to a device from a workstation, typically over USB (Android Developers, 2011a).

Android: An operating system based on the Linux kernel, maintained by the Open Handset Alliance (Google, 2012).

Apache/Apache Software Foundation: A non-profit organization that provides intellectual property and license guidance for open source projects (Apache Foundation, 2011).

Logical image: A sub portion of a physical image, or individual evidence elements such as files or folders (Ayers, Jansen, Moenner, & Delaitre, 2007).

Nandroid: A backup mechanism included with Android recovery boot mode (Hoog, 2011).

Open Handset Alliance: A consortium, led by Google, of mobile device carriers, manufacturers that support mobile device standards (Open Handset Alliance, 2007).

Physical image: A bit-copy, which in theory provides an exact copy of the original storage media (Ayers et al., 2007).

Root: A built in administrative account included with Linux.

SD/SD Card: Secure Digital, a type of flash memory based physical media, which is usually removable by the user and comes in a variety of physical and logical sizes and is commonly used on mobile equipment (SD Association, 2012).

Smart Phone: A cellular telephony device, which also serves as a general purpose mobile, computing device, which provides extended data and processing capabilities (Merriam-Webster, 2012).

Tablet/tablet device/android tablet: A small-scale digital device or mobile computing device that shares many characteristics of cellular smart phones such as data connectivity, storage and general purpose computing capability that uses touch input instead of a keyboard (Dictionary.com, 2012).

YAFFS/YAFFS2: Yet Another Flash File System/ Yet Another Flash File System 2.

These are file systems designed for use with Android and flash-based storage media. (Manning, 2002)

1.9 Summary

Android-based tablets were a recent introduction to mobile devices and their adoption rate was rapidly growing. As with any technology, tablets can be used for good

or ill. In order for forensic practitioners to obtain and present reliable evidence, error rates of tools must be explored and documented.

CHAPTER 2 REVIEW OF RELEVANT LITERATURE

2.1 Background

This chapter reviews relevant literature with respect to mobile device forensics specific to Android-based devices. Additionally, the technologies employed for mobile device forensics on Android devices was explored. It is important to note that the majority of existing research found has focused on Android smartphones rather than tablets. While Android tablets share many common characteristics of Android smartphones, tablets in general often lack cellular telephony features in lieu of cellular data or IEEE 802.11 wireless data connectivity. The relevance of Android tablet devices in forensic investigations would be expected to follow a similar increase in significance to investigations as has been seen with other mobile devices.

2.2 A Brief History of Android to Present

The Android mobile operating system, an existing project acquired by Google in 2005 (Elgin, 2005), has since has expanded into the mobile consumer electronics device market and currently holds a rapidly growing install base. One estimate suggests that approximately 45% of U.S. mobile wireless subscribers uses Android-based devices as of September 2011 (comScore, 2011). Given the current rate of adoption, it has been estimated that mobile phone usage may have outpaced even the rapid growth of Internet

users between at least 2001 to 2007 (Chen, 2008). These estimates support the assumption that mobile devices, and specifically Android devices, will become increasingly important in forensic investigations as consumer adoption continues to increase.

The official release of Android was announced by Google in November 2007, as an open-source, Apache 2.0 licensed project (Chen, 2008). Concurrent with the release of Android was the announcement of the Open Handset Alliance, which is a consortium to support mobile device standards, led by Google and initially 34 wireless carriers, hardware manufacturers, software companies and other enterprises (Chen, 2008; Open Handset Alliance, 2007).

While the first Android cellular phones were released shortly after the announcement of the operating system and consortium in 2007, the first Android tablets did not appear until mid to late 2009 (France, 2009). Perhaps given this later introduction, Android based tablets are not currently as prevalent as Android smartphones or Apple's iPad tablet (Crothers, 2011). Despite this apparent lag, there has been overall growth in adoption of tablets among adults in the U.S. (International Telecommunications Union, 2011).

The several versions of Android, codenamed by names of sweets, include Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, and Ice Cream Sandwich (Android Developers, 2011b). These names map to operating system version numbers, provided by the Android Developers website (2011) as shown in Table 1.

Table 1.1. *Android Platforms (Android Developers, 2011b)*

Platform	Codename
Android 1.5	Cupcake
Android 1.6	Donut
Android 2.1	Eclair
Android 2.2	Froyo
Android 2.3.x	Gingerbread
Android 3.x	Honeycomb
Android 4.x	Ice Cream Sandwich

As the above table shows, there are currently several major versions of Android that a forensic analyst may encounter. Even though the use of Android tablets is increasing, that is no guarantee that manufacturers will standardize on one specific version, and undoubtedly there will be many more versions forthcoming.

2.3 Android Operating System Architecture

As the practice of digital forensics requires at least basic knowledge of the operating systems being investigated, a cursory introduction of Android's architecture is presented in this section. However a comprehensive architectural review was beyond the scope of this literature review.

The Android operating system is built upon the Linux kernel and is commonly described as consisting of “layers” of software that build upon a base consisting of the kernel and hardware drivers (Chen, 2008; Distefano, Gianluigi, & Pace, 2010; Vidas, Zhang, & Christin, 2011). The next layer is termed the libraries layer. This layer provides services such as SQLite, WebKit, SSL, and others (Chen, 2008; Distefano et al., 2010; Vidas et al., 2011). At the same layer as the libraries, but abstracted from the kernel by the libraries layer sits the Android Runtime, containing core libraries, and the Dalvik

Virtual Machine, which provides for execution of applications (Chen, 2008; Distefano et al., 2010; Vidas et al., 2011). Above the libraries and runtime layer is the application framework layer, which provides resources like the package manager and telephony manager (Chen, 2008; Distefano et al., 2010; Vidas et al., 2011). Above the application framework is the applications layer, which contains what most users would consider ‘apps’ such as the browser or dialer (Chen, 2008; Distefano et al., 2010; Vidas et al., 2011). The following Figure, 2.1, presented by Chen (2008) illustrates this layered approach to the operating system.



Figure 2.1 Android Operating System Layers (Chen, 2008)

The Android operating system also includes security features such as setting application permissions at install time (Distefano et al., 2010), and sandboxing of applications (Distefano et al., 2010; Vidas et al., 2011) in order to isolate the processes. Linux file permissions (Shabtai, Fledel, Kanonov, Dolev, & Glezer, 2010) and the

restriction of superuser or root access to users are additional security measures built in to the operating system. This default restriction of root permissions is perhaps best evidenced in the multiple forensic approaches that require root to be granted to the investigator (Cardwell, 2011; Hoog, 2011; Lessard & Kessler, 2010; Shabtai et al., 2010; Vidas et al., 2011).

2.4 Android Tablet Nonvolatile Storage

As NAND flash based media is prevalent in Android devices for general-purpose non-volatile data storage, it is the target for many forensic acquisition techniques (Cardwell, 2011; Distefano et al., 2010; Hoog, 2011; Lessard & Kessler, 2010; Me & Rossi, 2008; Thing et al., 2010). The following section gives a brief overview of NAND flash based storage on Android devices, and explains differences pertaining to functionality, data storage and data retrieval from such devices. Additionally the file systems employed on Android mobile devices will be described.

2.4.1 NAND Flash Memory

The hardware of flash memory fundamentally differs from magnetic media by not having any physically moving parts and by not being based on magnetic storage (Hoog, 2011). For example, flash memory has no moving parts, meaning input/output (i/o) operations happen in a special electronic circuit instead of a rotating platter and read/write arm. These circuits are grouped into units called blocks, which is analogous to sectors in traditional hard drives. Specifically, each block (unit of storage) can only be written to a relatively finite number of times (Hoog, 2011; Samsung Electronics Co.,

2006), typically between 10,000 or 100,000 cycles depending on the hardware type (Hoog, 2011). Though one could assume future solid state media could mitigate these cycle limitations, the Android operating system uses methods and algorithms to overcome this problem, including a wear-leveling algorithm that ensure each block is written to nearly an equal number of times to prevent hardware degradation.

Additionally, the Linux kernel was designed for character or block devices (Lessard & Kessler, 2010) rather than raw flash hardware and so requires an abstraction layer. In Android the operating system serves this purpose and accesses the raw flash media without a hardware controller to translate. Instead, translation is performed in software at the Flash Transition Layer (FTL) (Hoog, 2011). This transition layer powers the Memory Technology Device (MTD) system, which provides this abstraction of the flash memory to the operating system. Additionally the MTD system enables easier development and provides functionality such as error correcting and wear leveling (Hoog, 2011; Lessard & Kessler, 2010).

For data operations (typically read, write, delete) at the hardware level, NAND differs significantly from magnetic storage. Specifically, data either read from or written to NAND flash can only be read or written in units termed “pages” or “chunks”, typically a multiple of a power of 2 unit (2^n), that is between 512 and 2048 bytes in size (Hoog, 2011). Alternately, deletions at the hardware level happen in blocks where blocks = $n \times$ pages in size (Hoog, 2011; Samsung Electronics Co., 2006). Hoog (2011) describes a typical block size of 128KB, which was described as being composed of 64×2048 -byte pages. As one can see, deletions end up affecting n times more blocks than writes affect.

For example, it could be extrapolated that writes happen in 2048 byte increments, while deletes happen in 128 kilobyte increments.

As described above, one of the significant implications of how data is read or written versus erased in flash is that data is erased in much larger increments than writes. As a result, delete operations affect a much larger number of NAND cells than do write operations. Due to the physical read/write count restrictions of the media, this means that a forensic examiner is very likely to find deleted files. This is due to the fact that the much larger block will not be deleted immediately. This delay is from the wear leveling algorithm in order to conserve the duty cycle of the NAND hardware itself (Schmitt, Spreitzenbarth, & Zimmermann, 2011).

2.4.2 File Systems

The physical differences in storage and the operational usage of mobile devices mean that there are considerations to make in selecting an appropriate file system for flash based mobile devices. There have been several attempts to create a file system that accommodates the unique functional requirements of NAND flash and the requirements of mobile devices while maintaining compatibility with the Android operating system.

One is the assumption is that mobile devices may experience a sudden loss of power through battery drain/removal or damage to the device (Samsung Electronics Co., 2008). In order to overcome this liability, file systems commonly used in Android devices implement journaling, which is a method of logging data operations such that they can be recreated if necessary in order to preserve the integrity of the file system.

Initially, Android was released with the YAFFS file system which supported older, smaller capacity NAND flash memory and specified 512 byte page sizes (Distefano et al., 2010). As newer NAND chips with larger page sizes became prevalent in devices, YAFFS2 was developed, which specified 2048 byte page sizes (Distefano et al., 2010). More recently, YAFFS2 has been replaced with the ext4 file system (Vidas et al., 2011) in Android 2.3 (Schmitt et al., 2011). Currently common file systems one could find on Android devices include YAFFS, YAFFS2, FAT, FAT32, ext3, ext4, and proprietary file systems (Hoog, 2011; Vidas et al., 2011).

While many of the above file systems are open-source, some manufacturers choose to create proprietary file systems. For instance Samsung, as a manufacturer of Android devices as well as NAND chips has created the proprietary “Robust File System” or RFS for its products. RFS is FAT32 compatible but also provides for journaling (Samsung Electronics Co., 2006).

In addition to the main file system, there is typically an additional file system for the recovery image that is a Linux “initramfs” containing a recovery partition. This recovery partition contains special system tools, such as data wipe or factory reset, and is typically only used by specifically booting into it. The booting contained in the ramdisk contains a header, initrd ramdisk or a secondary optional image (Vidas et al., 2011). While the recovery file system and image are useful tools, there typically isn’t user data stored there (Vidas et al., 2011).

One can see that multiple file systems must be expected by the forensic examiner and be supported by the tools employed. Due to this variability, some problems may arise for the examiner. For instance, YAFFS is not natively viewable on Windows (Distefano

et al., 2010), which means additional software must be used (Hoog, 2011). Additionally, SD card capable devices often employ FAT32 on the card (Hoog, 2011), which implies that an examiner may have multiple file systems to contend with on a single device.

2.5 Digital Forensic Science and Tool Testing

Scientific disciplines often make use of many types of technology to obtain observational data or perform experiments, ranging from pipettes to mass-spectrometers. Similarly, in digital forensics tools such as disk imagers or file carvers may be used. These technologies act as the eyes and ears of investigators and are essentially digital translators that provide the examiner with useful data. It is of utmost importance that the implementation and functions of the technology used in forensic investigations are sound and auditable. This essentially means that tools should work as advertised and be able to be verified as such. Also, the errors inherent in the tools or occurring through their use must be known. Testing is required to ascertain possible sources of error of the tools as they are used in forensic investigations. One such framework established for laboratory testing and accreditation is the ISO 17025 standard. This standard is widely used by laboratories to ensure procedures or tools are sound.

In order to ensure consistent, accurate, and repeatable use of technologies in forensic investigations, tools must be tested in order to account for error or failure modes that may detrimentally alter potential evidence. Digital forensics, while not born strictly from scientific disciplines, has since aspired to reach the standards and rigor of other scientific disciplines such as chemistry (National Academy of Sciences, 2009), and in this spirit has a concerted effort to understand the tools used by forensic investigators.

Forensics and digital forensics is not exempt from an expectation of scientific rigor by the academy. The Supreme Court ("*Daubert v. Merrell Dow Pharmaceuticals*," 1993) has also set forth specific criteria to establish validity and admissibility of scientific evidence at the federal level. This criteria includes among other measures, peer review, known error rates, and general acceptance by the scientific community (Baggili, Mislán, & Rogers, 2007; Hendricks, 2008). To this end, the computer forensic tool testing (CFTT) program was developed by the National Institute of Standards and Technology (NIST) under a National Institute of Justice (NIJ) grant. This program's goal is essentially to provide examiners with knowledge of forensic tools and errors so the examiners can attest to the validity of evidence presented (Ayers et al., 2007; National Institute of Justice, 2010). The CFTT approach is described as assessing a tool's specific functions individually, and developing a testing methodology for that function (National Institute of Standards and Technology, 2003).

Specifically NIST uses a ten-step process as defined on the CFTT website (National Institute of Standards and Technology, 2003): First, the tool is acquired; Second, the relevant documentation is reviewed; Third, the relevant test cases are chosen; Fourth, a test strategy is devised; Fifth, the tests are executed; Sixth, a report is produced; Seventh, a committee reviews the report; Eighth, the vendor reviews the report; Ninth, support software is published to the NIST website; Tenth, the NIJ posts the test report to its website. The NIST CFTT approach guided the methodology employed in this research, while it was not followed exactly.

2.6 The Nature of Errors in Science and Forensic Practice

Science in general and forensic science is anything but error-free. Allchin (2001) described the nature of errors in science and categorized them into types: Material, Observational, Conceptual and Discursive. One facet of the argument proposed by Allchin (2001) is that in order to address errors, the types of errors must first be articulated. In other words, to know the science is to know its shortcomings.

For example, Allchin's (2001) conceptual error types can be applied to the forensic practice of fingerprint identification. Benedict (2004) describes how a Seattle man was falsely accused of the 2004 Madrid terror attack. This accusation was based on, as the FBI asserted, a "100% match" for fingerprints found at the crime scene (p. 519). The real culprit in the attack was in fact an Algerian man whom the Spanish police later apprehended (Benedict, 2004). How and why did the FBI claim an absolute confidence in their results? This is in error that Allchin (2001) describes as a cognitive bias or entrenchment – the conceptual error type. In this instance the error centered on the belief that fingerprint analysis was foolproof and errorless. This entrenched belief was based on what were essentially decades of anecdotes of practitioners rather than empirical evidence. Even lacking basic scientific underpinnings such as a basic theory of uniqueness or false positive rates, fingerprint analysis is often generally asserted as infallible and trustworthy (Benedict, 2004) which led to false assertions. Though fingerprints may be useful identification sources, there is a need to ensure that the forensic sciences of fingerprint identification, along with all other forensic sciences, are based in empiricism.

Another type of error that may be found in forensic investigations is material errors (Allchin, 2001). These errors involve improper procedures or practices or methods that can, in some cases, alter what is being observed. For forensics this type of error can be fatal to evidence as it can call its provenance into question. For instance, the improper use or operational failure of a write blocker may make it possible for data to be altered, corrupted or deleted by an examiner.

Errors are implicit in the use of any technology to record observations. Saks & Koehler (2005) described how a measure of scientific meticulousness based in the practice of DNA typing is forcing other forensic sciences towards a more rigorous standard. Saks & Koehler (2005) explain how until recently, many forensic practices such as tool mark analysis, fingerprint, teeth marks, tire marks, etc. were regarded as being scientifically-based. In fact few, if any were grounded with theoretical or empirical underpinnings. In a survey of exonerations by DNA typing evidence, Saks & Koehler (2005) found that forensic science testing errors were implicated in 63% of the wrongful convictions. Additionally, false or misleading testimony by forensic scientists was implicated in 27% of the wrongful convictions. While the types of forensic practices (tire mark, tool mark, digital forensics) that were in error in these wrongful convictions were not discussed, it is clear that there are very real consequences regarding forensic errors.

One important part of understanding errors in forensic science is tool testing to determine error rates or modes (Baggili et al., 2007; Epstein, Tebbett, & Boyd, 2003; Lyle, 2010; National Academy of Sciences, 2009; National Institute of Justice, 2010; Pan & Batten, 2009). In order to meet the evidentiary requirements set forth by the judicial system, the tools used by digital forensic practitioners must at least be tested to determine

error rates and be subjected to peer review. Additionally, this testing research must be applicable and relevant to practitioners (Beebe, 2009).

2.7 A Survey of Android Device Acquisition Technologies

Several tools and methodologies were discovered in the literature pertaining to mobile device evidence acquisition. Nearly all the acquisition technologies found are software-based, with a few notable exceptions. This section will further discuss several of these technologies and their use. One important note is that these technologies often focus on the contents of the onboard flash memory and not other removable media such as SD cards or SIM cards. This is almost certainly because removable media already have more well-established acquisition methods.

2.7.1 Hardware Based Data Acquisition

Traditional forensics often involves physical removal and acquisition of storage media, such as a hard drive, so that evidence can be copied and verified using forensic technology (Lyle, 2010; National Academy of Sciences, 2009). Since mobile devices typically use flash memory for non-volatile storage, the problem of physically acquiring the contents of storage via hardware becomes more difficult.

A method for physical removal of mobile device storage include disassembly by de-soldering and chip extraction are described by Willassen (2005) and Hoog (2011). While Willassen's (2005) techniques proved fruitful in that data could be retrieved, an investigator would need access to specialized electronic tools and a mastery of performing delicate actions on microchips in order to use these techniques (Hoog, 2011).

Another, more common hardware-based access method used in cellular devices is a hardware port created by the Joint Test Action Group or “JTAG” (IEEE 1149.1) (Hoog, 2011; Thing et al., 2010; Vidas et al., 2011; Willassen, 2005). JTAG ports are used for diagnostics and allow direct access to memory hardware through specialized electronics. While JTAG is available on many cellular devices, it is not known how widely JTAG has been adopted by Android tablet manufacturers.

2.7.2 Methods That Require Rooting

Before describing software based digital forensic technology, a note on “rooting” mobile devices must be made. This is an important concept as most devices restrict root access, yet many forensic acquisition methods require root level access. As one can see these are in direct opposition and can present issues for the forensic examiner. Essentially, rooting involves gaining superuser or administrative access to a device. Rooting can be powerful, but can prove problematic and should be considered with much caution.

While many forensic methods may require root access (Distefano et al., 2010; Hoog, 2011), rooting is not generally required to use native tools like Android Debug Bridge or Nandroid. However, without root access, the system typically restricts full access to the data on a device (Hoog, 2011).

Gaining root often requires the exploitation of a flaw in a device’s operating system, may overwrite data during the process, and may leave device vulnerable to exploits. Additionally, rooting can pose issues with device compatibility as methods of rooting are typically device-specific (Lessard & Kessler, 2010; Vidas et al., 2011). Many methods to gain root access exploit flaws in the operating system that may be problematic

to justify in court. For instance, the exact workings of the exploit based rooting method may only be known to the method author and could not be described by an investigator (Vidas et al., 2011). It may also be problematic to explain in court why a method that some may construe as breaking or hacking a device was used. In most everyday instances this would not be a problem, however in forensics a practitioner may need to testify as to the exact workings of any technology used and as well justify their actions. Rooting a device, by definition, alters at least the portions of the operating system that will grant root access, but may also alter other data. The alteration of an unknown amount of data in unknown ways may make the evidence produced by forensic procedures inadmissible in court (Lessard & Kessler, 2010) or at least a point of contention.

2.7.3 Secure Digital (SD) Card Swap

Me and Rossi (2008) outline a methodology specific to Symbian devices that may also be applicable to Android devices. This methodology is somewhere between hardware and software-based acquisition in that it involves placing a special SD card containing forensic tools into, in the case of Me and Rossi's (2008) example, a Nokia 6630 Symbian smartphone. This method necessitates the availability of a compatible SD card port and would more than likely require root access on the mobile device.

Specifically, the tool written by Me and Rossi was loaded to the specially crafted SD card so that acquisition of the contents of the fixed memory of the device can be imaged locally to the contents of the SD card (Me & Rossi, 2008). Some disadvantages of this method included limited storage space on the SD card and the necessity to power cycle the device due to the inability of Symbian to hot-swap media (Me & Rossi, 2008).

One can also imagine that the physical location of the SD card could prove problematic as often times it is not easily accessible, for example if located under the device's battery.

2.7.4 dd Imaging

The Linux/Unix command `dd`, or its forensics variants `dcfldd` and `dc3dd` have long been favorite tools used to duplicate a drive in hard drive forensics. If an investigator can obtain a command line shell access to a device, common Linux tools such as `dd` may be used as a way to copy the contents of non-volatile memory to the SD card (Vidas et al., 2011). In addition, many other forensic software suites include `dd` to provide for the acquisition functionality of the suite (Lessard & Kessler, 2010). Additionally, Lessard and Kessler (2010) and Cardwell (2011) note that root access is necessary to obtain a full acquisition of the device or to copy off individual files from all parts of the file system due to file permissions restricting non-root access. Dd images are also widely readable by many different forensic software packages.

2.7.5 Android Debug Bridge (ADB) and SDK Tools

The Android Debug Bridge or “ADB” (Android Developers, 2011a) is another software tool, limited by root and file permissions, that can be used to acquire the contents of an Android device (Hoog, 2011). Using the “`adb pull`” command, an investigator may be able to copy off specified parts of the file system to a forensics workstation connected to the mobile device (Hoog, 2011). ADB pull may be of little value to an investigator needing to copy the full contents of non-volatile memory due to the aforementioned file permission restriction (Hoog, 2011). Another tool in the SDK is

Dalvik Debug Monitor Service, or DDMS, which is a GUI for developers to connect to a device. DDMS provides similar functionality as ADB pull for moving data.

2.7.6 Android Device Backup Applications

Several applications for making backups of user data on Android devices are available from the Android Market (Hoog, 2011). While the forensic relevance of these tools may be questionable, such applications may prove useful in acquiring user data. Again, root access is often required to make use of such applications. If used, it may be problematic to explain why an application was installed and used unless it can be determined that the application was already in place (Hoog, 2011).

2.7.7 Nandroid Backup Acquisition

Nandroid is a software option available on many devices from the recovery mode boot option. Nandroid has the capability to save data to at least a local SD card (Cardwell, 2011). Additionally, it may be possible to acquire a full NAND flash image including deleted items. This sounds very useful but other limitations exist. For example, Nandroid preserves the file system, so if a file system like YAFFS is on the device additional software may be required on the investigative workstation (Distefano et al., 2010).

2.7.8 Guidance Software

Commercial forensic software for Android data acquisition is available from Guidance Software's EnCase product (Cardwell, 2011; Hoog, 2011; Yates, 2010). Hoog (2011) briefly describes the acquisition process and noted its apparent quickness. While

the EnCase Neutrino product could acquire evidence, there were no indications regarding root access or a full physical image implying that perhaps only logical acquisition is available (Hoog, 2011).

2.7.9 AccessData Software

Access Data's Forensic ToolKit, or FTK software is another commercial option available to investigators and is referenced by Lessard and Kessler (2010), Cardwell (2011), Danker, Ayers, and Mislán, (2009), and Yates (2010). While the literature referenced FTK as a suite, no full peer reviewed feature list was found in literature.

2.7.10 Paraben Device Seizure

Cardwell (2011), Hoog (2011), Distefano (2010), Yates (2010) all make reference to Paraben Device Seizure software as a method to acquire Android device data. Hoog (2011) describes using Paraben and specifically states that the phone must be altered by placing it into debug mode prior to using Paraben, possibly changing data.

2.7.11 USB Copy via MTP

Although not intended as a forensic tool, acquiring files via the device's USB connection is another option for acquiring files. As is often the case, various combinations of devices, transfer protocols, storage media and device software may relegate an examiner to using basic tools to copy files. In this scenario, the tablet is connected to a Windows forensic workstation and files are extracted in a way that the device manufacturer intended a consumer to access the files. One such protocol used to

access device files over USB is the media transport protocol or MTP. MTP is commonly used on electronic devices to transfer the contents of storage, such as pictures or music files (USB Implementers Forum, 2007).

2.8 Comparison Between Mobile and Traditional Forensics

Perhaps the most obvious difference between standard computers and mobile devices is that of physical mobility and size. Storage hardware is another key difference as most mobile devices use flash-based memory rather than magnetic-based hard drives. It can be argued that mobile forensics has not attained the level of maturity level that traditional digital forensics has reached (Beebe, 2009) most likely to these new circumstances based in the technological differences and operational usage of mobile devices.

Mobile device forensics presents a challenge to the examiner as the devices have many different cables, firmware, and multiple power states instead of simply off or on (Owen & Thomas, 2011). Additionally, mobile forensics presents a different paradigm for investigators. For instance, to acquire evidence the tenets of traditional “static” digital forensic best practices must be violated (Cardwell, 2011; Distefano et al., 2010; Owen & Thomas, 2011). These well-established rules, such as principle 1 of the ACPO standard, that no data to be used in court can be changed by an investigator (Association of Chief Police Officers, 2007), is one such rule that must often be broken in mobile forensics. This violation is in order to acquire evidence of a mobile system which must remain powered on and “live” with the contents of volatile and non-volatile storage constantly changing (Ahmed & Dharaskar, 2008).

Another consideration for investigators is that many organizations and companies are requiring data encryption on mobile devices, which may further complicate investigations by hampering or preventing acquisition of evidence (Owen & Thomas, 2011). In fact even standard methods of verifying acquired evidence, such as cryptographic hashing are in question due to the volatile nature of mobile devices (Danker et al., 2009).

Anti-forensics and data hiding is one component where mobile and traditional forensics share a problem (Distefano et al., 2010; Ridder, 2007). Distefano et al. (2010) describe several techniques available to hide or obfuscate evidence including OS exploits, hidden folders, special apps, among other methods. While the literature did not contain much on anti-forensics, it is a possibility that must be considered by the examiner.

In summary, the literature shows that while errors may exist in any application of forensic science technology, digital forensics has not yet established the same corpus of knowledge of these errors, their nature, causes and rates. Additionally the newness of mobile technologies means that the technologies available to investigators, while growing, are not yet standardized. By studying technology that is currently available, this review guides the research to discover what is necessary for practitioners to know for scientifically based evidentiary standards, primarily error rates of these technologies. Additionally, a summarization of the tools discussed is presented in Table 2.

Table 2.1. *Selected Android Imaging Technologies' Affordances and Constraints*

Technology Name	Affordances	Constraints
Chip removal	Obtains the physical media for imaging. Can bypass file permissions.	Requires additional specialized knowledge and tools related to electronics. Risk damaging physical media or device.
JTAG	Allows direct access to hardware via specialized port. Can bypass file permissions.	Unknown adoption rate of JTAG on tablets. Requires additional electronic equipment and knowledge.
SD card/media swap	Images acquired on device. Novel methodology. Easy to acquire logical images. Special SD card can be inserted on-scene.	Requires specialized software. Requires accessible SD card slot. May require rooting. Limited SD storage capacity. May require power cycle to access SD .
dd, dcfldd	Commonly used, standard Linux tool. Can image to several media types. Image files widely supported in analysis tools.	May require rooting. Restricted by file permissions.
ADB	Included with Android SDK. Can back up any desired part of device data.	May require rooting. Restricted by file permissions.
Backup Applications	Can obtain user data.	May require rooting. Restricted by file permissions. May require installation by investigator.
Nandroid	Usually available on device. Capability to perform full device backup. Preserves file system.	May require rooting. Restricted by file permissions.
Guidance EnCase	Commercial support. Logical imaging support.	May require rooting. Restricted by file permissions.
Access Data FTK	Commercial support.	May require rooting. Restricted by file permissions.
Paraben Device Seizure	Acquisition of user data.	May require rooting. Restricted by file permissions. May require USB debug mode.

Table 2.1. Continued

Technology Name	Affordances	Constraints
USB Copy over MTP	Supported by device	No specific consideration for forensic integrity. USB debug mode required.

2.9 Connectivity to Device

The above methods describe a general set of tools that may be used on Android based devices, however tools should be selected that are known to work with a specific device. For instance, while the Galaxy can connect via USB to a forensic workstation, the device itself does not support mounting of a USB file system as one may encounter on a different device, or a mass storage device like a USB thumb/flash/external drive. Instead, the Galaxy uses a Microsoft proprietary protocol called Media Transport Protocol (MTP), which is a protocol designed for transfer of media files over USB (USB Implementers Forum, 2007). This means that Unix/Linux based forensic systems may not be able to natively connect to the device to acquire data. More of these considerations, specific to the Galaxy will be described in the following chapters.

CHAPTER 3 METHODOLOGY

This chapter describes the methodology that was employed to gain insight into the following research question: What are the relative success and failure rates among selected Android tablet evidence acquisition technologies? The goal of this study was to make observations of any potential errors in the acquisition of a known data set from an Android tablet in the furtherance of establishing error rates for the digital forensic practitioner community. Measurement of error rates provide for a general error rate and feature error rate (Baggili et al., 2007) for various software tools recorded in the findings.

The device to be tested, the tools and the procedures used to conduct this research used are described in this chapter. As time and resources constrained the number of tools that could be tested, a subset of the combinations of device and acquisition tools were selected for testing based on the criteria of relevance in literature and compatibility with the equipment used.

Specifically, five different software-based file acquisition tools were tested on a single tablet device. The functionality tested included logical acquisition of a set of known data. Additionally, for reasons such as compatibility to be discussed in Chapter 4, physical imaging was not available to test in this study.

3.1 Overview of Experimental Design

As this study aimed to measure the success or failure rates and determine if tools differed significantly in error rates, the experimental design centered on the measurement and verification of acquisition of files. By testing the functionality of file acquisition, an error rate could be determined for forensic acquisition. For the purpose of this study, a trial was defined as a single use of the logical image acquisition feature of a tool, or in other words a copying of the individual files of the evidence corpus. The outcomes of the trials resulted in either success (verified hashes of corpus files) or failure (all other outcomes).

In total, each tool was used to perform 20 file acquisition trials using procedures specific to each tool as outlined later in this chapter. In turn these trials extracted the predefined evidence corpus containing 25 varied files created internally and external to the tablet. After recording the results of the hashes of the acquired files, any errors encountered were logged and recorded as the feature error rates or FER (Baggili et al., 2007).

Initially this study set out to investigate error rates of logical and physical features of Android tablet imaging technologies, yet of the tools found and categorized, none included physical imaging support. In light of this finding, only logical imaging features were tested.

3.2 Criterion for Errors

The criterion for determining errors was as follows. Any deviation from a cryptographically hashed and verified-as-unaltered acquisition of data was considered an error. In other words, if the tool did not produce a forensically sound acquisition of a file based on the widely accepted criteria of cryptographically hashing evidence, an error was encountered. Alternately, if the tool fails to produce any evidence, an error had occurred. Put another way, the only non-error condition was when a valid cryptographically verified image was created. The technologies tested were not required to include hashing functionality as a feature, and in those cases standard hashing tools such as md5sum were used to verify the acquisition.

3.3 Hypothesis

The main hypothesis of this study was that forensic acquisition tools for Android tablets may produce errors at a measurable rate. Errors are defined as any condition other than a cryptographically verified acquisition of a file. The null and alternative hypothesis that guided this research were:

H_0 : Android tablet imaging tools do not produce errors when acquiring data.

H_a : Android tablet imaging tools do error when acquiring data.

Hypothesis testing was to be performed through 20 file acquisition trials per tool tested on a known evidence corpus consisting of 25 files. After all trials were completed, the acquired files' cryptographic hashes were compared to the originals. Any deviation from an exact hash match was recorded as an error.

3.4 Data Collection

One of the popular models of current Android tablets available during this research was the Samsung Galaxy Tab 10.1. The specific device tested used in this study was a Samsung GT-P7510, which had been factory reset prior to data collection. This specific device was tested as it represented a contemporary version of Android tablet devices, had a tablet specific operating system, and its operation was familiar to the author. The device was also not ‘rooted’ because this was the factory default condition. The device characteristics are included in Table 3.

Table 3.1. *Tablet Device Characteristics*

Operating System	Android 3.2 “Honeycomb”
Kernel	2.6.36.3 se.infra@SEI-30 #1
Build Number	HTJ85B.UEKMP P7510UEKMPs.d
Rooted	No
Wi-Fi	Yes, but disabled
3G/4G	No
USB Debugging	Enabled

As shown above, the operating system was Android 3.2 “Honeycomb”, Kernel 2.6.36.3 se.infra@SEI-30 #1. Build number of the tablet was HTJ85B.UEKMP P7510UEKMPs.d. Initial setup of the factory reset tablet included uploading of evidence files, creation of local evidence files via the camera/video application, enabling of USB debugging and disabling of Wi-Fi. All other settings were default.

3.4.1 Tablet Nominal State

To help control for effects of one set of trials on the next, the tablet was placed in a nominal state before each set of 20 trials. In this study, the term nominal state refers to a set of conditions that were set after factory resetting the device. This was performed as it was found during compatibility testing that some tools installed client software to the device. This state was manually reverted to between the end of a set of trials and before the start of a subsequent set of trials.

3.4.2 Forensic Workstation Setup

Forensic tools used in this study were installed on a Windows 7 forensic workstation, as all tools tested were software based and only compatible with Windows. The forensic workstation upon which testing was performed was a 24 inch iMac running OS X Lion, which contained a 250GB boot camp partition created via the wizard, which was then loaded with Windows 7 and updated fully as of March 15, 2012. Windows was chosen as the primary research platform as most of the acquisition tools found in literature were Windows based.

Samsung's USB drivers for the tablet, version 1.3.2360.0 for the Galaxy tablet were installed to the workstation from the manufacturer's website. The Google Android SDK, which depends on the Oracle Java Development Kit (JDK) was downloaded and installed the Windows 7 system with the following options selected. In the SDK installer, the packages installed were: Android SDK Platform-tools, SDK platform Android 3.2-API 13, Samples for SDK API, Google APIs, and Google USB Driver package, as shown in Figure 3.1. Table 4 contains the package options and dependencies installed.

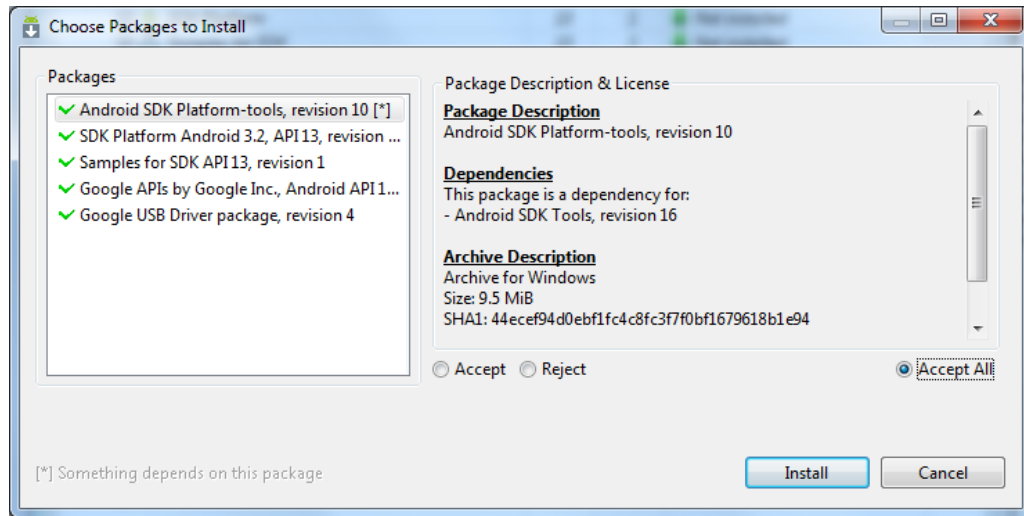


Figure 3.1: ADB Package Options.

Additional software installed included Microsoft Office 2010 and the Dropbox local synchronization app, were obtained from laboratory resources and the vendor's website, respectively.

Table 3.2. *SDK Options and Dependencies*

USB Driver	1.3.2360.0
SDK Packages Selected	Android SDK Platform-tools SDK platform Android 3.2- API 13 Samples for SDK API Google APIs Google USB Driver package
Additional Software	Dropbox Client Microsoft Office
Dependencies	Oracle Java Development Kit (JDK)

3.4.3 Evidence Corpus

25 files were generated as a known body of evidence in order to provide files to acquire from the newly factory reset device. Specifically, five of each type of file was created, also shown in Table 5: MPEG Layer 3(mp3 audio), MPEG Layer 4 (mp4 video), JPEG (jpg image), Microsoft Word documents, and Adobe Portable Document Format documents. In order to test acquisition of files created internally or external to the device, screenshots, pictures and videos were generated on the device itself in addition to documents uploaded to storage. The rationale for choosing these file types was that they were able to be created on the device (MP4, JPEG) or were commonly found as email attachments (Word, PDF), and often contained on a device (MP3, JPEG, MP4).

Table 3.3. *File Types in Corpus*

File Type	Content
MP3	Audio
MP4	Video, with audio
JPEG	Photograph, Screenshot
MS Word	Text
PDF	Text

Text files were populated with placeholder “lorem ipsum” text to provide a more realistic file size as well as some amount of textual content. Each of the generated files was then cryptographically hashed using DigestIT, a Windows tool, selected as it was familiar to the researcher and to provide an additional tool for producing hash values. Additionally, files were hashed using md5sum and sha1sum on an OS X Lion system to provide a cross check for validity. A master list of the original files including the file type, name, location, MD5 and SHA1 hashes was recorded with the names and hashes as a

reference point for comparison between trials. This master list is included in Appendix A. The results of all trials were placed in a Dropbox sync folder to provide for replication and backups of the data.

3.4.4 Hashing Tools Used

During the initial hashing of the elements of the evidence corpus, two separate tools were used in addition to the built-in hashing functionality of the acquisition tools in order to cross-check that the hashes were accurate. These tools were DigestIT 2004, a Windows platform tool, and the md5sum/sha1sum package installed on the OS X system. These tools were chosen as the author was familiar with their use and they offered different platforms and implementations of the cryptographic hashing algorithms to reduce the chance that one may have had errors. MD5 and SHA1 hashing algorithms were chosen due to the frequent references to these tools in the forensics literature (Danker et al., 2009; Lyle, 2010; Me & Rossi, 2008; Pan & Batten, 2009). Other hashing tools used included those provided by the forensic software vendors, if applicable. This included MPE + (via the FTK imager tool) and Oxygen Forensic Suite.

3.5 Initial Compatibility Testing

A challenge to forensic researchers at the time of this research was the myriad combination of devices, operating systems, cables, and transport mechanisms available across the mobile device spectrum. For the purposes of this study, cursory testing of tools to determine feasibility was performed. The initial pool of tools tested included Paraben Device Seizure, ViaForensics VIAExtract, AccessData MPE +/-FTK Imager, Android

Debug Bridge (ADB), Dalvik Debug Monitor Server (DDMS), FTK 2.0, dcfldd, dc3dd, Android File Transfer, gMTP, and the native MTP functionality of Windows 7. In the end, five tools were chosen from a pool of tested tools. The results of this testing will be discussed in Chapter 4.

For this study, five Android evidence acquisition technologies were selected based on a review of the literature and preliminary compatibility testing between the tablet and the forensic workstation. The criterion for selecting these tools was as follows. First was assessing in literature how widely used the tools are by digital forensic practitioners for the acquisition of images of Android devices. Second was a lack of research related to error rates for each tool with respect to Android tablet imaging. Third was compatibility between the mobile device and the potential forensic workstations.

As tools' compatibility across mobile devices was not readily distinguishable in the literature, this left the selection criteria much wider. In other words, the selection is based on a gap in the current knowledge of with respect to these tools, along with the tool's use by practitioners. It was found that the five tools best suited for this study were: Android Debug Bridge (ADB), Dalvik Debug Monitor Server (DDMS), Mobile Phone Examiner Plus (MPE +)/FTK Imager, Oxygen Forensic Suite, and the native MTP features of Windows 7. All of these tools are Windows-based, although final hashing of files not hashed by the tools was performed on OS X Lion.

3.6 Tool Specific Procedures

A known set of cryptographically hashed evidence files, listed in Appendix A was created as an evidence corpus. These known files were placed on the tablet in specified

locations on internal storage. In this case, all storage comes from a pool of flash memory and the file system directory structure and permissions provide the only separation between the system and user space. Files were created on and off the device to account for realistic scenarios of file creation.

3.6.1 ADB procedures

Android Debug Bridge (ADB) was a multi-purpose tool that is included with the optional platform-tools package of the Android Software Development Kit (SDK). One such feature of ADB often referenced in literature was file transfer via the adb “pull” option. The steps taken to perform the trials via ADB pull were as follows.

First, ADB was installed from the Android SDK as described in the workstation setup section. Next, nominal-state tablet was connected to the workstation using the proprietary cable. The tablet was placed into recovery mode by pressing volume down while holding the power button to boot into recovery. Upon boot, ADB was executed on the forensic workstation by running “adb devices” to list all connected Android devices. As the device was identified, the command “adb shell” was executed to access the Linux shell on the tablet itself. By running the “mount” command in the shell, it was ascertained that the tablet contained several file systems as shown below in Figure 3.2.

```

shell@android:/ $ mount
mount
rootfs / rootfs ro,relatime 0 0
tmpfs /dev tmpfs rw,nosuid,relatime,mode=755 0 0
devpts /dev/pts devpts rw,relatime,mode=600 0 0
proc /proc proc rw,relatime 0 0
sysfs /sys sysfs rw,relatime 0 0
tmpfs /tmp tmpfs rw,relatime 0
/dev/block/mmcblk0p4 /system ext4 rw,relatime,barrier=1,data=ordered 0 0
/dev/block/mmcblk0p5 /cache ext4 rw,nodev,noatime,nodiratime,barrier=1,data=ordered 0 0
/dev/block/mmcblk0p10 /preload ext4 rw,nodev,noatime,nodiratime,barrier=1,data=ordered 0 0
/dev/block/mmcblk0p8 /data ext4 rw,nodev,noatime,nodiratime,barrier=1,data=ordered 0 0

```

Figure 3.2 Output of Mount Command on Tablet

In order to facilitate the automation of file acquisition, a Windows shell command was used to run the ‘adb pull’ command 20 times. The shell command code to perform n number of trials, which was executed from the directory location of adb on the system, is as follows:

```

for /l %x in (1, 1, 20) do adb pull /mnt/sdcard "c:\adb_pull\pull%x"

```

Figure 3.3 Windows Shell Command Used

This for loop was used, and 20 trials were performed and the contents of the adb pull were saved in a directory named with the trial number. In order to hash the acquired files, an OS X Lion system with md5sum and sha1sum installed was used. The following shell command was used to iteratively hash the files from the Dropbox sync location as shown in Figure 3.4. A similar command was performed using sha1sum in place of md5sum to obtain SHA1 outputs.

```
find /Users/harshbwg/Dropbox/111THESIS/data/ -type f -print0 | xargs -0  
md5sum >> /Users/harshbwg/Desktop/md5.txt.
```

Figure 3.4 MD5 Hash Shell Code Used

3.6.2 DDMS procedures

The Dalvik Debug Monitor Server or DDMS is another Android SDK tool that was used to acquire files from an Android device. As the Android SDK was cross-platform on Mac and Windows, these trials were performed on the Mac system. DDMS was also a GUI based tool as opposed to ADB, and provided more visual feedback to the user. The procedures used were as follows. The tablet was connected to the workstation via the proprietary cable. Next, DDMS was executed by running `./ddms` from the `android-sdk-macosx/tools` directory. Upon running, DDMS presented a GUI to the user as shown in Figure 3.5.

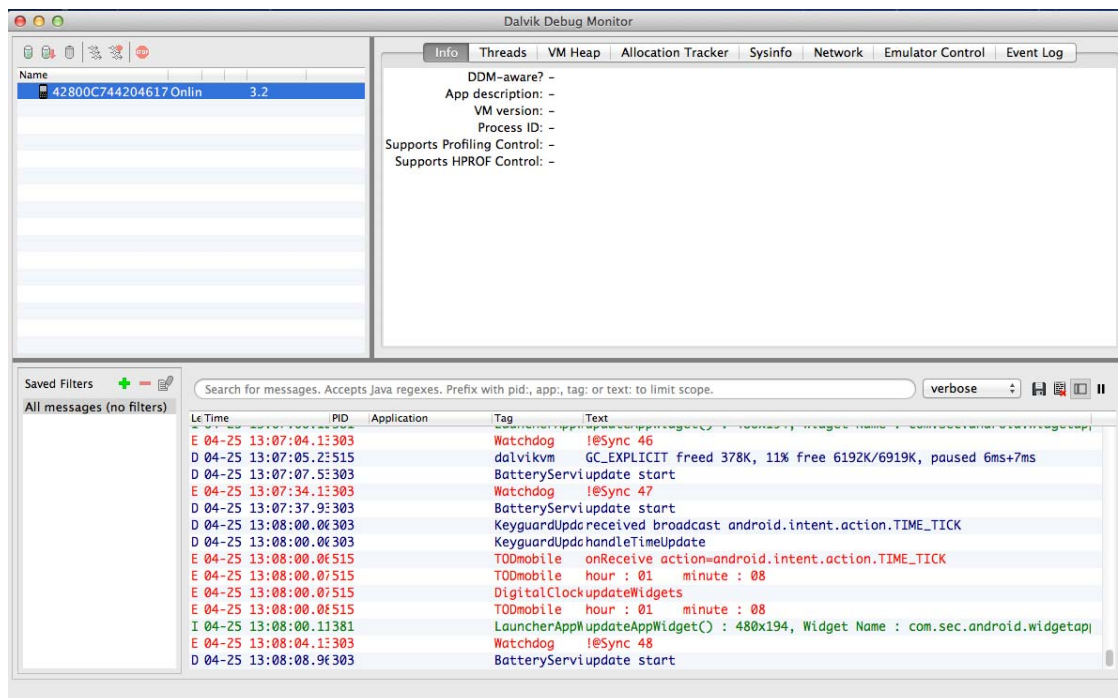


Figure 3.5 DDMS GUI.

Next, on the DDMS toolbar, file explorer was chosen as shown below in Figure 3.6.

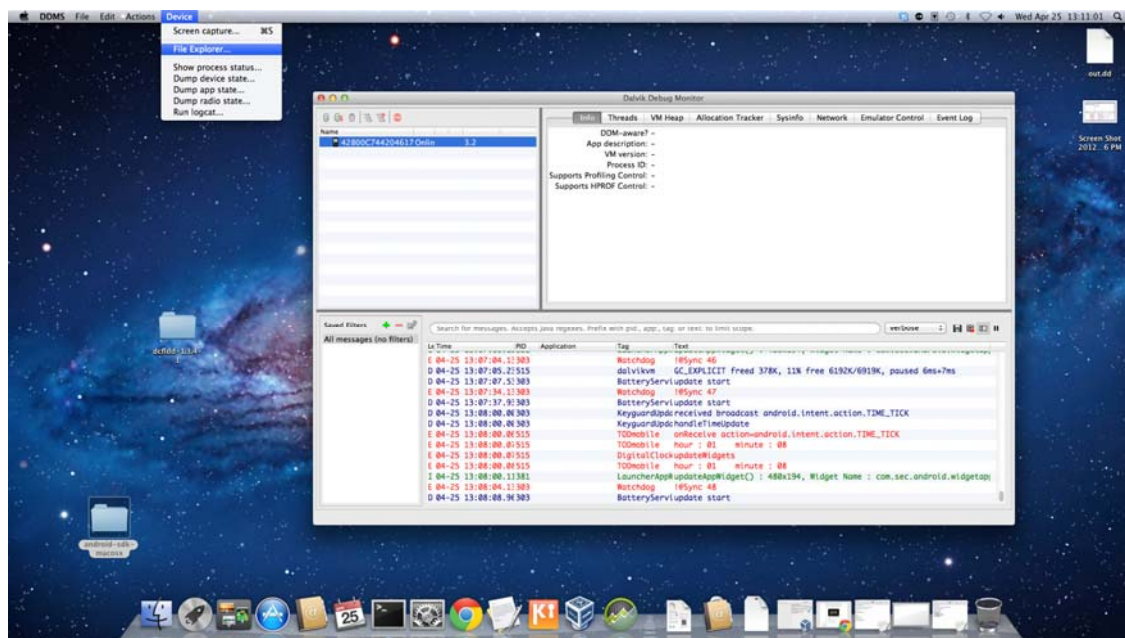


Figure 3.6: Choosing DDMS File Explorer.

This opened a dialog as shown below in Figure 3.7.

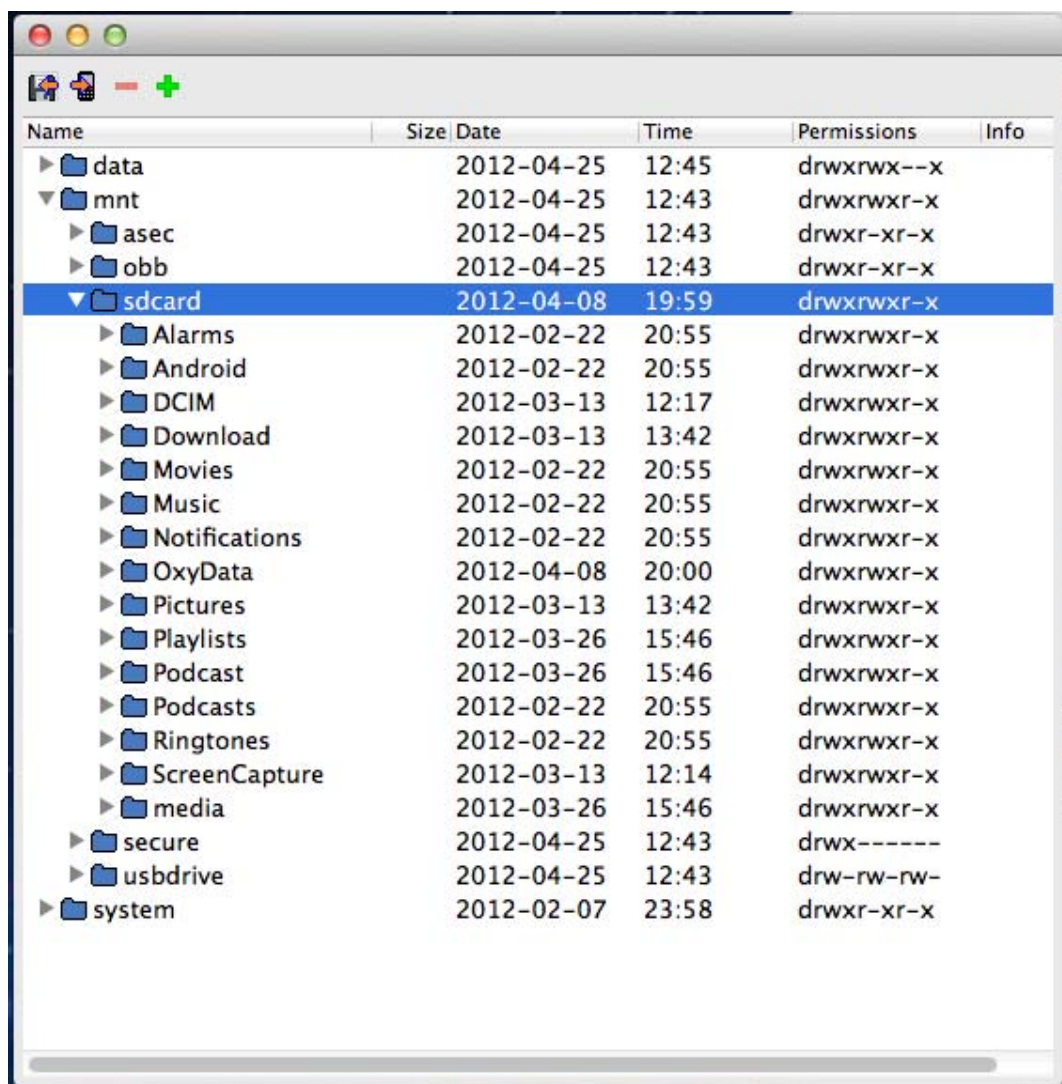


Figure 3.7: DDMS File Explorer Dialog

Once the sdcard directory was chosen, a recursive download of the desired structure was performed. All evidence files were contained at various locations in the sdcard file structure as this is where user data was stored by default.

3.6.3 Mobile Phone Examiner Plus Procedures

Mobile Phone Examiner Plus (MPE+) was a forensic suite for Windows from AccessData software. The exact version tested was 4.4.1.44195. While MPE+ was used to acquire files into an AD1 database format, the AccessData FTK imager tool 3.1 was used to read the AD1 file and produce a hash list.

The procedures used were as follows. First MPE+ was installed on the Windows 7 forensic workstation, along with the license dongle and license software. Next, the tablet, booted into its normal operating mode and containing the evidence corpus was logged into in order to enable USB debugging, which was required by MPE+. Upon starting the application, a choice was presented to acquire a mobile device or SIM. Mobile device was chosen as this tablet had no 3G capabilities.

Next, a device selection screen was displayed allowing the examiner to choose the manufacturer, model and port of the device. As shown in Figure 3.8, the appropriate values were entered.

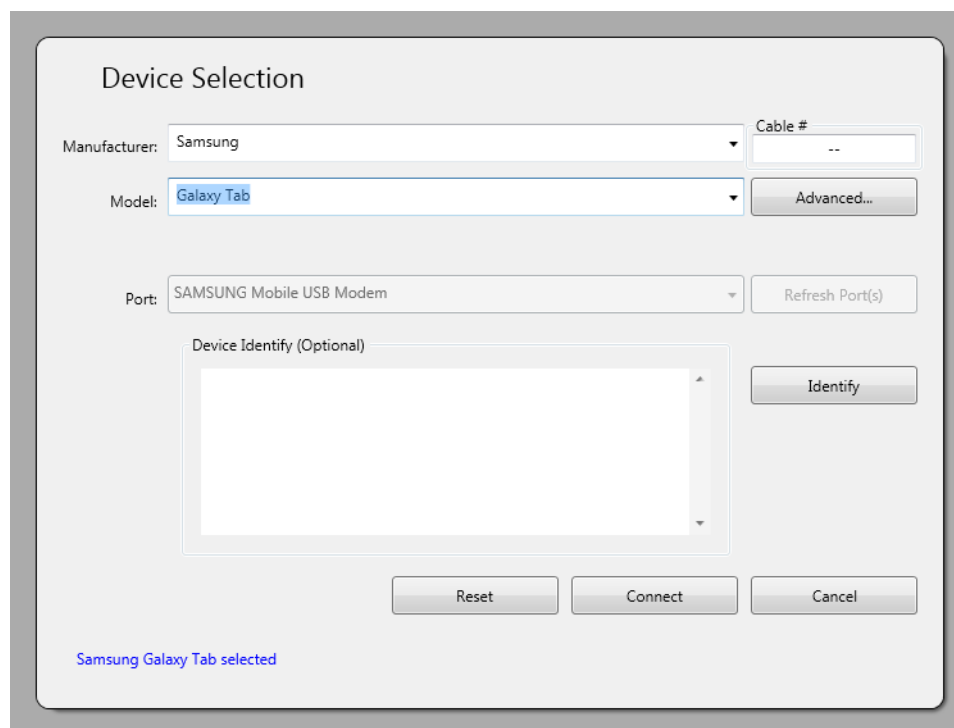


Figure 3.8: MPE+ Device Selection

Next, the application warned that a forensic flash card should be in the tablet and that USB debugging was enabled as shown in Figure 3.9. Only the USB debugging requirement was met as the tablet had no port to install a removable flash device.

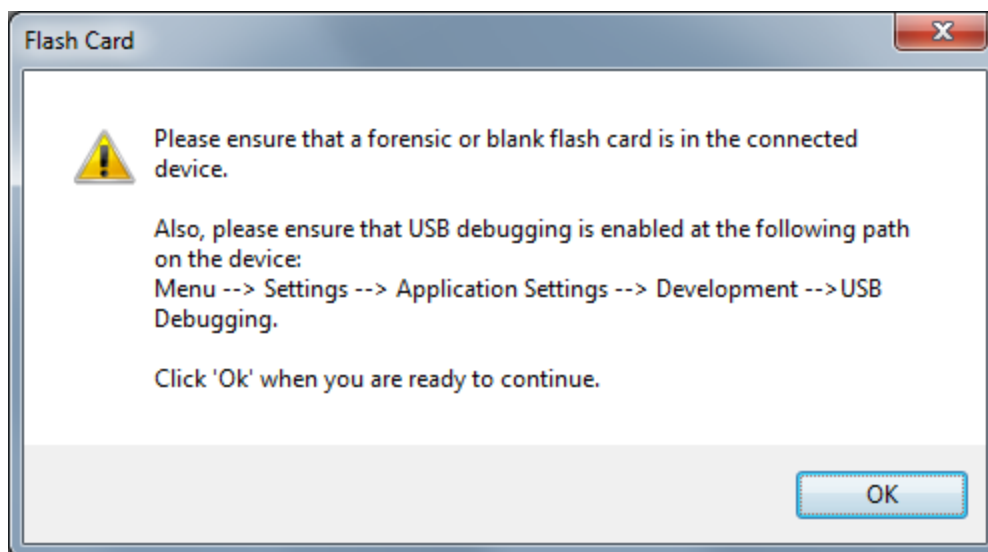


Figure 3.9: MPE+ Error Dialog

At this point, MPE+ connected to the device and prompted a selection of available data to extract including: phonebook, call history, calendar, SMS and file system. Only the “file system” option was chosen as shown in Figure 3.10.

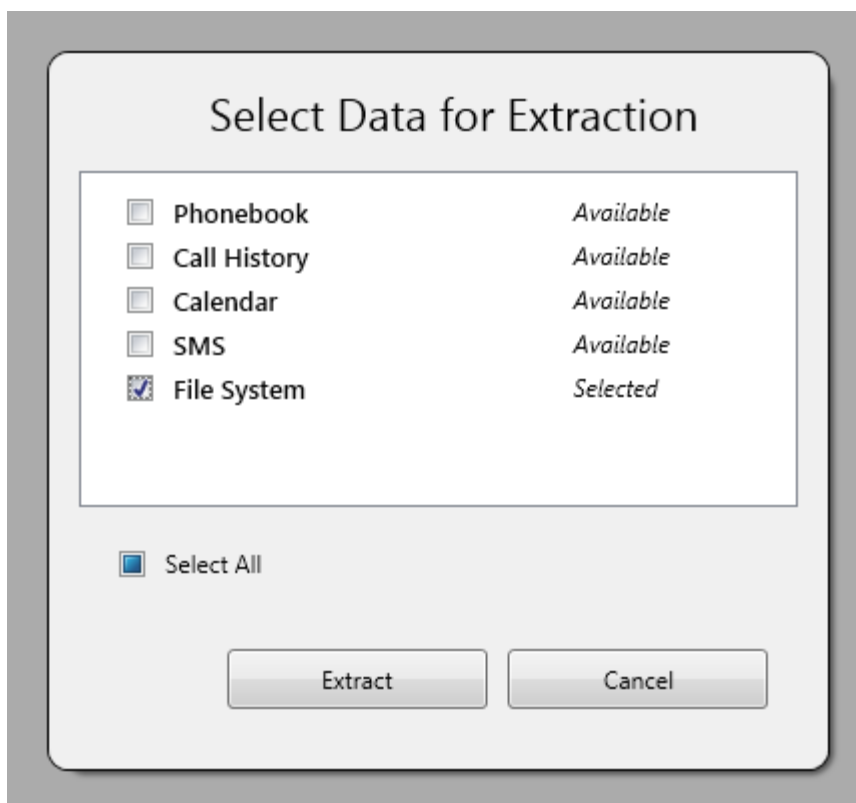


Figure 3.10: MPE+ File System Extraction Selection.

At this point the software extracted the files of the course of several hours, which was then saved in an AD1 database file format. Once this file was saved, FTK imager was used to import this image file, parse the files, and generate a hash listing. This procedure was performed 20 times.

3.6.4 Oxygen Procedures

Oxygen Forensic suite 2011 standard was also found to be compatible and was tested. Upon downloading the free demo and installing the software, the tablet was placed in a normal booted state and connected to the workstation with the proprietary cable. The

Oxygen Connection Wizard was used to connect to the device for file extraction. The first option was to select a connection type, of which “connect via cable” was chosen as shown in Figure 3.11.



Figure 3.11: Oxygen Device Connection.

Once the software found the tablet, it prompted to upload “OxyAgent” to the device. It was found that this agent was required for connectivity, so the option to install was chosen as shown in Figure 3.12.



Figure 3.12: Oxygen Wizard.

Upon uploading of the agent, a connection wizard was presented to connect to the successfully detected device as shown in Figure 3.13.



Figure 3.13: Oxygen Wizard with Device Information.

It was found that if USB debug was not enabled, the serial number would not populate and the connection would fail. Upon connection, the software presented an error that the device was considered a new phone, and was not supported, as shown in Figure 3.14.

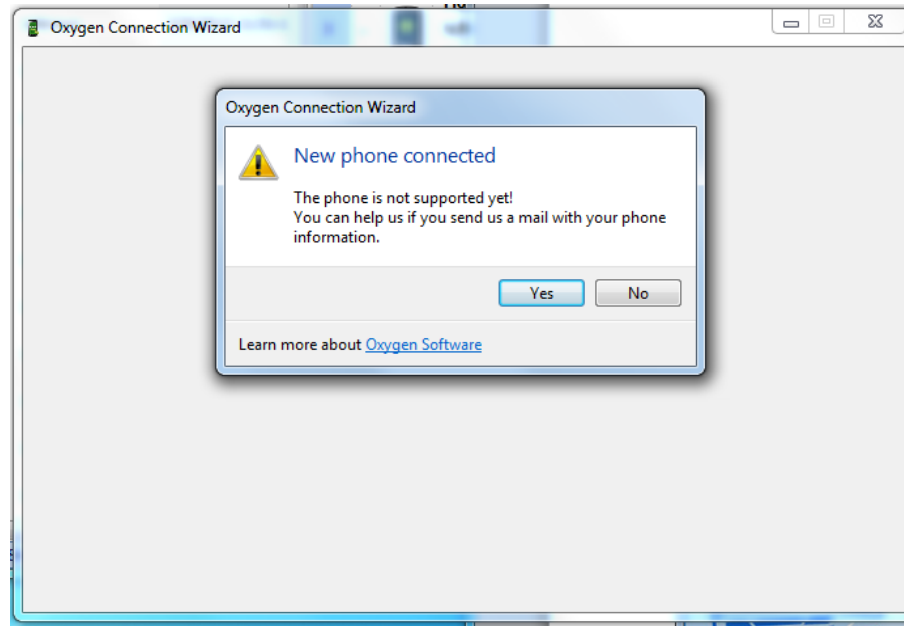
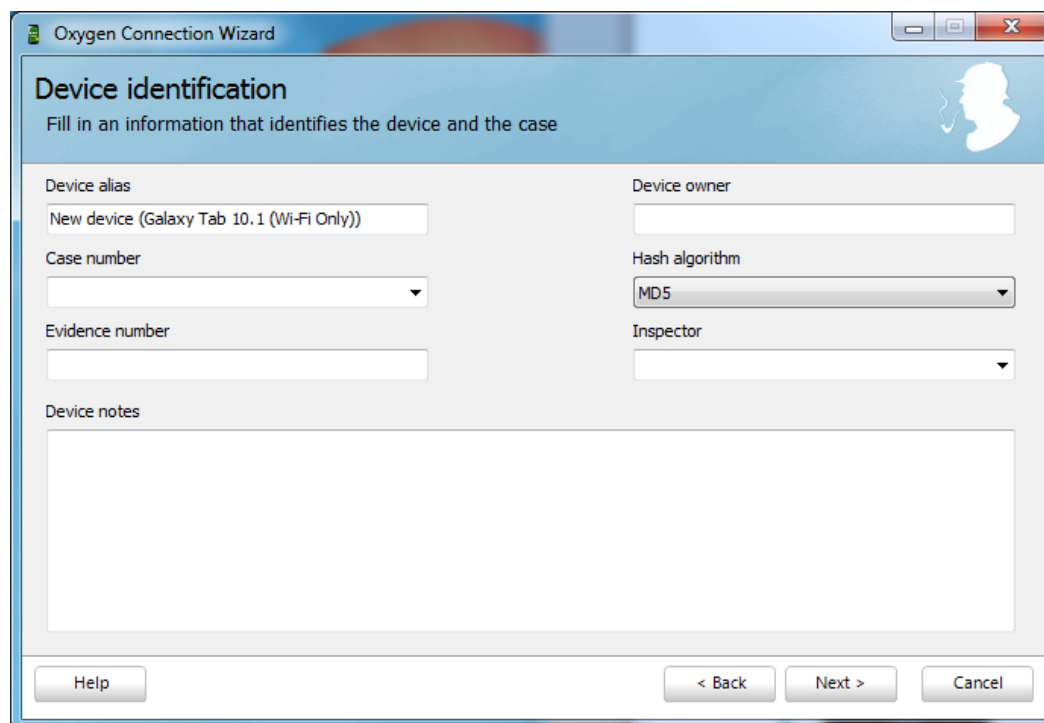


Figure 3.14: Oxygen Connection Wizard.

Yes was selected to continue and the data extraction wizard was presented. The Connection wizard then displayed the device information, and options dialog as shown in Figure 3.15.



The screenshot shows a window titled "Oxygen Connection Wizard" with a "Device identification" section. The subtitle reads "Fill in an information that identifies the device and the case". The form contains several fields: "Device alias" with the text "New device (Galaxy Tab 10.1 (Wi-Fi Only))", "Case number" with a dropdown arrow, "Evidence number" with a text input field, "Device owner" with a text input field, "Hash algorithm" with a dropdown menu showing "MD5", and "Inspector" with a dropdown arrow. There is also a "Device notes" section with a large text area. At the bottom, there are buttons for "Help", "< Back", "Next >", and "Cancel".

Figure 3.15: Oxygen Device Selection.

At this step, the hash algorithm was chosen and the trial number was documented in the case number field. Since this step dictated hashing, a total of 40 trials were performed with this tool to obtain 20 MD5 and 20 SHA1 hash sets. Once these selections were completed, the data to extract was chosen in the dialog shown in Figure 3.16.

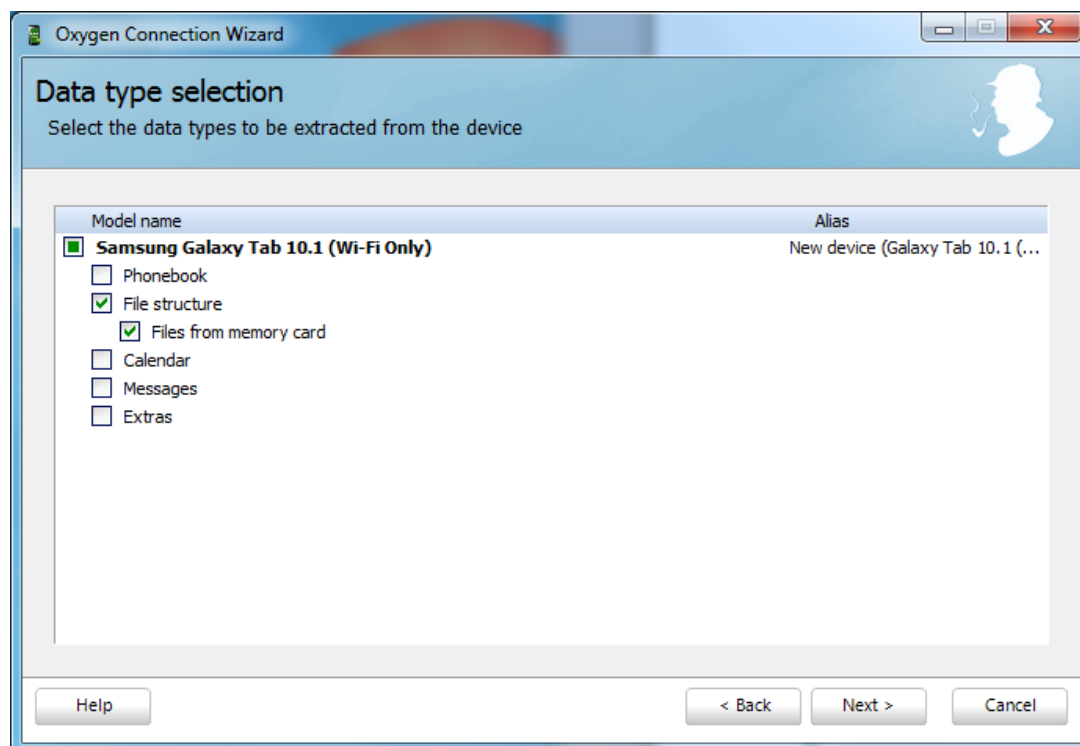


Figure 3.16: Oxygen Data Selection.

Once the selection to extract the file system was chosen, the extraction process proceeded as shown in Figure 3.17.

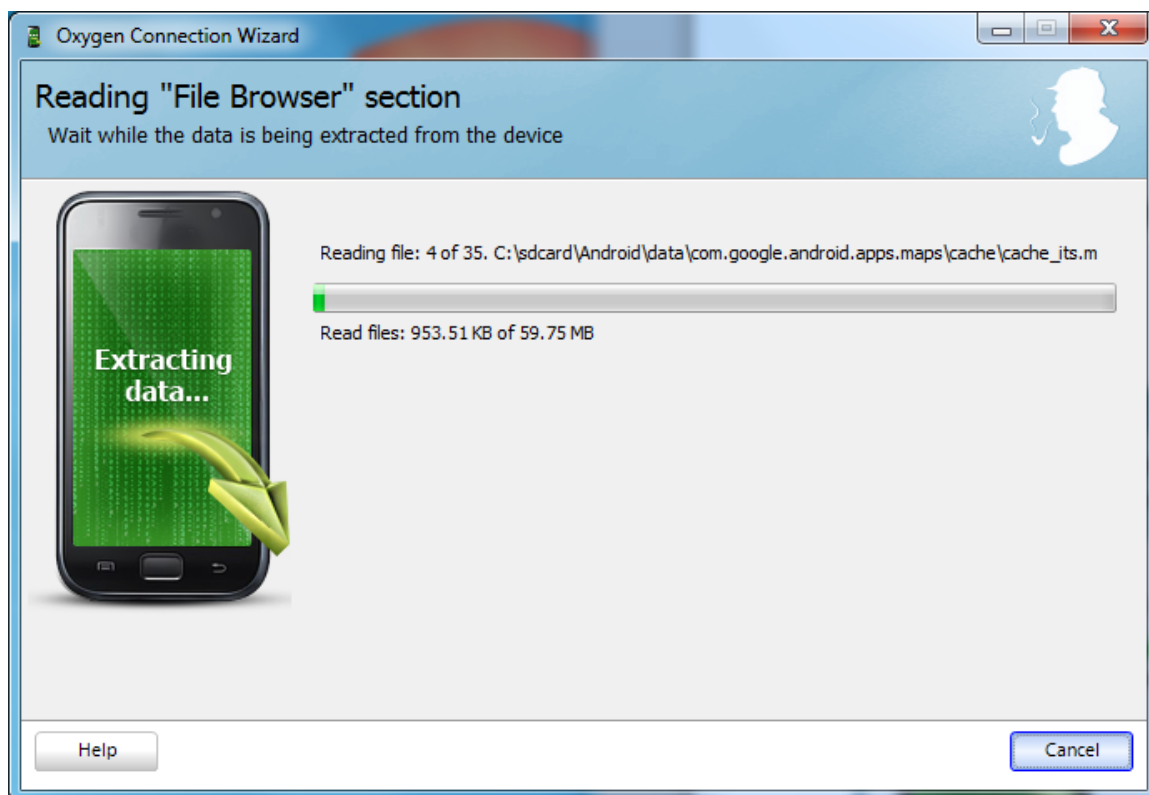


Figure 3.17: Oxygen Progress Bar.

After extraction was complete a dialog to choose a final action was displayed. In each case, the option to export and print was chosen in order to obtain a file hash list. In each trial an export to file option was chosen, and a comma separated values (csv) file was chosen and placed in the Dropbox location in the local file system. Once saved, the file was named appropriately as the software did not support this function. Oxygen then kept that trial in its internal database as a case.

3.6.5 Windows Native MTP over USB Procedures

To contrast special software tools, it was decided to perform trials based on standard USB connectivity. In this case it was found that the tablet supported only the media transport protocol (MTP) over USB on Windows 7. This meant that the device could not be mounted as a true file system and instead acted more like a digital camera or MP3 player, which also precluded the use of tools such as dd, dcfldd, or dc3dd.

First, the device was booted to a nominal state, then connected to the Windows 7 workstation via the cable. Since the workstation had the drivers for the Galaxy Tab installed, it would auto mount the device and display it as a media device instead of a lettered file system. For 20 trials, the contents of the device were copied using the Windows explorer GUI to the Dropbox repository. Similar procedures were tried on OS X and Linux, but it was found that neither OS available at the time natively supported MTP devices.

3.7 Limitations and Bias

Another factor that was considered was the serial nature of testing and the possibility that one test could influence subsequent tests. To control for this, the file hashes for each trial were verified, and the tablet was factory reset after every tool's 20-trial completion. Although this provided at least some control across tools, it could still represent a confounding variable within a set or tool trials and was considered when errors were encountered. Also, files were created both internal and external to the device to control for unknown potential differences. Table 3 illustrates where the evidence files were created.

Table 3.4. *Evidence Corpus Creation*

Tablet	Windows Forensic Workstation
20120313_121548.jpg	Doc2.docx
20120313_121616.jpg	Doc3.docx
20120313_121717.jpg	Doc4.docx
20120314_151245.mp4	Doc5.docx
20120314_151306.mp4	PDF2.pdf
20120314_151402.mp4	PDF3.pdf
20120314_151425.mp4	PDF4.pdf
20120314_151442.mp4	PDF5.pdf
SC20120313-121352.jpg	PDF1.pdf
SC20120313-121412.jpg	2012-03-12 14.14.00.jpg
	2012-03-12 14.14.35.jpg
	2012-03-12 14.14.59.jpg
	2012-03-12 14.16.06.jpg
	2012-03-13 12.04.27.jpg

Even though this research is limited in scope due to time and resource constraints, the methodology could be expanded over more tools and devices to provide a wider and more accurate view of the state of Android tablet imaging tools. It is expected that the methodology described will be repeatable by openly providing the exact steps used. Tools were selected based on references in literature, as well as compatibility testing on the forensic workstation as time permitted, again described in Chapter 4.

CHAPTER 4 FINDINGS

This chapter outlines the data collected, general findings, and the data for observed errors. The main finding with respect to the research question and hypothesis testing was that statistical analysis was not possible due to the limitations of the data collected.

4.1 Overview of Collected Data

Across all 20 trials per tool, 25 files were acquired and cryptographically hashed. A total of 50 hashes were generated, 2 per file, per trial, to provide a control against hashing algorithm implementation errors. In this study the data is comprised of the counts of errors determined from this set of hashes. A total of 500 files were transferred across trials per tool, generating 1000 hash values.

In the end, only one trial round of one tool, MPE+, encountered errors as defined as not acquiring a cryptographically hashed verified original file. All other trials resulted in zero error conditions, as verified by MD5 and SHA1 results of each file obtained. A table of error conditions observed is included in Appendix B. As only logical images were created, hashes and errors were evaluated on a per-file basis instead as a single entity as a physical image is usually hashed.

While this study set out to perform quantitative statistical analysis on the collected data, it was not possible to do so given that 4 out of 5 tools produced no measurable error rate, meaning that the mean error rate was zero which precluded statistical analysis.

4.2 Feature Error Rates

For the one tool that did encounter errors, MPE+, the following Feature Error Rates (FERs) (Baggili et al., 2007), defined as the proportion of failed acquisitions of files, were observed for the trials. In this case the FER is defined as the number of failed logical acquisitions, counted per file, divided by the number of trials. Overall, rates in this study were found to be zero, however one trial of MPE+ did result in error conditions. As each trial consisted of 25 files acquired, a total of 500 file acquisitions were performed per tool. As errors were observed in 25 file acquisitions, the sample mean FER for MPE+ was calculated as 25 total errors per 500 total files hashed, therefore indicating a sample FER of 25/500, or 0.05. The table below outlines the sum rates found per tool in this study,

Table 4.1. *Feature Error Rates Per Tool Tested*

	ADB	DDM S	MPE+	Oxygen	Windows MTP
Mean FER	0	0	0.05	0	0
Standard Deviation	n/a	n/a	~0.218	n/a	n/a

4.3 Description of Errors Encountered

After analysis of the second MPE+ trial, it was found that neither the MD5 nor the SHA1 hashes matched the original values. Additionally, the values did not match those hash values found in other trials for files in the corpus. In other words, the hashes for all files were the same for each file in the corpus among the same hashing algorithm, but differed across the two hashing algorithms used. Specifically, the SHA1 for all files acquired in trial two was found to be da39a3ee5e6b4b0d3255bfef95601890afd80709. Similarly, the MD5 value of all files repeated the value of d41d8cd98f00b204e9800998ecf8427e. As these repeating MD5 and SHA1 values were unique to this trial, it is possible this error is the result of some conditions specific to this trial and could likely represent an outlier in the data however statistical analysis to prove this was not possible. Determining the root cause of this repeating hash error was outside the scope of this study.

While the data collected was not appropriate for statistical analysis, a qualitative single-subject design, or case study was used instead. In other words, the observations of the researcher are used in place of statistical analysis of quantitative data. Based on this design it was observed that overall the tools did not error, given the same treatment, consisting of 20 trials on the same control evidence corpus. Only in one trial did MPE+ present error conditions, and based on an analysis of the data, all tools performed essentially the same with a low or nonexistent error rate.

4.4 Error Rates and Sample Size

It should be noted that as four of the five tools tested did not produce observed errors, which could indicate too small a sample size, or that sufficient quality assurance has produced very reliable tools. By increasing sample size in subsequent studies, it is hoped that a more comprehensive view of error rates can be obtained.

4.5 Android Tablet Acquisition Technologies

Specific to the sub-question of what technologies exist to image an Android tablet, it was found through a review of literature and research that numerous tools exist to acquire at least a logical image of an Android device, although most tools were focused on Android mobile phones and not tablets.

Numerous technologies existed to image an Android tablet; however the research found that during compatibility testing that not all technologies were compatible with the Galaxy Tab.

Table 4.2. List of Android Forensic Tools Discovered

Title	Platform
Android Debug Bridge	Windows, Linux, Mac
Dalvik Debug Monitor Service	Windows, Linux, Mac
Access Data Mobile Phone Examiner Plus	Windows
Oxygen Forensic Suite	Windows
Native Windows MTP Support	Windows
Android File Copy	Mac

Table 4.2 Continued

Title	Platform
gMTP	Linux
ViaForensics VIAExtract	Windows
Paraben Device Seizure	Windows
SD Card Swap	Android
Chip Removal	N/A – physical method
Dd, dcfldd, dc3dd	Linux
Backup Applications	Android
Nandroid	Android
Guidance EnCase	Windows
AccessData FTK	Windows

Based on a review of the literature, many technologies to acquire data from an Android device in a forensically sound manner were found. Indeed, several forensic suites, manufacturer tools, native Android tools and OS tools existed that offered at least some amount of support for acquisition of Android devices.

The thirteen tools were compatibility tested to determine a set of five to evaluate. Ubuntu Linux, Mac OS X Lion, VirtualBox appliance, and Windows 7 workstations were used for compatibility testing, depending on the support of the tool. For instance, commonly used forensic tools like dcfldd, dc3dd, and ADB are multi-platform, while other tools such as ViaForensics, Oxygen, MPE+, Paraben, and Android File Transfer were specific to one operating system.

In general, the types of technologies available to image a tablet device were broken into the following categories: mobile forensic acquisition tools, third party tools, mobile development support tools, manufacturer tools and OS features. Mobile Forensic acquisition tools are those that specifically claim to be for the purpose of forensically acquiring and analyzing a mobile device. Third party tools are tools that are not OS-native, but provide for some functionality and are commonly found referenced in digital forensic literature, including open source tools such as dd, dcfldd, dc3dd, gMTP, md5sum, sha1sum, etc. Mobile development tools are those that the mobile device OS maker may provide for testing and development of the device. The Android SDK and associated tools like ADB and DDMS fell into this category. Manufacturer tools include software, often proprietary, that provides some functionality but is often very device or device class specific. The following Table 11 lists the tool categories based on the results of compatibility testing into these categories.

Table 4.3. *Android Tablet Forensic Tool Categories*

Category	Names
Mobile Forensic	Paraben Device Seizure, ViaForensics, MPE+, FTK imager
Third Party	dd, dcfldd, dc3dd, gMTP, md5sum, sha1sum, Android File Transfer
Mobile Development	Android SDK, ADB, DDMS
Manufacturer	Samsung Kies
Operating System Feature	MTP support over USB

4.6 Differences in Presentation of Acquisition Results

Addressing the sub-question which asked “How do the technologies differ in their presentation of results?” the research found that the outputs of the tools tested varied dramatically. For instance while ADB, DDMS, and MTP copied files locally to a folder, MPE+ and Oxygen created a proprietary data file with the contents and metadata. Additionally, the forensic tools, MPE+ and Oxygen offered reporting functionality to report hash values, filenames, file sizes, and so on. Table 10 illustrates the information presented natively by each tool tested to the examiner.

Table 4.4. *List of Reporting Elements Per Tool Tested*

Tool	Imaging Report Elements
ADB	File listing in command line output
DDMS	GUI list of files/directories. Log window with status.
Oxygen	Ability to export report file containing at least: device, alias, manufacturer, model, serial, software revision, IMSI, ICCID, owner information, rooted status, software version, case number, extraction times and dates, extracted file path, file names, file sizes, file creation dates and file cryptographic hashes.
MPE+ and FTK imager	File hashes, file names, file path on mobile device
Windows MTP	Windows GUI representation of directories

As the table above shows, with respect to the reporting elements of the imaging report elements, the outputs varied in the presentation of results. Although the imaging portion of a tool may not have included reporting functionality, this functionality could have been included in a different portion of a suite or in a separate tool.

CHAPTER 5 CONCLUSIONS, DISCUSSION, AND RECOMMENDATIONS

5.1 Discussion

Even in the absence of statistical analysis, the observed data at least partially supported the alternative hypothesis. While 4 of the 5 tools did not error, one tool was observed to error indicating at least some error rate. Additionally, some common features were described among the tools, while there were numerous differences in how tools presented data to the user.

Computers by their nature are designed to copy data reliably and accurately, and software is often written to control for errors using several techniques (Lin, 1983). Similarly, forensic logical acquisition tools are designed with intent to be reliable in acquiring data. Based on these properties, a low error rate was expected. Although errors in Android tablet imaging may occur through user error, improper implementation of a feature in code, or an incompatibility with a device, the data gained from the research suggest that when errors occur all acquired data is unverifiable. Partly this is because hashing was used as the criterion for errors and verification, but also because the observed errors affected all data in a trial. Specifically, when errors were encountered during the MPE+ trial, all file hashes for both MD5 and SHA1 were affected meaning that the logical image could not be verified, resulting in an error. The error, taken by itself was not determined to be caused through the software or the user.

The research performed implies that at least a minimal error rate may exist for a tablet forensic acquisition tool, but also that the notion of an error rate may require further definition. However, as error rates were not found to be measured, known, or reported in literature then lack of these measured error rates may cause an incorrect assumption to be made that forensic tools do not error. In any case even the limited data collected suggests that any implication of error-free operation was rightfully questioned.

5.1.1 Summary of Relative Error Rates

With respect to the primary research question, it was observed that for four of five tested technologies, no error rate, based on the Feature Error Rate, or FER (Baggili et al., 2007). Based on the data, it appeared that overall the technologies tested had low error rates, if any existed at all.

These findings implied that existing Android tablet acquisition technologies may be reliable enough to error infrequently or at least at a rate low enough to be undetectable. However, the findings also show that any assertion that errors did not exist at all was a false one. These findings show that researchers and practitioners need to be cognizant that error rates do exist, that they are measurable, and that any work relying on assumptions of zero error rates may be unreliable. Overall this data represented the first known error rates with respect to Android tablet acquisition.

5.1.2 Discussion of Hashing Issues

With respect to hashing as a measure for errors, a question arose during the research. If a tool reports obviously incorrect hash values, in this case a repeating value, should the files simply be re-hashed with the same tool, or hashed with another tool? The

safest choice was to verify the output by hashing with a different tool. However, this task often proved impossible based on how each tool differed in collecting data. For instance some tools saved acquired files in a database, while others copied files to a directory structure. This was a core problem that may require a new methodology to assure veracity in hashing operations, especially when tools acquire the raw files to a database or otherwise proprietary format.

5.1.3 Lack of Physical Imaging

As discussed in Chapter 4, no forensic imaging tool evaluated for compatibility or used in the study supported a physical image of the Galaxy Tab. In the context of this research this was likely due to the lack of USB mass storage support on the device, meaning the tablet was not designed to expose its file system to a user. Mass storage support would expose this file system for mounting which a forensic imaging tool such as dd could read and image. Instead, the MTP protocol was used to provide file transfer functionality on the Galaxy Tab. While MTP was designed for transferring files, it was not adequate in a traditional forensic sense where all bits of a medium must be copied as in a physical image, or for imaging technologies that mount a file system.

5.1.4 Meaningfulness of Error Rates

Due to the relative absence of errors found in the data, one could call into question the meaningfulness of testing for error rates for Android tablet forensic imaging software. The software tested could have been sufficiently quality tested to be bug free, but there were numerous variables including device compatibility such as drivers, and file system presentation. Additionally, human error while not explicitly studied in the

research, could have been a factor in contributing to Allchin's (2001) material, or improper implementation, errors. However, in order to control for potential errors introduced by improper procedures, the same methodology was used. Determining error rates is important for admissible scientific evidence and for research.

5.2 Conclusions

Restated, the research described in this thesis was guided by the following primary research question: What are the relative success and failure rates among selected Android tablet evidence acquisition technologies? Additional sub-questions that further framed this research included: What technologies exist to acquire a logical or physical image? How do the technologies differ in their presentation of results? What are the success/failure and error rates with respect to the imaging tools tested?

In response to these research questions, it was found that 4 of the 5 Android tablet imaging technologies did not encounter any errors. Additionally, a list of acquisition technologies was developed as presented in Table 9, and a comparison of presentations was completed and presented in Table 10.

5.3 Limitations

In addition to the pre-stated constraints, additional limitations were encountered during the research. First, the fact that none of the tools found supported physical imaging meant that this feature was not able to be tested and that only logical imaging was assessed. Second was the wide variation in the amount of time required to perform data collection. Although three weeks were budgeted for data collection from 5 tools, it

was found that two additional weeks were required to perform analysis. This limitation highlighted the recommendation that tool testing be performed in as automated fashion as possible. Another limitation found was that only Windows-compatible tools were found to be usable with the Galaxy Tab tested. Not only this, but of the Windows-only tools, only two mobile forensic specific tools could be tested. This limitation was also due to time constraints in negotiating resources from vendors in order to evaluate full-version products instead of demo products.

Perhaps the limitation most impacting this research was the fact that quantitative assessment of the data could not be performed. While the data cannot be improved for this study, it may still be possible for future studies to gather a larger sample. This constraint did limit any possible quantitative hypothesis testing and therefore that question remains unanswered statistically.

5.4 Significance

While the data obtained in this study precluded a quantitative analysis of the Android imaging tools tested, a single-subject method was used to determine that errors did in fact present in the data. This finding was significant for researchers and practitioners of Android tablet forensic acquisitions as it shows that errors exist and are measurable. Secondly, this research represents the first attempt at determining an error rate for Android tablet imaging tools. Error rates, and specifically known error rates are a standard for admission of scientific evidence ("*Daubert v. Merrell Dow Pharmaceuticals*," 1993) as well as a way to allow researchers and practitioners to better understand the tools used in Android tablet forensics. It was hoped that an improved

methodology and larger sample could be used to further understand this problem. Finally the methodology described should be straightforward enough to replicate across additional tools and devices.

5.5 Recommendations

As this study found that a sample of 20 trials did not produce errors in four of the five tools tested, this suggests that a larger sample size may be required to study Android imaging error rates. The following section will discuss recommendations gathered from the observed data and performing the research.

5.5.1 Common Root Cause Investigation

As the observed errors were found to be generated during all file acquisitions in a trial, this could imply that in addition to a measurable rate a root cause of the error should exist and be quantifiable. Discovering the root cause of errors in acquisition operations would likely be a software engineering task that further studies could undertake, perhaps with forensic researchers collaborating with software engineering researchers. A recommendation of this research is that further study into the conditions that can cause errors to occur be performed in order to if they are correlated with certain causes. This approach was performed by Nakajo using Fault Tree Analysis (1991), even though the proprietary nature of some software may prevent thorough analysis of source code in some instances.

5.5.2 Dealing With Software Updates

An issue facing the research was the relative ease and frequency of software updates to the tools, or even updates to the operating system used in the study. While some tools remained static, others underwent version changes during the course of research. Additionally, continual operating system updates could potentially introduce errors by altering the code which comprises each tool. This problem was not specifically researched in the review of literature, so it was not determined how any prior approaches have dealt with this issue. Complete automation of tool testing may be one way of providing timely error rate measurements in the face of regular software updates.

5.5.3 Automated Error Testing

Based on limitations discovered during this research, a recommendation that testing should be performed in as automated fashion was recommended. In other words, based on the difficulties in manually performing trials on GUI based tools, automation was found to be the best response to this problem. Of the tools tested, the number of trials completed in a given amount of time was far lower with command-line scriptable tools than GUI-only tools. An standard for forensic software to include an interface to allow for scriptable, automated testing was recommended.

5.6 Summary

In closing, while the research was not able to produce statistical measures for all tools tested, errors were observed. It is hoped that the methodology presented can provide

useful guidance for further research into the area of Android tablet forensic acquisition.

While numerous file copying tools exist for Android tablets, not every tool was equally compatible with the device tested. By performing testing based on specific features, such as logical imaging, examiners may be able to build a base upon which error rates can be known from a scientific perspective.

LIST OF REFERENCES

LIST OF REFERENCES

- Ahmed, R., & Dharaskar, R. (2008). *Mobile forensics: an overview, tools, future trends and challenges from law enforcement perspective*. Paper presented at the International Conference on E-Governance.
- Allchin, D. (2001). Error Types. *Perspectives on Science*, 9(1), 38-58.
- Android Developers. (2011a). Android debug bridge. Retrieved November, 12, 2011, from <http://developer.android.com/guide/developing/tools/adb.html>
- Android Developers. (2011b). Platform Versions. Retrieved November, 9, 2011, from <http://developer.android.com/resources/dashboard/platform-versions.html>
- Apache Foundation. (2011). The apache software foundation frequently asked questions. Retrieved January, 2, 2012, from <http://www.apache.org/foundation/faq.html> - what
- Association of Chief Police Officers. (2007). *Good practice guide for computer-based electronic evidence*. Retrieved from http://www.7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence_v4_web.pdf.
- Ayers, R., Jansen, W., Moenner, L., & Delaitre, A. (2007). *Cell phone forensic tools: An overview and analysis update*. Gaithersburg, MD: National Institute of Standards and Technology.
- Baggili, I. M., Mislan, R., & Rogers, M. (2007). Mobile phone forensics tool testing: A database driven approach. *International Journal of Digital Evidence*, 6(2), 1-11.
- Beebe, N. (2009). Digital forensic research: The good, the bad and the unaddressed. In G. Peterson & S. Shenol (Eds.), *Advances in Digital Forensics V, IFIP AICT* (Vol. 306, pp. 17-36). Boston: Springer.
- Benedict, N. (2004). Fingerprints and the daubert standard for admission of scientific evidence: Why fingerprints fail and a proposed remedy. *Arizona Law Review*, 46, 519-549.
- Cardwell, G. S. (2011). *Residual network data structures in android devices*. Naval Postgraduate School, Monterey, CA.

- Casey, E. (2004). *Digital evidence and computer crime* (2nd ed.). San Diego, CA: Elsevier.
- Chen, J. (2008). *An introduction to android*. Paper presented at the Google I/O, San Francisco, CA.
- comScore. (2011). Comscore reports september 2011 u.s. mobile subscriber market share. Retrieved November, 7, 2011, from http://www.comscore.com/Press_Events/Press_Releases/2011/11/comScore_Reports_September_2011_U.S._Mobile_Subscriber_Market_Share
- Crothers, B. (2011, October, 23, 2011). In tablets, android's star is rising but... Retrieved November, 9, 2011, from http://news.cnet.com/8301-13924_3-20124401-64/in-tablets-androids-star-is-rising-but.../
- CTIA. (2011). Wireless quick facts. Retrieved November, 1, 2011, from <http://www.ctia.org/advocacy/research/index.cfm/aid/10323>
- Danker, S., Ayers, R., & Mislán, R. (2009). Hashing techniques for mobile device forensics. *Small Scale Digital Device Forensics Journal*, 3(1), 1-6.
- Daubert v. Merrell Dow Pharmaceuticals* (United States Supreme Court 1993).
- Dictionary.com. (Ed.) (2012).
- Distefano, A., Gianluigi, M., & Pace, F. (2010). Android anti-forensics through a local paradigm. *Digital Investigation*, 7, S83-S94.
- Elgin, B. (2005). Google buys android for its mobile arsenal. *Bloomberg Businessweek*. Retrieved from http://www.businessweek.com/technology/content/aug2005/tc20050817_0949_tc024.htm
- Epstein, D. M., Tebbett, I. R., & Boyd, S. E. (2003). Eliminating sources of pipetting error in the forensic laboratory. *Forensic Science Communications*, 5(4), 1-6.
- Feenberg, A. (2006). *What is philosophy of technology?* New York, NY: Palgrave-MacMillan.
- France, J. (2009). New archos 5 internet media tablet does it all. Retrieved November, 9, 2011, from http://reviews.cnet.com/8301-12519_7-10353176-49.html
- Gartner. (2011). Gartner says android to command nearly half of worldwide smartphone operating system market by year-end 2012. Retrieved November, 28, 2011, from <http://www.gartner.com/it/page.jsp?id=1622614>
- Godwin-Jones, R. (2008). Emerging technologies mobile-computing trends: Lighter, faster, smarter. *Language Learning & Technology*, 12(3), 3-9.

- Google. (2012). Android open source project philosophy and goals. Retrieved January, 2, 2012, from <http://source.android.com/about/philosophy.html>
- Hendricks, R. (2008). Admissibility of small scale digital devices in u.s. civil litigation. *Small Scale Digital Device Forensics Journal*, 2(1), 1-4.
- Hoog, A. (2011). *Android forensics: investigation, analysis and mobile security for google android*. Waltham, MA: Syngress.
- International Telecommunications Union. (2011, June, 29, 2011). ICT statistics newslog - 8% of u.s. adults own tablet devices. Retrieved November, 7, 2011, from <http://www.itu.int/ITU-D/ict/newslog/8+Of+US+Adults+Own+Tablet+Devices.aspx>
- Lessard, J., & Kessler, G. C. (2010). Android forensics: simplifying cell phone examinations. *Small Scale Digital Device Forensics Journal*, 4(1).
- Lin, S. C., D. J. (1983). *Error control coding: fundamentals and applications*. Englewood Cliffs, N.J.: Prentice-Hall.
- Lyle, J. (2010). If error rate is such a simple concept, why don't I have one for my forensic tool yet? *Digital Investigation*, 7, S135-S139.
- Manning, C. (2002). YAFFS: the NAND-specific flash file system introductory article. Retrieved January, 2, 2012, from <http://www.yaffs.net/yaffs-nand-specific-flash-file-system-introductory-article>
- Me, G., & Rossi, M. (2008). *Internal forensic acquisition for mobile equipments*. Paper presented at the IEEE International Symposium on Parallel and Distributed Processing.
- Merriam-Webster. (2012). from <http://www.merriam-webster.com/dictionary/smartphone>
- Nakajo, T. K., H. (1991). A case history analysis of software error cause-effect relationships. *Software Engineering, IEEE Transactions*, 17(8), 830-838.
- National Academy of Sciences. (2009). *Strengthening forensic science in the united states: A path forward*. Washington, D.C.: The National Academies Press.
- National Institute of Justice. (2010). The computer forensics tool testing program. Retrieved November, 10, 2011, from <http://www.nij.gov/nij/topics/forensics/evidence/digital/standards/cfft.htm>
- National Institute of Standards and Technology. (2003). CFTT methodology overview. Retrieved December, 10, 2011, from http://www.cfft.nist.gov/Methodology_Overview.htm

- Open Handset Alliance. (2007). Industry Leaders Announce Open Platform for Mobile Devices. Retrieved November, 1, 2011, from http://www.openhandsetalliance.com/press_110507.html
- Owen, P., & Thomas, P. (2011). An analysis of digital forensic examinations: Mobile devices versus hard disk drives utilising ACPO & NIST guidelines. *Digital Investigation*.
- Pan, L., & Batten, L. M. (2009). Robust correctness testing for digital forensic tools. In M. Sorell (Ed.), *Forensics in telecommunications, information and multimedia* (Vol. 8, pp. 45-64). Berlin, Germany: Springer.
- Ridder, C. K. (2007). *Evidentiary implicaitons of potential security weaknesses in forensic software*. Paper presented at the DEFCON 15.
- Saks, M. J., & Koehler, J. J. (2005). The coming paradigm shift in forensic identificaiton science. *Science*, 309, 892-895.
- Samsung Electronics Co., L. (2006). *Introduction to samsung's linux flash file system - rfs*. Hwasung-City, Korea.
- Samsung Electronics Co., L. (2008). *Linux rfs power off recovery: technical paper version 1.0*. Hwasung-City.
- Schmitt, S., Spreitzenbarth, M., & Zimmermann, C. (2011). *Reverse engineering of the android file system yaffs2*. Erlangen, Germany: Univeristy of Erlangen-Nuremberg.
- SD Association. (2012). SD Standards Family. Retrieved January, 2, 2012, from <https://http://www.sdcard.org/developers/overview/family>
- Shabtai, A., Fledel, Y., Kanonov, U., Dolev, S., & Glezer, C. (2010). Google android: A comprehensive security assessment. *IEEE Security & Privacy*, 8(2), 35-44.
- Thing, V., Ng, K., & Chang, E. (2010). Live memory forensics of mobile phones. *Digital Investigation*, 7, S74-S82.
- USB Implementers Forum, I. (2007). USB media transfer protocol specification.
- Vidas, T., Zhang, C., & Christin, N. (2011). Toward a general collection methodology for android devices. *Digital Investigation*, 8, S14-S24.
- Willassen, S. (2005). Forensic analysis of mobile phone internal memory. In M. Pollitt & S. Shenoi (Eds.), *Advances in Digital Forensics* (Vol. 194, pp. 191-204). Boston, MA: Springer.

Yates, I., M. (2010). *Practical investigations of digital forensics tools for mobile devices*. Paper presented at the Information Security Curriculum Development Conference, Kennesaw, GA.

APPENDIX

Table A-1.1 Evidence Corpus Master List

File Type - created	Filename	MD5 SHA1	Tablet Location
JPEG - external	2012-03-12 14.14.00	39323bcf817f6016b62573f983f736b7 7169de021a42e3ba667bd96a563ecf3b8f569b80	\mnt\sdcard\Pictures\external images
JPEG - external	2012-03-12 14.14.35	59e74a1900733ff745f8a8f26dadfd3e 95f200160e3c9a66d7e16808b161d6b5b3ecdc37	\mnt\sdcard\Pictures\external images
JPEG - external	2012-03-12 14.14.59	92fe3848d6fa64604394899a446bcfd2 3694c025a76f3d0feb81a2310f6139aa7d54b8cd	\mnt\sdcard\Pictures\external images
JPEG - external	2012-03-12 14.16.06	93c7becbaa3d267a7975af11a70ae8be 4e271041e2063e2bf8d71cc5fd6902440ab7d41f	\mnt\sdcard\Pictures\external images
JPEG - external	2012-03-13 12.04.27	70cd36f2ea4ba7e19193599afc7d2f00 033940894b9cc209472b31c5ae62fa04c6925aed	\mnt\sdcard\Pictures\external images
JPEG - internal screen capture	SC20120313-121352	338bce9e43d4272edf72f0ca909ecab8 eda8f5f0a84b943e7ac7e1466058f40149ba5341	\mnt\sdcard\ScreenCapture
JPEG - internal screen capture	SC20120313-121412	17b97e45671277032112066ed6844e22 fffad508a3e90802a37ac9191be46c0bef6e769e	\mnt\sdcard\ScreenCapture
JPEG - internal camera	20120313_121548	9956d0e06e0bca0f126bd3478511d42f e4ec9567e2adb96a6a6639353688b000ba611ab4	\mnt\sdcard\DCIM\Camera

Table A-1.2 Evidence Corpus Master List

JPEG - internal camera	20120313_121616	e49bbce1d5b2477be97284a810f003a4 f37404665c9d8b0d39c7ccb67377b17e5107cf59	\mnt\sdcard\DCIM\Camera
JPEG - internal camera	20120313_121717	bf7a9ffc9fdd7067474d7f4895f75079 5bad672b2fb7dcfac86ca4980819626a9e6774e2	\mnt\mnt\sdcard\DCIM\Camera
PDF - external	PDF1	997a622582e5251c9bb92cf6ace7b60a d1f66972b35cc562628fc0d3528c450b85b73e76	mnt\sdcard\Download\PDFs
PDF - external	PDF2	b2470f2cd060b847f5f4f04cbe0a839e c76f438f08d4bf3010268ec90ab42f2e2b702bc7	\mnt\sdcard\Download\PDFs
PDF - external	PDF3	21f0a6b70e849edd85db30dfc04aca52 2e0e42afc10426acefa36200991abd0dbccffba64	\mnt\sdcard\Download\PDFs
PDF - external	PDF4	3ed9a373366d58bd023a2bb23fbebc21 d86d3caca2ab519df1a8270229e82441067dd99b	\mnt\sdcard\Download\PDFs
PDF - external	PDF5	ac6b7f196c5d18c035da6491124fe7bc c6c83ee66d95b42f9cd9657c954f48bd6ee6c843	\mnt\sdcard\Download\PDFs
DOCX - external	Doc1	65b982f0b4d8d1d6b3c06c0e214ebe04 6487503fbd5d56ad25a7cbc18724f7b51626b193	\mnt\sdcard\Download\Docs
DOCX - external	Doc2	f965c9ed917d9b3d5565f2bb3e291ded 0a4d72161d2466e94f056dc8f269c7b363bd40ef	\mnt\sdcard\Download\Docs
DOCX - external	Doc3	07fb6eb2a914653479fbb41ffdf11475 eaed2373852515e69bdf1722812afb52b28e3f87	\mnt\sdcard\Download\Docs
DOCX - external	Doc4	de6d681cb34264bf36a9a0ee8b911112 e9f22738127d1dd9bcbc9ed04897b45d2d2badee	\mnt\sdcard\Download\Docs

Table A-1.3 Evidence Corpus Master List

DOCX - external	Doc5	d6bc854f3d8957667d3cf28a11bda674 2bfb289d5c6c9654e777600dbeba0c5dd4d0cd4b	\mnt\sdcard\Download\Docs
mp4 - internal camera	20120314_151245	b87cca36288ce1e5bad291f55a0ba891 4eb7cdbcbcd5e1662bdecc9ee993a54176b1be75c	\mnt\sdcard\DCIM\Camera
mp4 - internal camera	20120314_151306	7e823aa65319ad24612376de8ab56c6d 0188d4a4004ecafaaa9ae90990a6ea067f07acf8	\mnt\sdcard\DCIM\Camera
mp4 - internal camera	20120314_151402	0e44b70e516ef64486d48fae508dec72 75d859ce5fa70d4fb40f085762fb7bead0d80aa8	\mnt\sdcard\DCIM\Camera
mp4 - internal camera	20120314_151425	47bb216f3f670fa7b09fbadc11f74016 e0f9e2ea9b3b10fd67688d90c406d02ad67f2759	\mnt\sdcard\DCIM\Camera
mp4 - internal camera	20120314_151442	2824136ca94ace7a803929d86c6fd573 8e7ff57a0bc030efc2e9366b5373475acbb38fec	\mnt\sdcard\DCIM\Camera

Table B-1.1 Table of Error Conditions Per Trial

	ADB	MPE+	DDMS	Oxygen	MTP over USB
Trial 1	0	0	0	0	0
Trial 2	0	25	0	0	0
Trial 3	0	0	0	0	0
Trial 4	0	0	0	0	0
Trial 5	0	0	0	0	0
Trial 6	0	0	0	0	0
Trial 7	0	0	0	0	0
Trial 8	0	0	0	0	0
Trial 9	0	0	0	0	0
Trial 10	0	0	0	0	0
Trial 11	0	0	0	0	0
Trial 12	0	0	0	0	0
Trial 13	0	0	0	0	0
Trial 14	0	0	0	0	0
Trial 15	0	0	0	0	0
Trial 16	0	0	0	0	0
Trial 17	0	0	0	0	0
Trial 18	0	0	0	0	0
Trial 19	0	0	0	0	0
Trial 20	0	0	0	0	0