A critical review of 7 years of Mobile Device Forensics

Konstantia Barmpatsalou[a], Dimitrios Damopoulos[a], Georgios Kambourakis[a,*], Vasilios Katos[b],

[a]*Info-Sec-Lab Laboratory of Information and Communications Systems Security, Department of Information and Communication Systems Engineering, University of the Aegean, Samos, Greece*
[b]*Information Security and Incident Response Unit, Department of Electrical and Computer Engineering, Democritus University of Thrace, University Campus, Kimmeria, Xanthi,Greece*

Abstract

Mobile Device Forensics (MF) is an interdisciplinary field consisting of techniques applied to a wide range of computing devices, including smartphones and satellite navigation systems. Over the last few years, a significant amount of research has been conducted, concerning various mobile device platforms, data acquisition schemes, and information extraction methods. This work provides a comprehensive overview of the field, by presenting a detailed assessment of the actions and methodologies taken throughout the last seven years. A multilevel chronological categorization of the most significant studies is given in order to provide a quick but complete way of observing the trends within the field. This categorization chart also serves as an analytic progress report, with regards to the evolution of MF. Moreover, since standardization efforts in this area are still in their infancy, this synopsis of research helps set the foundations for a common framework proposal. Furthermore, because technology related to mobile devices is evolving rapidly, disciplines in the MF ecosystem experience frequent changes. The rigorous and critical review of the state-of-the-art in this paper will serve as a resource to support efficient and effective reference and adaptation.

[*] Corresponding author. Department of Information and Communication Systems Engineering, University of the Aegean, Samos GR- 83200, Greece.

Email addresses: `tbarbatsalou@gmail.com` (Konstantia Barmpatsalou), `ddamop@aegean.gr` (Dimitrios Damopoulos), `gkamb@aegean.gr` (Georgios Kambourakis), `vkatos@ee.duth.gr` (Vasilios Katos)

October 24, 2013

Keywords: Mobile Device Forensics; Smartphone; Security;

1. Introduction

Internet and Information Technology (IT) are no longer a novelty, but a necessity in almost every aspect concerning people's lives, extending to a great variety of purposes, from business, education and public health to entertainment, commerce and even more. Models and behavioral patterns succumbed to changes in order to adapt to the new conditions that IT has created. It is thus inevitable that delinquent actions and patterns are expected to follow the same direction concerning their evolution and differentiation. Cybercrime, including the involvement of IT infrastructures in minor and major criminal activities, led to the creation of a new discipline, namely Digital Forensics (DF), equivalent to classical forensics where "evidence analysis takes place using data extracted from any kind of digital electronic device" (Harrill and Mislan, 2007). Although a digital device can participate in a crime by different means, "unless hardware itself is contraband, evidence, an instrumentality, or a fruit of crime, it is merely a container for evidence" (Casey, 2011b). Due to different attributes among computering devices, DF has developed several sub-disciplines, including Computer Forensics, Memory Forensics, Multimedia Forensics, Network Forensics, Small Scale Device Forensics or Mobile Device Forensics. (Casey, 2011b).

Technology concerning mobile devices has presented revolutionary growth during the last decade. Mobile phones, enhanced with hardware and software capabilities, not only serve as a means of communication, but also as small-scale portable computers with advanced communication capabilities. For instance, smartphones are able to store a rich set of personal information and at the same time provide powerful services, e.g. location-based services, Internet sharing via tethering, and intelligent voice assistance to name just a few. In addition to the traditional cyber attacks and malware threats that plague legacy computers, smartphones now represent a promising target for malware developers that struggle to expose users sensitive data, compromise the device or manipulate popular services (Damopoulos et al., 2011, 2012a, 2013). Additionally, the number of stolen or lost smartphones has increased rapidly over the last few years. In a research conducted by McMillan et al. (2013) among British legal databases from 2006 to 2011, the involvement of smartphones in delinquent activities presented an average growth of 10 cases per year. Some of the devices mentioned above may be used as a stepping stone (Chavez, 2008) to spoof the real identity of the attacker or to take advantage of the stored sensitive personal information.

Without a doubt, the widespread use of portable, small scale devices significantly increases the likelihood of a such devices being involved in a criminal activities. According to Jansen and Ayers (2007), MF is the science of recovering digital evidence from a mobile device under forensically sound conditions using accepted methods. The field of MF is challenging by default, due to the fact that smartphones have limited processing and memory resources, different CPU architecture and a variety of well-secured Operating System (OS) versions compared to those of a personal computer, making forensic processing a complex task. These challenges are compounded by the rapid rate of change in mobile device technology. While some forensic methods may be effective for a certain device or OS version, they may be useless for its successor(s). The variety of models and OSs can also raise a barrier concerning usage training. Investigators charged with the task of interacting with the devices have to be advanced users, in order to minimize the risk of human-driven errors. On the other hand, the amount of acquired data from small-scale devices may be considerably less than the amount of data retrieved from personal computers (Yates, 2010). Finally, when it comes to power consumption matters, which leads to resources vanishing quicker than in devices such as notebooks, since forensic labs are equipped with power supply cables, the only challenge occurs when the battery is almost depleted upon seizure. If the device shuts down before or during an *on-the-fly* acquisition, crucial information will probably be lost, including volatile data. Preservation of volatile data is becoming particularly important in modern devices in general, due to the proliferation of Web 2.0 and the underlying technologies. A representative example is *Volatility* (Volatile Systems, 2011), a memory analysis tool where addons have been developed in order to scavenge twitter posts and Facebook related data directly from RAM on mobile devices. Such data are dynamic and do not exist in the non-volatile storage.

Additionally, Lessard and Kessler (2009) mentioned that, one of the major difficulties in the field of MF is the general lack of hardware, software and/or interface standardization within the industry. This fact makes forensic processing a hard task, especially for unified research.

Attempting to cover the complete history of research and development in MF from the very beginning would be a time-consuming and outdated procedure, particularly because literature related to early developments in the field is scarce and older generation devices are no longer in use. The complexities of functionality in smartphones, combined with the growing number of such devices that are in use worldwide led to the decision to limit this research solely to the smartphones area.

Based on these facts, this work reviews and categorizes the main milestones, methodologies and significant studies for MF based on several factors, aiming to provide a comprehensive view of the state-of-the-art, by performing an in-depth study of the field.

More specifically, this work provides a thorough overview of the field of MF, by reviewing and presenting a detailed assessment of the actions and methodologies taken throughout the last 7 years. As further explained in Section 4.1, the decision to concentrate on the past 7 years of research and development in MF was based on the type and interdependence of the various contributions in the particular time frame and in an effort to provide a holistic view of the state-of-the art in MF area. In short, an effort was made to focus the time period to the minimum possible – this is critical for this rapidly evolving area – but doing so without omitting important parts that would make difficult to assemble the whole MF puzzle. A schematic timeline of the most significant studies so far is given, in order to provide a quick but complete way of observing the milestones and trends within the field. By doing so we offer an analytical progress report concerning the evolution of MF. Since standardization activities in the area are quite far from being mature, this work can help shape a common framework for MF. As already pointed out, technology concerning mobile devices is evolving at a rapid pace, and MF must continue to adapt to these frequent changes. A strong fundamental infrastructure will help support efficient and effective adaptation. As a result, newcomers or even experts of the MF field will be able to have a compact image of the state-of-the-art. Also, this work exposes existing problems in the field of MF, thus providing motivation and future direction for further research in the area.

While there is a satisfying number of studies throughout literature (Yates, 2010; Hoog and Gaffaney, 2009; Grispos et al., 2011; Satheesh Kumar et al., 2012; Casey et al., 2010) that have cross-evaluated commercial forensic suites for extracting conclusions about the retrieved data, this is the first work that presents a different approach focusing more on new methods, *but not totally neglecting the former*. For instance, in some types of smartphone platforms, the academic work is practically nonexistent while commercial tools have an excellent coverage. Therefore, due to the fact that the development of commercial forensic tools is at large based on elements of the main acquisition methodologies (addressed in this paper), we did not dedicate an extended session on them.

The rest of this work is structured as follows. Section 2 enumerates, classifies and analyzes the criteria that will be used throughout this work, providing this way the necessary background knowledge. Section 3 surveys the state-of-the-art in the field. Section 4 elaborates on the research work done so far contributing a complete categorization of the major MF approaches. The last section draws a conclusion.

2. Preliminaries

Considering the guidelines presented by ACPO (ACPO, 2007) and the EDRM model (EDRM LLC, 2013), which are dealing with forensically sound practices, this work classified the context of MF into the following three categories that are suitable for conducting the analysis and comparisons used throughout this research: Evidence *acquisition methods*, *Operating Systems*, and Acquired *Data Types*.

2.1. Acquisition Methods

Forensic acquisition from devices is divided into three categories: manual, logical and physical. Each one uses different attributes of the device for extracting the desired amount of data. Manual acquisition is defined as whatever an individual is capable of acquiring by interacting with the device itself. This procedure may consist of two separate phases: keeping a log of the actions taken (Grispos et al., 2011) and interacting with installed applications to copy the existing data (Mokhonoana and Olivier, 2007). Additional means, such as cameras can be used in order to record the device state (Grispos et al., 2011). Since the probability of human error is very high and crucial elements can be bypassed, this method should be used as supplementary. Due to the fact that manual acquisition is the only technique returning data in human interpretable format, it is necessary to take place simultaneously with the other two kinds. As a result, it will not be examined as a separate category, but will be integrated in to the other two.

Logical acquisition retrieves a bitwise copy of entities such as files and directories that reside inside a logical storage means and “provides context information for the formerly mentioned objects, such as date-time stamps and location within the file system of the target mobile device” (Casey, 2011a). It mainly concerns data that has not been deleted and is achieved by accessing the file system of the device (Hoog, 2011). Nevertheless, information that is not practically deleted , but “disguised as” available space for further overwriting within databases may be retrieved by *file system* access. Data that has already been deleted are less likely to be acquired. Logical acquisition techniques and tools interact with the file system whereas physical acquisition methods access lower areas. This leads to the conclusion that physical and logical acquisition show different strengths and weaknesses concerning the files they retrieve. For instance, physical acquisition retrieves deleted files, whereas logical acquisition is more efficient for recovery of user data (call and SMS logs, contacts) (Grispos et al., 2011). “Sometimes logical acquisition is not possible, for instance when the device is broken beyond repair, or when the device does not have a standard interface to do the logical acquisition over”(Klaver, 2010). Physical acquisition may also be conducted before logical, if there is no other way to bypass user security mechanisms such as passwords and screenlocks (Breeuwsma, 2006). Summarizing, logical acquisition can be divided to the following categories: partition imaging, copying files-folders, content provider and Recovery Mode (Hoog, 2011; Vidas et al., 2011; Son et al., 2013).

On the other hand, physical acquisition is solely related to the physical storage medium. Such a technique is also mentioned as a bitwise copy of the internal flash memory (Grispos et al., 2011; Quick and Alzaabi, 2011; Thing and Chua, 2012). This kind of acquisition is more likely to retrieve deleted data (Husain et al., 2011), which is treated as unallocated but still exists in memory. However, physical acquisition procedures are more likely to damage the device while it is being dismantled. According to (Klaver, 2010) “True physical acquisition can either mean physically removing memory from the device, using hardware techniques like Joined Test Action Group (JTAG) (Breeuwsma, 2006) to extract data from the device or use an (adapted) bootloader to gain low level access to the device”. These kinds of techniques “are not only technically challenging and require partial to full disassembly of the device, but they require substantial post-extraction analysis to reassemble the file system (Hoog, 2011). Nevertheless, it is generally acceptable in a forensic context that physical acquisition prevails over logical, because it allows deleted files and any data remnants present to be examined (Jansen and Ayers, 2007). Sometimes, however, as in the case of the Windows Mobile OS (Klaver, 2010), research has led to the development of alternate acquisition methods that lie somewhere in between a physical acquisition and a logical one (usually referred to as *pseudo physical*).

2.2. Operating Systems

A factor of heterogeneity which is an impediment against the development of a common MF framework is the existence of different OSs (mobile platforms). Current market share gives Android and iOS the prevailing percentages (Becker et al., 2012). Other OSs, such as Blackberry and Windows Phone remain also a popular choice. In the past, the need to exploit vulnerabilities in these operating systems in order to perform physical acquisitions posed a challenge to admissibility in court (ACPO, 2007; Jansen and Ayers, 2007). However, such concerns have decreased as MF techniques became more mature and better understood.

In generalized terms, *low-level modifications* grant access to system areas which were by default protected by each OS manufacturer. The privileges users are gaining after the application of a low-level modification vary among different OSs. Low-level modifications can have a variety of names depending on the OS they are applied to. They are either known as Rooting (Android, Windows Mobile), Jailbreak (iOS) or Capability Hack (Symbian). For example, Android users are able to install and run applications that require access to the root directory, such backup features. In addition to root privileges, iOS users can install applications not available in the AppStore. *Capabilities* on Symbian devices are security mechanisms that can be bypassed by installing a root certificate and thus allowing users to install and execute unsigned applications. A brief overview of the OSs characteristics will be made in the next paragraphs, alongside with their impact to forensic acquisition.

Android was first released in 2007 and in less than five years achieved to be the dominant OS in the mobile handsets market. The OS runs on a Linux 2.6 - based kernel, which serves for supporting fundamental functions, such as device drivers, network infrastructure and power management (Yates, 2010; Hoog, 2011; Vidas et al., 2011). The next level of the Android architecture is the domain of the libraries, split to application and Android runtime ones. The former category provides the appropriate infrastructure for applications to run properly, such as binaries and graphics support, while the latter consists of the Dalvik Virtual Machine (DVM) and the core libraries that provide the available functionality for the applications (Yates, 2010). Its main purpose is the creation of a stable and secure environment for applications execution. Each application runs in its own sandbox (virtual machine). Therefore, it is not affected by other applications or system functions. Using certain resources is only permitted by special privileges. This way, a satisfying level of security is preserved. While the Android Runtime Libraries are written in Java (Yates, 2010), DVM translates Java to a language that the OS can perceive (Simao et al., 2011). The rest of the architecture consists of the Applications Framework and the Applications Layer that manage general application structure, such as containers, alerts and the applications themselves.

Due to the small chip size, non-volatile nature and energy efficiency, NAND flash memory was selected to equip Android devices for storage purposes (Hoog, 2011; Zimmermann et al., 2012). NAND flash memory needed a file system that was “aware of the generic flash limitations and take these into account on the software level when reading and writing data from and to the chip”(Zimmermann et al., 2012). Yet Another Flash File System 2 (YAFFS2) was the first file system implemented for devices running Android. After some years of actual use on the other hand, many issues concerning system performance, velocity of input/output actions and large files coverage occurred. As mobile devices architecture tends to follow the path of desktop computers and acquire multiple core processors, another obstacle arises, since YAFFS2 cannot support the specific technology (Kim et al., 2012). Right before the release of ver. 2.3 of the OS (Gingerbread), the file system was replaced to EXT4. The specific file system, apart from successfully coping with the weak points of YAFFS2, is enhanced with the *journaling event function* (Kim et al., 2012), which provides recovery options and facilitates acquisition of unallocated files.

Android provides potential developers with the SDK (Software Development Kit), which includes a very important tool for forensic and generic purposes, the Android Debug Bridge (adb). Adb uses a TCP or USB connection between a mobile device and a computer. The appropriate software is installed at both sides in order to acquire debugging information, start a shell session with the provided interface, initiate file transactions and add or remove applications (Hoog, 2011; Simao et al., 2011; Vidas et al., 2011). Since adb grants a terminal interface, actions like rooting and memory image extraction can be easily performed.

NAND flash memory was incompatible to the Linux-based kernel. A new technique had to be implemented to provide the software components with the ability to access the flash memory areas (Vidas et al., 2011). The Memory

Technology Devices (MTD) system was one of the facilities serving as an intermediary between the kernel and the file system and is present in many Android devices. Handsets that do not support the MTD system usually utilize the plain Flash Transaction Layer (FTL) that enables communication between the two parts (Hoog, 2011). Although there are no restrictions concerning the MTD numbers or types, a certain standard had been adopted from many device manufacturers (Lessard and Kessler, 2009; Hoog, 2011; Vidas et al., 2011). MTDs are divided to several partitions, according to the type of information they store. They can contain information about booting, recovery, user data, configurations, cache and system files.

Blackberry OS devices are designed by the Research in Motion (RIM) company and have a diversity of popularity among different countries and groups worldwide. Few things concerning the Blackberry OS itself and its ingredients are known from official sources, since the manufacturer does not provide sufficient documentation. However, substantial amounts of information concerning support were obtained via reverse engineering. These acquisitions are certain to trigger further research. A significant attribute concerning the OS is that it consists of two separate runtime environments, one Java ME-based destined for applications and one MDS-based, destined for network functionality and operations. One of the most forensically interesting elements of the OS is the *Interactive Pager Backup (IPD)* file, a collection of databases where information such as call logs, SMS and other user data are stored (ipddump, 2011). MacOS X is also known to generate IPD .zip compressed files, but with the *.bbb* suffix (Forensics Wiki, 2012). User data, such as contacts, messages, images and OS artifacts are stored in databases, which are the acquisition target of every forensic operation.

iOS was first released in 2007. It is a UNIX-based OS, partially following the architecture of the MacOS X equivalent. The main storage device of a mobile phone running the iOS is divided into two partitions. The first contains the OS fundamental structure and the applications, while the second contains all the user-manipulated data (Husain et al., 2011). The two bottom layers, Core Services and Core OS provide support for low-level data types, network sockets and file access interfaces. The Media Services layer consists of the infrastructure responsible for 2D and 3D graphics, audio and video. Finally, the Cocoa Touch layer contains two subcategories, the UIKit, which is equipped with the appropriate interface material for applications and the Foundation framework, which is supporting file management, collections and network operations (Yates, 2010).

Maemo is a Linux-based, open source OS. Even though it is not widespread and its development has been frozen since Oct. 2011, there are some research-oriented interesting features, such as the fact that user data, OS functions and swap spaces are situated in different partitions (Lohrum, 2012).

Apart from popular brands, massive production (Fang et al., 2012b) is also present in replicas of the former. One of the most popular brand names of this kind is the *Shanzhai* iPhone imitation. Their low cost is a motivation for potential buyers. In addition to the fact that they are not easily tracked down, they make a valuable "weapon" for delinquent actions. Lack of documentation on infrastructure and system manuals provokes impediments for forensic investigations.

Symbian is one of the older OS in the category, with its first release taking place in 1997 as EPOC 32 and discontinued after January 2013. Applications are mainly written in Java, while its native language is Symbian C++ (Mokhonoana and Olivier, 2007). Since many different versions of the OS exist, it is inevitable that slight variations concerning its architecture will also be present. The UI Framework is the upper level and consists of the infrastructure responsible for user interface functionality. Below that resides the Application Services Layer, hosting essential services for applications to run properly. A separate layer is devoted to Java ME, in order to provide compatibility with the OS. It contains the virtual machine and some supportive packages. Networking services, handlers and components, graphic support elements and generic services are combined under the OS Services Layer. Lastly, the lowest level concerns the hardware and kernel infrastructure (Morris, 2006; Yates, 2010).

WebOS is a Linux-based OS, designed especially for HP smartphones and tablets. It can be considered as a hybrid mobile OS, since its partitions have different formats, according to the use they had been destined to (Casey et al., 2011). The user partition, which contained user generated data and multimedia files had an FAT 32 file system format, whereas the system partition had an ext3 file system. The fact that it was supporting a bunch of innovative features (Hewlett-Packard Development Company, L.P., 2011), such as a web browser backbone infrastructure (Hewlett-Packard Development Company, L.P., 2011; Casey et al., 2011) and multitasking was not able to hamper an upcoming fall; from being the default OS for many devices, it was purchased by LG electronics so as to equip Internet TV sets after some modifications.

Similarly to other mobile OSs, it also provided an SDK and a special version giving the developers the opportunity to interact with code situated in lower levels of the system, such as binaries and libraries, mainly written in C and C++.

The *Windows Mobile OS* is the evolution of Windows CE, used mainly on handheld devices, such as palmtops and PDAs (Satheesh Kumar et al., 2012). The Windows Phone OS is its successor, with many structural elements of forensic importance in common, such as EDB files (Kaart et al., 2013). It is a Windows-based system, with similar properties specially modified so as to apply to the nature of mobile devices. One of the basic examples in this category is its file system. The T-FAT file system (Transaction-safe FAT) is a variation of the FAT file system used in desktop versions of Windows, enhanced with recovery options (Klaver, 2010; Yates, 2010). Devices incorporating this OS support NOR and NAND flash chips, so it is possible to either encounter a device running all its functions solely on one of the two categories, or a hybrid running its OS from NOR and storing its user data in NAND. Likely to mobile OSs mentioned before, the architecture of the Windows Mobile OS consists of similar layers. That is, the upper layer, Application UI the median between the user and the applications and the lower layer (above hardware) that provides the appropriate infrastructure for completion of system-oriented routine tasks, such as start-up, networking and other functions (Sasidharan and Thomas, 2011). The Framework and CLR layers contain libraries serving to execution and performance of applications.

2.3. Data Types

Data acquired from forensic examinations can be also classified, depending on their types and the entity that has access to them. The first group consists of data handled and altered strictly by OSs, such as connection handlers (GPS, WiFi) and OS defaults and structural elements (IMEI, IMSI). The second group concerns data imported and edited by users, such as text messages, contact lists, pictures and all sorts of customized application data. Data used by applications as background procedures and other similar entries manipulated by applications, form the third category.

Table 1 offers a complete view of the areas where forensically significant data are stored in each mobile platform. Note that Databases and External Storage are present in every OS, thus pointing out that a common framework can be implemented towards their acquisition. Taking into account previous research in the field, it seems than the RAM Heap as source of information was only researched in Windows Mobile and Android carrier devices (504ensics Labs, 2013) despite the fact that other OSs may also store valuable data in it. The more popular an OS is, the more research effort is anticipated to be devoted to it. However, this not always the case especially between different countries. For instance, while in Canada the BlackBerry platform is still quite popular, relatively little academic research has been devoted to it, but Cellebrite has made significant progress acquiring Blackberry devices physically and decoding the information they contain. Popularity though is only one of the factors that affect potential research activity. The availability of documentation concerning a specific OS also contributes to future decisions on research. For example, Blackberry was known for the lack of distributed documentation, so it was harder, but not impossible to examine.

Table 1: Forensically Significant Data per OS

OS	**Databases or Files**	**External Storage**	**Shared Preferences**	**Network**	**System Logs or Registry**	**RAM Heap**
Android	X	X	X	X	X	X
Blackberry	X	X				
iOS	X	X		X	X	
Maemo	X					
Symbian	X	X				
WebOS	X	X		X	X	
Windows Mobile	X	X		X	X	X

3. State-of-the-art

3.1. Standards Background

Given the fact that MF is a relatively new discipline and presents big deviations from computer forensics, developing standards in this area is challenging. Some early efforts on MF standardization include the series of guides by ASTM International (2009), focusing on *digital and multimedia evidence* principles and Best Practices for Mobile Phone Forensics by SWGDE (2009). Both organizations have been publishing updated versions of the standards until 2013. One attempt to create an ISO certification was published in Oct. 2012, containing guidelines of general acceptance. “The fundamental purpose of the digital forensics standards ISO/IEC 27037, 27041, 27042 and 27043 is to promote good practice methods and processes for forensic investigation of digital evidence”(ISO/IEC, 2012).

Within his research, Marshall (2011) reviewed the state-of-the-art dealing with standardization. Similarly to ancestors and successors, the author annotates the standardization problem and enumerates attempts of publishing guidelines and standards. One of the most distinctive attributes of the research was the schematic representation of the relationship among the ISO /IEC 27xxx standards. Each entity of the diagram refers to an ISO publication and is mutually connected to the others. Right above the standards, we can find the *Investigation Principles and Processes* level, which has one-way relations to the entities. The ISO upcoming publications are sorted according to the EDRM model (EDRM LLC, 2013). The relevance between ISO publications and model entities are described in Table 2.

Table 2: ISO and EDRM Relevance

ISO Publication	**EDRM Entity**
ISO/IEC 27035 Incident Management	Information Management
ISO/IEC 27037 Identification, Acquisition and Preservation of Evidence	Identification, Collection and Preservation
ISO/IEC 27041 Assuring Suitability and Adequacy of Methods	Processing and Review
ISO/IEC 27042 Analysis and Interpretation of Digital Evidence	Analysis, Production and Presentation

National Institute of Standards and Technology (NIST) has been actively involved in publishing guidelines for MF investigation regulations, in a series of Special Publications (SP). The two most recent publications are SP 800-101 (Jansen and Ayers, 2007) and Reference Materials NIST IR-7617 (Jansen and Delaitre, 2009), which had superseded the obsolete SP 800-72 (Jansen and Ayers, 2004a) and NISTIR 7100 (Jansen and Ayers, 2004b) issues. At least for the time being the work conducted by Jansen and Ayers (2007) is considered a milestone in the field of MF. However, it seems that this version of guidelines is going to be replaced soon by the later candidate that is currently under review (Ayers et al., 2013). For more information about this under preparation publication, refer to Section 4.2.

However, these guidelines are not a theoretical set of rules, but rather assumptions deriving from experience in field investigations. They determined their own work as means *to help organizations evolve appropriate policies and procedures for dealing with cell phones, and to prepare forensic specialists to contend with new circumstances involving cell phones, when they arise* (Jansen and Ayers, 2007). They set their report of standards as an essential background for every upcoming investigation but not as a strict set of orders, since exceptions to rules can always be present. Their report was edited according to the regulations imposed by the Federal Information Security Management Act (FISMA) of 2002, Public Law 107-347.

The authors also state that a forensic investigation must be a sequence of actions, consisting of the following steps: acquisition, examination, analysis and reporting of retrieved data (Jansen and Ayers, 2007). An official set of regulations demands a proper technical background setup. As a result, the authors provided the potential readers with the most important infrastructure elements of mobile phone providers' networks and devices. It is therefore notable that researchers who took this guide into consideration followed the same pattern with small or no deviations, since it was used as a list of prerequisites and limitations in many research papers (Bader and Baggili, 2010; Morrissey, 2010; Pooters, 2010; Lai et al., 2011; Vidas et al., 2011).

Afterwards, they classified the acquisition techniques into two main categories; physical and logical, while making a brief explanation of the characteristics of each one. They proposed that conduction of both acquisition types would be the most complete solution under real-time investigation circumstances. In addition, they enumerated the officially certified forensic tools and the attributes coverage each one has. By the time of publishing, the majority of tools implemented concerned SIM modules. Tools supporting mobile devices OSs have also been developed, but in a more limited scale.

Lastly, on the specific area of interest, they posed the main challenge concerning efficiency of forensic tools. That is, an acceptable tool should be capable of preserving the data integrity of the retrieved data to their state right after the completion of acquisition. This is typically achieved by the use of cryptographic hash functions, which are used to prove if the evidence was preserved by means of verifying its integrity, and therefore are detective in nature.

Their study also concerned the setup of the investigation scene and limitations that would prevent an acquisition from being admissible upon court. The most significant part of the research was the one concerning data handling upon and after the crime scene. Based on the *Good Practice Guide for Computer-Based Electronic Evidence* (ACPO, 2007) guidelines, they concluded that there should not be any data modification after device seizure and every step of the investigation should be documented by certified professionals and guided by the overall responsible practitioner. These specific restrictions are the main source of trouble to the technical part of the investigation because their attributes are contrary to the nature of mobile devices. Description of data and evidence preservation was the next key point in their report. Seized devices should be stored in forensically *sterile* means, be disconnected from any networking source and their state should remain as close as possible to the one it had at the moment of discovery. "Since it is impossible not to alter the state of a device upon seizure, the specific principle is usually omitted. The next acceptable and realistic option is to leave unaffected critical data that are supposed to be presented upon court. In exceptional circumstances, where a person finds it necessary to access original data held on a computer or on storage media, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions" (ACPO, 2007). That is, since modifications are inevitable in the case of examining a smart phone, the first responder would need to be an expert in the underlying technology in order to be able to fully explain and defend any alterations performed on the device, which were inevitable and a direct consequence of their acquisition and analysis activities.

Moreover, investigators are challenged to acquire any possible non–technical evidence from the device, such as fingerprints and DNA, before proceeding to further technical examination. Network isolation had been the apple of discord, since there have been two different disciplines towards that direction. More specifically, switching the device off not only alters the device state, but it might also invoke security mechanisms (when switched on again), such as PIN code and screen lock protection that will cause new impediments to the investigation; while network isolation through *flight mode* would alter the device state as well. One of the first dilemmas concerning acquisition, was whether it should be performed *on the fly*, or inside a forensic laboratory. After enumerating the essential information about the device without interacting with its software, investigators were able to begin the acquisition procedure for all the memory types present on the device. The first step was to ensure that certain prerequisites are present so the acquisition procedure to start without any errors or misuses. Some important features to be acquired in the beginning were date and time. Extra attention should be paid in case of intentional changes on date and time from the side of the suspects or unintentional, in case of battery removal. In this direction, the authors provided an extra appendix with a detailed acquisition process, enhanced with screenshots. They next set a priority queue for acquisition types. Due to the risks that may appear during physical acquisition, it is preferred that logical acquisition should be performed first. Then, acquisition for identity modules is described.

A whole chapter section is devoted to powered-off devices or handsets that require any kind of security code, providing the available solutions in order to bypass or acquire the essential codes that can be either hardware/software driven, manual or even supported by help from a provider.

Special sections are also provided for removable storage media and other peripherals. The section concerning the examination procedure is the phase after acquisition where there is no longer interaction with the device. Logical and/or physical images had been examined for data of forensic interest, or even for data that could work as medians to other data, such as passwords stored in a database. Acquired data had been classified to an extent that can determine the actors, places, time and motive of an incident. The same section also contained information about subscriber details and call records that can be acquired from a telephony service provider. In the last chapter referring to reporting, the authors provide a detailed guideline concerning on diffusion of the data acquisition results.

3.2. Android Forensics

Lessard and Kessler's work (Lessard and Kessler, 2009) concerns the procedure of acquiring a physical image and performing logical acquisition on an Android device. The test phone used during the experiments was an HTC Hero running Android 1.5. They suggest a simultaneous examination of both the flash memory and removable SD memory card. Removable memory cards may contain data that could be useful to the investigators. Examination process referring to the removable card can easily be performed by a commercial tool, in their case AccessData's FTK Imager v.2.5.1. The use of a write blocker was a prerequisite, in order to avoid data modification. This is partially achieved by documenting all the actions performed on the target device. It can later be verified that the sequence of actions did not compromise the integrity of the evidence (Mokhonoana and Olivier, 2007; Jansen and Ayers, 2007). Internal memory contains different types of data, such as contacts and calls lists, text messages etc. The type of physical acquisition used in this work could not take place without gaining superuser privileges on the target phone. Rooting is achieved by using a third party program or an exploit, but it alters system data and results in preventing the acquisition from being forensically sound (ACPO, 2007). The authors assume that this is the only way to achieve an acquisition of a physical image at least until some other way of altering data in a minor scale is discovered. After successful rooting of the target phone, the investigators were able to access every possible area within the phone memory. MTD blocks, which allow for the embedded OS to run directly on flash, contain useful information, such as system files and user data (Lessard and Kessler, 2009). By using the disk duplicate (dd) command on adb shell, images of each mtd file present in the "\dev\mtd" directory were acquired. Afterwards, memory image files were examined by a commercial forensic tool kit and useful files and entries are extracted. Using a forensic toolkit not only facilitates the forensic acquisition, but also enables partial or full reading of fragmented and/or corrupted files. Flawless extraction of data was impossible. Nevertheless, data from all three categories (strictly OS manipulated, user data and application data) were returned to the researchers and formed a satisfying image concerning the use of the target mobile device. Useful information has also been retrieved from system databases and web browsers via logical examination of the "\data\data" directory.

Cellebrite UFED, another hardware forensic tool performing logical acquisition only was used by the authors in order to have a more accurate image of the extraction results. After gathering and comparing data extracted from both kinds of acquisitions, Lessard and Kessler concluded that each one contributes by providing satisfying results in different kinds of data. Despite the fact that their research was conducted during the very beginning of the Android forensics discipline, the results provided are accurate and the research attracted significant attention in future works.

Andrew Hoog (Hoog, 2011) set the milestone in the field of Android forensics. After a brief introduction to forensics of all scales, he categorized and presented in a detailed way the two dominant acquisition techniques, logical and physical. In this work also, the data integrity and presentation of a forensically sound set upon court was disputed, since the nature of mobile phones functionality is not compliant to the existing guidelines for computer forensics (ACPO, 2007). At first, Hoog set the framework for handling a device before the beginning of any acquisition process. The steps proposed include bypassing user passwords for screen locking or deactivating it by the settings menu, dealing with switched-off devices, isolating the device from any network sources and being aware of providing charge when needed, so as to avoid interruption before or during the acquisition process. Right after enumerating the possible ways of network isolation and evaluating the advantages and disadvantages of each one, the author concluded that the most appropriate method is enabling flight mode of the device, if possible. Then, he suggested the use of adb in order to check for an existing usb connection and then perform the task of logical acquisition. Naturally, if the target device is not rooted, adb connection will not be completed successfully.

Later on, he described the physical acquisition techniques, Joint Test Action Group (JTAG) and physical extraction (chip-off) and proposed their application when the investigators wouldn't be capable of performing a logical acquisition. Hoog provides an in depth presentation of a forensic examination of the removable (SD) or embedded (eMMC) media, while similar field studies simply neglect the subject. Bypassing security codes was a major issue faced. One applicable technique, concerning on-screen shape locks, was the so called smudge attack. That is, pattern locks could be revealed by using different angles when shedding lighting on the screen.

An alternate solution was booting into recovery mode, considered obligatory if the device had been powered off. Actually, booting into this mode consists a popular approach for embedded devices (Vidas et al., 2011). The author focused on the use of a write blocker in order not to alter the state of the device by mistake. Afterwards, he enumerated and described the most popular logical acquisition techniques, including adb, AFLogical tool, Cellebrite UFED and others. Also he provided a detailed description of hardware (JTAG) and software (bootloaders) physical acquisition techniques and emphasized in the importance of the use of MD5 hash functions in comparing the acquired files.

As smartphones started becoming more complicated than their ancestors, new attributes have been added at their functional state. One of them is the Global Positioning System (GPS), uses of which varied from checking in actions for social networks to navigation purposes and images metadata. Data logs from such applications or even raw forms from the GPS system itself are a valuable source for investigators, since they are able to enlighten cases in depth, especially when they are enhanced with timestamps. In this context, the work by (Maus et al., 2011) introduces a technique that serves in acquiring all the available geodata present inside a smartphone. They claimed that rooting is obligatory in order to gain access to applications containing the needed data. The extracting procedure consisted of the following steps: (a) performing search queries for attributes such as latitude, longitude, and altitude in application databases, and (b) storing them to a database suitable for the specific type of data. Also, the authors correctly observe that data concerning geo-position can be stored in other forms, such as text metadata that could make their acquisition a difficult task. If so, the data can also be retrieved by searching for specific terms related to location and then stored in a database appropriate for their form.

Geodata conversion functions are taking geodata or metadata as input and return them in human interpretable format for further editing (XML). A way of presenting geodata in a compact form is by the use of an API, similar to Google maps, containing pins and other significant enhancements. Apart from specific locations, the presentation procedure can also return the routes the device owner followed. Taking this into account, the researchers implemented a forensic acquisition tool, coined Android Forensic Toolkit, suitable for Android ver. 2.2 devices and geodata analysis and presentation was a feature under development. No linkage between the data extraction procedure and the program was provided. They came to conclusion that forensic analysis deriving from geodata is a precious source of evidence and the data to be extracted should be taken into consideration in potential acquisition procedures.

Lai et al. (2011) implemented a live-forensic acquisition procedure, based on commercial forensic suites through cloud computing, designed for Android devices. After a brief introduction to the Android OS and forensic legislative guidelines (Jansen and Ayers, 2007), they enumerated the prevailing features of cloud computing and how it could preserve a secure background, suitable for conducting forensic acquisition. They concluded that a cloud computing service, Google Cloud Service in their case, could satisfy a variety of conditions, such as security prerequisites, browser-based applications, bigger storage capacity and lack of time and location restrictions. At first, the researchers demonstrated the system architecture, which consisted of an https bridge between the cloud service provider and the workstations and devices. As soon as investigators downloaded the appropriate forensic software, they would be able to start data extraction. Acquisition type was not specified, but, judging from the data types the software was able to retrieve and the fact that no rooting techniques were mentioned, the procedure resembled to logical acquisition that can apply to rooted devices as well. One interesting and unique feature of the method was actual date correction, since proper time-stamping is an essential for the validity and integrity of forensic analysis methods. The research would be considered complete as soon as results of acquired data would be diffused. A study focusing on comparison between cloud and classic forensic acquisition will enlighten the effectiveness of the method.

YAFFS and YAFFS2, the file systems present in devices running the Android OS, are the most frequently discussed within literature. The authors in (Quick and Alzaabi, 2011) performed an experimental research, with logical and physical acquisition techniques and tools (adb pull, NANDump, xRecovery and Yaffs2utils) on a rooted Sony Xperia 10i device. Logical acquisition was not able to acquire the full size of the file system, while physical, as expected, achieved a bitwise acquisition of the flash memory. Physical acquisition with spare data included followed a different approach, since the researchers needed to rebuild the YAFFS folder structure. Through the hex viewer WinHex, they had been capable of recognizing the retrieved file headers and add timestamps to many of them by converting the equivalent hex values. Since many different tools were used, acquisition results had been fulfilling. Quick and Alzaabi claimed that NANDump has generated a complete copy of the internal NAND memory. Lastly, they proposed the implementation of a tool or method which would be able to directly read and interpret the YAFFS file system elements. Such a utility would facilitate forensic investigations. Since Google swapped the file system type to EXT4, such a research would be useful for the older devices, but outdated.

Simao et al. (2011) proposed a forensic acquisition framework for the Android OS. Their framework has been presented in a flowchart form, since there had been many different states of target devices, such as rooted or not, switched on or off, upon access control or not. Even if their model can be applicable to many scenarios, it is missing some crucial elements concerning a real-time investigation. The additional information in their proposal concerned acquisition on damaged devices and fragmented memory page analysis. In order to validate the effectiveness of the model, (Simao et al., 2011) conducted experiments on devices with different conditions and figured out that the proposed scheme was applicable. However, they admitted that further research should be conducted so as the framework can be kept up to date with the upcoming versions of Android. A more enhanced version of the existing model was introduced by (Park et al., 2012), even though their goal was not the implementation of a framework.

Sylve et al. (2012) referred to a lack of studies applicable to physical acquisition in the context of MF. They highlighted the importance of this issue, unlike most other research which bypasses the subject. The researchers presented "a methodology for acquiring complete memory captures from Android, code to analyze kernel data structures and scripts that allow analysis of a number of user and file-system based activities" (Sylve et al., 2012). Also, they enumerated the existing methodologies on volatile memory analysis for Linux and Android OSs and compared the capabilities of the corresponding tools. Before proceeding to acquisition, they had to face the rooting challenge. They considered it a necessary evil because the code expected to return the memory image had to access the device kernel. There had also been an attempt for memory acquisition with the use of methods destined for the Linux OS.

The results of their experiments proved that Linux oriented techniques were incompatible to the Android OS, since plentiful bugs, such as not existing functions, limited size of offsets supported by the (well-known) dd command and insufficient percentage of acquired memory appeared. Moreover, some global issues arose, since not every kind of device showed identical behavior. The primary cause for this incident was the difference among ROM types and kernel modes and is yet to be discussed in future work.

Next, they presented the implemented method, namely DMD. The procedure consisted of the following steps: accessing the iomem_resource kernel structure to acquire the beginning and ending point of RAM, converting to virtual memory and copying the selected segment to a removable storage device or a TCP port. This method prevented from parsing of useless parts of the memory and enabled unlimited execution of commands, such as dd, cat etc Additionally, the procedure had been less time and resources consuming.

The researchers performed a case study of both types of acquisition on a rooted HTC device. TCP acquisition used adb bridge to achieve port forwarding between the device and the workstation. Communication between the two sides was established through a socket and a message header containing the memory range limits triggered the image acquisition. Memory image was acquired through port 4444 and when the procedure completed, DMD terminated the existing connection. They observed that one of the main differences between TCP and SD card acquisition is the initiate parameter path in the iomem command. The rest of the procedure was similar to memory dumping on SD cards in other research papers. Last but not least, they proposed new aspects on future research, concerning the Dalvik VM memory analysis, which offers an analysis to the total Android applications space.

Vidas et al. (2011) took research to a different level, facing the challenge of forensic acquisition on devices protected by a screen lock. Since a brute-force attack on the device was not a preferable method and may lead to further blockage and inevitable data modification, another technique had to be implemented. In this direction, booting with a recovery image could easily bypass any kind of active lock code. After enumerating the criteria for a proper forensic analysis, they proposed an acquisition method based on the use of an acquired recovery image and adb software on the workstation the device is connected to. One of the MTD files present in the root folder of Android devices, known as mtd3 (recovery mode boot) was significant for the acquisition process of the recovery image. “By booting a device into recovery mode, the normal boot process is circumvented and the boot target is the bootimg currently loaded in the recovery partition” (Vidas et al., 2011). After this step, using a modified boot image (bootimg) for the device became a routine task. The bootimg the writers used consists of existing modified files used in adb activation, the most useful demons (dd, nand, su, dump) and other transfer binaries. The researchers implemented TCP transfer software and used a hash value for integrity preservation of the data moving back and forth. Last but not least, they providence for data dumping whether the target device was MTD based (NAND dump) or not (dd command). Unlike many other research papers (Lessard and Kessler, 2009; Hoog, 2011; Racioppo and Murthy, 2012), the specific one disapproved of rooting the target devices, presenting some potential disadvantages of the method. Also, the authors correctly observed that boot options differ between different brands of mobile handsets. As a result, they examined three separate case studies; one of which was a Samsung device without MTD partitioning. A weak spot of the research is that there were no statistic results of the retrieved data. However, this can pull the trigger for future experiments and case studies, since the technique can be applicable to all kinds of data concerning logical acquisition.

Case studies are experiments adapted to real time conditions. In the field of MF, where device behavior is unpredictable (Grispos et al., 2011), case studies can greatly contribute to the creation of a holistic pattern concerning investigations. The work by (Racioppo and Murthy, 2012) presented the case study of physical and logical forensic acquisition to *HTC Incredible*, a device running the Android OS, ver. 2.3. The first part of the device to be examined was the removable storage media and since its structure is relatively identical to the ones used in desktop computers, a computer forensic tool, AccessData FTK Imager ver. 3.0.1 was used for acquiring a physical image copy. It is notable that the researchers used a write blocker to preserve the forensic soundness of the copy. For data integrity purposes, a hash value of the extracted image was used. Physical acquisition of the internal memory was a more complicated procedure. No useful data could be extracted if the phone was not rooted, so, even the technique wouldn't be admissible upon court, they used a third party program in order to root it. After gaining access to the root directory, they were able to create a bitwise copy of the seven MTDs present in /dev/mtd folder. The next step concerned analyzing the acquired images of the MTDs by using the Ubuntu program scalpel, along with Andrew Hoog's *scalpel-Android.conf* (Racioppo and Murthy, 2012; Hoog, 2011). After examining the results, they concluded that physical acquisition was as effective as described; even unallocated files were retrieved. The fact that some files were corrupted or destroyed was considered a normal side-effect of the method. Logical acquisition provided access to areas where physical was unable to, such as databases. Every piece of information stored on databases, such as contacts, GPS positions, voicemails etc had been retrieved. As in the majority of research papers the key future challenge the authors identified was the ability to root a device without disabling its forensic soundness.

Andriotis et al. (2012) implemented a forensic acquisition method concerning the usage of wireless networking devices (WiFi and Bluetooth) on four devices running the Android OS. One of its characteristics worthy to mention was the fact that results did not present big deviations, even if they were carrying a different version of the OS. Similarly to studies mentioned previously, they made a brief introduction on Android hardware infrastructures and software characteristics. Secondly, they enumerated some milestones of the MF ecosystem and examined their compliance to the ACPO (2007). One of the most significant parts of the research was indisputably the fact that devices used were involved in actual crime scenes. Afterwards, they presented a detailed step-by-step procedure to complete logical acquisition, which was common for all the devices participating in the experiment. The experiment was considered a success, since critical evidence was recovered in every networking attribute. Later on within the research, the authors created a complete table consisting of the paths were data acquired had been spotted.

EXT4 became the successor of YAFFS after the release of Android ver. 2.3. Among other features, this version supports the Journaling File System, which, by keeping a record of actions, enables error recovery mechanisms. During their experiment, (Kim et al., 2012) used two rooted devices running the Android OS and their research was limited to logical acquisition. Next, they provide a detailed description of the file system and its crucial elements, memory block contents and journal log area attributes. Forensic acquisition for the journal log area was summarized in locating the appropriate indicator-block, extracting the following blocks until the metadata one and repeating the same until finding a block signifying the end of the sequence. If the hex values of the block and metadata block are not identical, then the technique had met an unallocated file.

Mylonas et al. (2013) studied the involvement of context-measuring devices (accelerometers, GPS, compasses, etc) of smartphones in forensic investigation procedures. They agreed that this kind of data can be of great importance, but concluded that a special approach is required because of their volatile nature. Afterwards, they proposed a scheme concerning data acquisition from sensors, which ranged from theoretical background of compliance to the standards to practical procedures at the laboratory level. They also proceeded to a classification of sensors, depending on the post-mortem capability of acquisition. Then, they introduced the data acquisition system they developed, named after Themis and enumerated its technical specifications. Security mechanisms on the target devices and bypassing procedures were taken highly into consideration. The authors designed the basic behavior pattern of the system in accordance to legal standards for mobile device forensic (Jansen and Ayers, 2007). Themis consisted of two major parts, the workstation and the mobile agent. Since it was impossible to implement an agent for every mobile OS, the researchers chose Android as the prevailing representative, due to popularity, open source nature and flexibility on creating bootable ROMs. One of the possible uses of Themis, would be data acquisition from a device belonging to a potential suspect of a delinquent action. As a result, they had to obfuscate the installation and functionality of the agent, using methods such as social engineering or fake error messages. While examining the Android sensors security modules, the authors figured out that 12 out of 15 sensors need absolutely no permission when someone wants to gain access to them. Thus, security mechanisms concerning sensors could easily be bypassed. Each time the user accesses a sensor, the agent gets triggered. It acquires and encrypts the needed data and whenever the device is connected in a network and is capable of sending data, it decrypts them and transmits them to the workstation. One of the most significant features of this study is that full experimental support with different kinds of devices was provided and various scenarios were tested. Further analysis had been undertaken concerning the resources consumption and bandwidth requirements of the device running the agent. Themis is a promising and well documented tool that can facilitate forensic investigations, but has to prove its effectiveness in practice. Developing such an agent for other OSs would be an interesting option, given of course that the investigation procedure respects the ethical and legal frameworks. One of the distinctive attributes of this work was the ways the authors used so as to bypass the existing security mechanisms and start mutual communication between the target phone and the forensic workstation. Apart from the classic techniques used, more information about recently discovered possible intrusion methods that could come handy are discussed in Section 4.3.

Guido et al. (2013) used live forensic methods as a means for surveillance for malware activity on Android devices. The selection of the Android platform as the appropriate one was taken after taking into account the OS popularity and the number of applications and malware development. The developed solution had the form of a mainstream Android application, so as to avoid rooting of the devices. It comprised of five modules programmed in python, each one detecting changes in specific parts of the OS, such as bootloader, recovery, file system, deleted files and APK files. Its structure also consists of a remote database that "stores both the collected bytes from the phones and any detector output or logger events (Guido et al., 2013)". The experiment consisted of three rounds of malware infusion on target mobile devices, with many succesful detections, but some weak points as well, such as malware false positive recognizing and inability to detect some deleted entries. Despite the defects, the proposed method is a promising contribution to the MF field.

In his research paper, Grover (2013) developed an application coined *DroidWatch* that performed continuous tracking of events and data flow on an Android device and sent the information to a Web Server. More specifically, he used content providers in order to access data stored from other apps and DroidWatch itself, broadcast receivers so as to track events such as SMS and content observers for database changes. One of the author's policies was to avoid rooting of the device. This had an effect on the acquired data, that were limited comparing to the use case of a rooted target phone, which would have access to more resources. As a result, one of the future aspirations concerning the mechanism was to expand to more data sets, like voice mail logs, and other system information.

The research paper by Vidas et al. (2011) was the motive for Son et al. (2013) to conduct an evaluation on the Recovery Mode method the former proposed, in terms of data integrity preservation. Seven rooted Samsung devices running the Android OS took part as a sample. The results from the use of JTAG method, which was used whenever it was possible, served as a comparison vector to the Recovery Mode. First of all, a section was dedicated on the acceptable practices during the data acquisition phase in Recovery Mode. The authors introduced a flowchart related to the steps taken during the acquisition procedure. They emphasized on the importance of using the appropriate bootloader for each device and also mentioned some issues with encrypted ones. After flashing the device with the modified bootloader, it was able to start functioning in Recovery Mode and they were able to acquire the needed data by using the ADB command. The next step of the procedure concerned reverting the device to the last operating state before the bootloader flashing. Unlikely to other research papers, they highlighted on actions that should have been taken into consideration during the restoration process, for example the prohibition of interaction with the menu elements in Recovery Mode and the USB cable separation from the device before battery removal. Additionally, *Android Extractor*, a C++ GUI was developed so as to conduct the data extraction tasks and check the integrity of the method. Finally, the hash values of the data partition that were extracted in both cases was calculated and proved to be equal, assuming that integrity was preserved. Nevertheless, the authors claimed that the same experiment has to be undertaken in devices belonging to different brands.

In a very recent work, Muller and Spreitzenbarth (2013) investigate innovative techniques in an effort to assess how much valuable information can be extracted from encrypted Android smartphones. The authors performed a *cold boot* attack by freezing the device so as to gain physical access to the RAM and acquire precious information, such as encryption keys or personal data. Additionally, the authors presented a a recovery tool named after "Frost" to be used as a means of acquisition. The method proposed presents an important limitation, well known in the forensic ecosystem. The user partition gets wiped out when the device bootloader unlocks. Still, it is the first work to perform a successful and effective cold boot attack on smartphones and the encryption barrier seems to start being bypassed. Promising research efforts like this one should evolve and taken highly into consideration, to ameliorate the techniques presented and to simultaneously tackle any limitations.

3.3. Blackberry Forensics

Blackberry is a relatively popular OS. The structure of the OS itself made the IPD file the first place for a potential researcher to search for significant data. An early attempt to acquire the contents of the IPD file in terms of backup retrieval was carried out by the development of ipddump (ipddump, 2011), a Java application first released in 2008. It was initially capable of solely extracting SMS (Fairbanks et al., 2009), but newer versions are claimed to support acquisition of other types of user data as well. The last stable version was released in 2009, while a release candidate was available in 2011. Its function was summarized to parsing, "extracting and exporting all types of records into customized open text formats as well to edit records like service books and contacts" (ipddump, 2011).

Another IPD acquisition-oriented work was suggested by Fairbanks et al. (2009). The authors implemented a python tool that parses the IPD file upon user request for specific resources. The range of data sources the user/investigator would be capable to choose among memo, contacts and SMS databases. The tool functionality was divided in two parts. During the first, it proceeded to data acquisition from the databases existing in the IPD file and transformation of them to a "more contiguous format". (Fairbanks et al., 2009). The second part triggered a different parsing mechanism for each one of the data sources mentioned previously. Afterwards, they tested the tool on a Blackberry 7290 device, with satisfactory results but did not provide the potential readers with enough data.

Similarly to other kinds of devices, the acquisition procedure used to be limited to the official Blackberry Desktop Manager (BDM) via the IPD file generation (Sasidharan and Thomas, 2011). Apart from other features, BDM also serves as a backup manager, which returns an IPD file to the user and then ABC Amber BlackBerry Converter interprets it to a readable format. The importance of using BDM so as to create an IPD file was highlighted in (Casey, 2009), since not only it can work as a means of comparison to other extracted versions, but also encourage more in-depth analysis with the formerly mentioned tools, like ABC Amber. The work in (Sasidharan and Thomas, 2011) proposed a different approach, oriented to database acquisition with the use of an a software agent. After exposing the most important rules that preserve the legitimacy of the forensic methods (ACPO, 2007), they compared them to their method and summarized the deviation between each one. For example, the use of an agent itself provoked changes to the state of the device, since data were modified. This problem was not only spotted on Blackberry, but on every kind of OS. Moreover, the lack of official documentation or technical design outline provided by the vendor obligates forensic analysts to start research from scratch, with great risks. The method the authors proposed was a type of logical acquisition, since it was a program interacting and extracting the databases within the Blackberry file system. In terms of this research, they implemented Blackberry Acquisition and Analysis Tool (BAAT), that was claimed to offer a complete forensic solution of data retrieval and analysis. It is notable though that they had not mentioned any acquisition types or classifications. The .cod agent was based on Client-Server architecture and developed in BlackBerry Java Development Environment (JDE). Apart from accessing databases, it was capable of creating a communication and data exchange stream between the target device and the computer it was connected to. Moreover, the returned results did not need the converter in order to be decoded. Data integrity was preserved by the use of MD5 hash functions for the retrieved data. The next step consisted of analyzing the extracted data with BAAT. It extracted acquired data concerning the target device on .html format. The main challenge they faced does not differ from the ones of other researchers and concerns the means a mobile device can comply to the existing standards of no modification after seizing a device.

Belenko (2012) focused his research on overcoming the possible impediments upon a Blackberry forensic acquisition, which are the device password protection and file encryption. He introduced the use of *Elcomsoft Phone Password Breaker*, a tool that can recover the device password, even without interacting directly with the device and taking advantage of a file used in the external storage media. In order to be functional, the tool "requires Media Card encryption to be switched on and set to either "Security Password" or "Device Password" mode" (Belenko, 2011). Similarly, the device key can be helpful in the decryption process of encrypted files, but there are occasions that other keys are used, such as the *WhatsApp* database in the SD card. Such entities can be decrypted by device dump.

Marzougy et al. (2013) discussed the logical acquisition of the .bbb file produced by a *Playbook* tablet device. However, the authors argue that, due to the independent nature of the tablet version of the OS, assumptions based on the mobile version may not be applicable to the specific one. Before performing any type of automatic acquisition, the researchers proceeded to a manual examination of the device. After some experimental use of the device for basic functions, BDM was used aiming to produce the logical image, which consisted of three *.tar* files (Media, Setting and App), as well as an *Manifest.xml* type-of file. The information they had been able to extract were varying from system information to application elements and user data. As expected, the method had not been able to retrieve deleted entries, but it was able to locate a list with the form of the file system before some files were deleted. The authors although stated that the research was considered incomplete, due to the fact that only one device was tested. They finally mentioned that further research has to consider the *4G/LTE* enabled version of the device as well.

3.4. iOS Forensics

Zdziarski (Zdziarski, 2008) achieved the breakthrough of implementing a physical acquisition technique, especially designed for the iOS. There are no other similar attempts in literature at least for the time being. It was generally claimed that even the jailbreak technique he used was superior to other widespread ones (Hoog and Gaffaney, 2009). Specifically, the unique feature of the method focused on changing an amount of data in the system partition but left the user data partition untouched. In any case, ideal state of no data modified had not been achieved; a forensically sound image of the user data though had been a breakthrough. Then, he booted the test device with a *recovery toolkit* (Zdziarski, 2008), which contained the essential software enabling him to obtain a bitwise copy of the memory image. Another notable feature was the utilization of SSH in the recovery toolkit for establishing an encrypted bridge between the device and the workstation. Bypassing the protection code was accomplished by the installation of the iPhone Utility Client (iPHUC) on the workstation. Finally, file carvers, SQLite Database Software and other recovery/viewing programs were used to convert the acquired image to human interpretable format. Zdziarski contributed a major advance in the iOS forensics field. The research though needs to be continued, since new versions of the OS are implemented and previous techniques may have been already outdated.

Hoog and Gaffaney (Hoog and Gaffaney, 2009) set the basis of forensic investigation for the iOS, with a detailed outline of the state-of-the-art and applicable methods till that point. After presenting the most important technical attributes, they classified acquisition methods (manual, physical and logical). Commercial forensic tools performing physical, logical or both types of acquisition took place in the survey, as well as Zdziarski's physical dd method (Zdziarski, 2008). Data extraction was carried out by the iTunes backup feature, with the automatic synchronization option deselected. The test device, an iPhone 3GS (2.2 firmware), not having been through a jailbreak process was filled with any kinds of data that can reveal user interaction with the phone, as in a real case study. The researchers then implemented an evaluation method for each acquisition procedure, consisting of certain factors, such as ease of installation and use, acquired data integrity, etc. Each of the factors contributed in a different scale to the final result, according to its relevance to the acquisition procedure. When the evaluation was completed, Zdziarski's method gathered the highest score. Research results have shown that different forensic tools lead to different acquired data quantity and quality, according to their characteristics.

Morrissey (Morrissey, 2010) had also discussed logical acquisition on iPhone devices by the use of the iTunes backup feature. The research was applicable to iOS versions prior to 4 that had not been jailbroken. The acquisition procedure was enhanced by the use of mdhelper, a command line-utility specialized on data parsing. Mdhelper was not considered an essential add-on; on the other hand, it was able to facilitate investigators navigate through retrieved data at low time cost. Automatic synchronization had to be disabled since the beginning of the procedure in order to reassure the forensic soundness of the retrieved data sets (Jansen and Ayers, 2007). After the acquisition point, evaluation testing was similar to the one performed by Hoog and Gaffaney (Hoog and Gaffaney, 2009). However, contrary to Hoog and Gaffaney, evaluation results were calculated only from the amount and quality of retrieved files. Morrissey concluded with low tempered attitude towards forensic tools, implying the need for more efficient techniques.

As already pointed out, one of the prevailing acquisition techniques concerning forensic acquisition from iOS devices is obtaining a logical backup via the iTunes backup feature. This approach was also studied by the work in (Bader and Baggili, 2010). The test device was an iPhone 3GS, not having been through a jailbreak procedure. They claimed that, even if there exist a few physical acquisition techniques, iPhone devices are mainly examined through logical acquisition. After ensuring that conditions of the research were compliant to forensic standards (Jansen and Ayers, 2007), they connected the device to two workstations (Windows and Mac) and initialized the backup procedure without triggering the synchronizing option. This is because if the synchronizing feature have been powered on, data on the phone would be altered and could not be considered an admissible piece of evidence (Jansen and Ayers, 2007). On the other hand, some modification to the device data had been traced, since activation of a write-blocker failed when the computer was connected to the workstation via USB. “The acquired backup was parsed and viewed using specific tools, such as pList editor, SQLite Database Browser” (Bader and Baggili, 2010) and other file parsing utilities. A unique trait present in that research was the detailed outline of the whole examination procedure imposing the framework and its limitations. Some limitations had come to surface during the experimental procedure the authors followed. iTunes versions prior to 8.2 were unable to interact with the iPhone 3Gs device. This assumption pulls a trigger to a general discipline concerning attitude towards experiments. The more approaches are examined, the more detailed and appropriately oriented a research can be. Since logical acquisition leads to extraction of a vast amount of data, isolating and analyzing the most important for the investigation can become demanding.

Both workstations backup folders were compared to each other. The same sequence of alphanumeric characters appeared to be the folder name for the two of them, leading the authors to the conclusion that the file name was a hash function output, unique for each acquisition deriving from the same device, with the same timestamp. The backup folder contained several subfolders, each named after a hash function value. “Backup data is stored in three file formats, pList files which store data in plaintext format, mddata files which store data in a raw binary format and mdinfo files which store encoded metadata of the corresponding mddata files” (Bader and Baggili, 2010). Special software was used for decoding the files mentioned above, such as pList and SQL editors. They also made a small reference to manual acquisition with terminal commands, as an additional data source and means of comparison. Next, they made a classification concerning the data types acquired. Parsers were used to convert binary files to its original state. Finally, they proposed the development of open-source software in charge of handling data from pLists, databases and printing the appropriate reports. Another challenge they picked out was the acquisition methods for password protected devices, or groups of encrypted data.

Husain et al. (2011) after expressing a general disappointment towards commercial forensic tools performance and disadvantages of individual acquisition methods (Zdziarski, 2008) or acquisition techniques involving the *Jailbreak* procedure, proposed a framework for iPhone forensic investigations, consisting of three phases; data acquisition, data analysis and data reporting. The presented procedure does not have notable differences to other studies in the field, but could be proposed as a general framework. Once more, acquisition had been performed via the iTunes backup utility, but the researchers did not mention sync deactivation. Decoding retrieved data was carried out through the use of file parsers and plist editors. For bypassing security codes, they proposed the seizure of the device that may belong to the suspect or victim. This assumption though isn't always applicable, since the investigators might not discover any workstations or even the suspect might not posses any computing devices. The arguments they present are not referring to a general condition, so the proposal is rather insufficient to become an official framework for investigations.

Social networks have become the center of attention, since they gain more and more subscribers every single day. Data deriving from mobile versions of social media can be an important source of information for investigators. The authors in (Jung et al., 2011) discussed social media data exported from mobile versions of social networks for the ver. 3.x and 4.x of the iOS. Even if their research is restricted within the limits of a country, the sample can be representative since some of the social networking applications used, are popular worldwide.

Before beginning the experimental procedure, the researchers split the acquisition procedure to two main categories; the first concerned devices security mechanisms of which had been bypassed by the use of the jailbreak technique, while the other was referring to devices that had not been through any kind of change. Less attention was given to the first category, since the authors skipped the acquisition method used. On the other hand, they gave pretty much attention to the second category, where they used one of the common approaches for acquisition (Bader and Baggili, 2010), the backup feature of iTunes. Extracted backup data are the same in both OS versions, but metadata have a different format. For instance, in ver. 3.x, the extracted data file has the .mddata extension and the .mdinfo is created for the metadata. In ver. 4.x, "all information on the backup file is saved as a pair of files, manifest.mbdx and manifest.mbdb" (Jung et al., 2011).

The next step in the survey was to gather and compare the retrieved results from the social networking applications, as well as the paths to data. Data deriving from backup acquisition were stored under a hash value, while those extracted from the jailbroken devices were stored directly with their file name and type. Next, the information of major forensic importance, such as multimedia images, user-driven social media attributes and geolocation data were categorized. A quite interesting aspect concerning multimedia files (photos and videos) was spotted, since an iPhone device by default relates pictures taken from its camera to the GPS coordinates of the place it was taken. This information could be extracted separately, but this feature couldn't be used in social media applications, due to the fact that the OS itself creates another picture folder, where photos to be uploaded are stored without the geolocation data available. In the end, they focused on behavioral analysis of data manipulation concerning each application and concluded that all of them have different structural attributes. Information concerning temporary files from Facebook could only be retrieved from Jailbroken devices. The research gives food for thought for further experiments in the field of social networking applications.

Within their research, (Tso et al., 2012) determined the iTunes backup utility as a prevailing method for logical acquisition from devices running iOS. First off, they detail on the features that provide forensic importance to iTunes. One interesting aspect was the explanation on how application data reside in, and can be collected from an iPhone handset. Technical details had been analyzed in depth; however, there was no trace of compliance to MF standardization. For example, the option of de-activating synchronization was not even mentioned. After that, they enumerated the arguments concerning data deriving from social network and chatting applications (Facebook, WhatsApp Messenger, Skype, Windows Live Messenger and Viber) use, which was the initial scope of the research. The test device was an iPhone 4, running iOS ver. 4.3.5. The experiment consisted of two phases. The first concerned data acquisition after apps installation, while the second after deletion. All the applications tested, apart from Facebook, stored additional data in the backup folders, which could be easily decoded by the use of pList editors and an SQLite browser. Results were satisfying the needs of a forensic investigation, but there were still some parts needing extra attention, such as those concerned with encrypted and unallocated data. The case study of a device being through jailbreak has been neglected in this research as well.

Recently, Arrifin et al. (2013) presented an acquisition method of deleted image files after interaction with the *iOS journaling system* "from an Hierarchical File System (HFS) Plus volume in an iOS device". They highlighted the complexity of unallocated data retrieval procedures and the importance of data encryption during them, as these were the two main directions of their research. Afterwards, they designed the scheme of the implemented technique, containing the four stages (identification – preservation – analysis – presentation) of the McKemmish (1999) framework. The proposed method was implemented on a device that had been through jailbreak and used a *customized RAM disk* that was loaded in the device RAM; as a result, it affected the flash memory the less possible. Then, a physical image was acquired and the authors applied the following technique for the deleted files retrieval. Since the deleted files existed in the journal file but not in the catalog file, it was relatively easy to track them. An encrypted copy of them was extracted. The file system key (EMF) was employed for encrypting structural OS elements, such as journal and catalog entities, metadata and file system particles, whereas user data used to be encrypted by the AES engine. Taking into consideration that the acquired files are split in blocks within the journal, "the EMF and per-file (AES encrypted) keys have to be used respectively"(Arrifin et al., 2013). The technique was successful in the two target devices (3GS and 4) the researchers tested and they aspired the study to be extended to newer versions.

3.5. Maemo Forensics

Lohrum (Lohrum, 2012) studied forensic acquisition from the Nokia N900 smartphone running the Maemo OS. Maemo isn't a widespread OS. Hence, forensic studies on the field are limited to few resources only, but physical and logical (mentioned as triage) acquisition methods are present as well. The author made an introduction to the technical details of the Maemo OS, focusing on its Linux origin. Features present on the phone that were considered innovative by the time it was released in the market were the Unix terminal, VoIP communication and game consoles emulators.

Since there is poor documentation concerning forensic acquisition in Maemo devices, the author's first goal was to find the places data of major importance were residing in. The procedure he followed didn't deviate much from methodologies used for other mobile platforms. It consisted of the following steps: manually entering data and interacting with the device, acquiring a bitwise physical copy of the phone memory image, decoding them via a forensic tool, comparing the data to the original ones and finally performing a limited triage extraction onto a microSD card (Lohrum, 2012). Before proceeding to the physical acquisition, the researcher eliminated possible interactions of the phone with every potential network connection. However, for achieving the physical acquisition procedure successfully by the use of the dd command, he had to root the target device. A unique feature of the acquisition process was that the image was transmitted to the workstation computer by an SSH tunnel. Another characteristic of the specific handset that could facilitate forensic research was the fact that user data, OS and swap space were situated in different partitions. This way, rooting could affect only one of them, enabling forensic soundness preservation to the not affected ones. After locating the most important data within the file system, he conducted a logical acquisition by an automated script, coined *N900TriageExtraction.sh*, which contained copying commands. In the end of the acquisition, the script was deleted.

3.6. Shanzhai Forensics

He et al. (2012) published a holistic review on such counterfeit devices, consisting of simultaneous examination of both technical elements and forensic techniques. They performed physical acquisition with bootloaders on a Shanzhai device, with a MediaTek MT6235 processor and a 132 MB NAND flash chip embedded.

Meanwhile, the authors compared the results of the acquisition they performed to a previous study on the field (Fang et al., 2012a), which was dealing with forensic acquisition from devices of the same manufacturer equiped with a MediaTek6253 processor and a NOR flash chip. He et al. concluded that different kinds of memory images present a completely different outcome. Acquisition "snapshots" were numerous for the device carrying a NOR memory chip while the one with a NAND memory chip presented none. He et al. (2012) successfully retrieved allocated files concerning contacts, text messages and calls but unallocated ones were overwritten and replaced by new entries. Since the quality, structure and technological details of such devices is obscure, there is a compelling need for further and substantial research, so as to acquire accurate results.

3.7. Symbian Forensics

The work by (Mokhonoana and Olivier, 2007) discussed the development of an on-phone forensic logical acquisition tool for the Symbian OS (V. 7), which is based on the dd technique on portable devices running Linux (Mokhonoana and Olivier, 2007). At first, they made an introduction to Symbian OS characteristics and then classified potential acquisition methods. Their approach consists of manual acquisition, use of forensic tools, logical acquisition including a connection agent, physical acquisition and data acquired from service providers. Use of forensic tools as a separate category is disputed, since they serve as automated solutions in order to interpret logical and physical acquisition results to human readable format. On the other hand, data retrieval from a phone service provider was beyond the scope of this paper. It would be more appropriate, if its use was complementary to the results deriving from manual, physical and logical acquisition tasks, for guaranteeing data integrity. However, the classification provided by the authors, depending on the period of time the article was written, is quite accurate, since it was only then that mobile devices started becoming complicated and needing different kinds of acquisition, apart from SIM module and phone service provider ones.

Moreover, they clarify that when interaction of a forensic tool with the target mobile phone is kept at a low scale, then it is more likely the evidence deriving from it to be admissible upon court. After enumerating the possible ways of installation concerning the on-phone tool, they came into the conclusion that the way that causes less data differentiation is saving the tool installer on an external memory card and then placing it inside the target device. They characteristically state "Even though it may change certain parts of the OS, the changes are very little compared with placing an entire installer which still has to be extracted" (Mokhonoana and Olivier, 2007). This approach was the optimum, preventing alterations of data, but presented other disadvantages. Specifically, acquired data were stored in the same memory card carrying the application installation file, so the acquisition destination was not forensically sound.

Furthermore, they used recognizers, which "are written as plugins to the MIME Recognizer Framework and are scanned for and loaded during operating system startup" (Morris, 2006). Thus, their tool was able to start when the phone was booting, and thus avoiding further interaction. Choosing between Java and development of a native application using C++ Symbian programming language was another challenge faced. The option selected was the native application one, since it gained access to lower levels of the OS by default; this way, acquisition of bigger portions of data was possible. Another negative trait of the tool was that it was unable to acquire data from applications being executed at the same time, since it couldn't handle the processes running. As a result, data of high forensic importance could not be acquired, such as call logs and contact lists (CallLog.dat and Contacts.cdb files). Data retrieved were mainly modified and created by the users and handled by applications. There was no trace of data manipulated by the OS or other structural elements. Despite the fact that the method presented had major issues waiting to be solved, it became a source for further research.

Breeuwsma et al. (2007) 's object of research was physical acquisition of flash memory from different types of embedded systems, mobile devices included. They firstly introduced the characteristics of physical acquisition techniques (chip-off, JTAG, pseudo-physical) and enumerated the advantages and disadvantages of each. Afterwards, they highlighted the importance of “placing the sectors of data as used by the high level file system before any kind of file system analysis (Breeuwsma et al., 2007). Moreover, they developed and used a python script, *ListLSN* that facilitated the reconstruction of memory blocks by checking and sorting the logical sector numbers (LSNs). The last part of the study concerned the experimental process of applying the proposed methods on several Symbian devices. It was notable that they didn't show similar behavior, mainly during the file reconstruction procedure. Lastly, they focused on the need for further research on the subject, so as to succeed compliance to newer versions of Symbian devices.

After a brief but fruitful presentation of the state-of-the-art concerning both smartphone usage spread and forensics standardization, Distefano and Me (2008) proposed another on-phone logical acquisition tool for devices running the Symbian OS v. 8 and older. Even during a period when acquisition was conflicting with the use of commercial forensic tools and performing a bit-by-bit acquisition of the internal memory of the device was considered an impossible task, they achieved to retrieve the complete Symbian file system. MIAT (Memory Internal Acquisition Tool), the tool they developed, is considered the evolution of Mokhonoana and Olivier's equivalent (Mokhonoana and Olivier, 2007). However, taking into consideration that the versions tested were older than 9, there was no security mechanism to bypass. During the acquisition process, the tool opened and copied on read-only mode each entry it stumbled upon while traversing the file system tree.

The authors also managed to correct previous defects, since the memory card inserted into the device for performing the acquisition was forensically sound and divided into two different partitions; one containing the tool application installer and the other destined for the acquired data. Moreover, the acquisition procedure was enhanced with the use of a hash function in an effort to assure that the data retrieved were identical to the original. They also conducted experiments comparing MIAT to Paraben Device Seizure, a commercial forensic tool, and P3nfs, an application which is not a forensic one, but was claimed to mount Symbian file system into Linux file system (Distefano and Me, 2008). After completing the experimental process, it was assumed that MIAT showed many advantages compared to the other tools. Parallel support for simultaneous examinations reduced the time needed in order to complete the investigation. Moreover, files that succumbed to changes after the use of MIAT were fewer than the ones Paraben tool changed.

Last but not least, the extracted data size was almost the same for both tools. MIAT was properly documented and researchers set the proper base for future research, support of ver. 9 and ameliorations. Another accomplishment noted was the further development in order to give the tool a form that could support real-time investigations. It is obligatory to mention that there was no reference concerning the types of data acquired. This fact would make the research procedure more complete and clarifying.

Yu et al. (2009) noted the forensic techniques incompatibilities between smartphones and ones running the Symbian OS and claimed that the latter call for a totally separate approach. Firstly, they conducted a research concerning the investigation models applicable to the DF field. Before proposing a similar model for the Symbian OS, they exposed the state-of-art for the specific kind of phones. New security mechanisms, starting from ver. 9.0 (capabilities) and their effects on a potential forensic investigation were discussed.

The proposed model consists of 5 stages. The first one concerns acquisition of the version and model without directly interacting with the OS. It is also checked whether the mobile device is protected by security mechanisms or not. If security mechanisms are present, then investigators use a protocol or a hardware approach concerning physical acquisition (via remote connect and response protocols or JTAG connection respectively). The techniques mentioned are the ones interacting less with the security mechanisms. Their only interference includes gaining access to the *swipolicy.ini* file and then root privileges on the device. If security mechanisms are not present, then the use of every forensic tool is applicable. Poor documentation concerning acquisition types, other than a reference to Mokhonoana and Olivier‘s tool (Mokhonoana and Olivier, 2007), is reducing the reliability of the model. The next steps apply to extracted data analysis and dissemination of the result. Another negative aspect is the lack of documentation concerning standards in the field. Nevertheless, if extended, their (Yu et al., 2009) process model can be useful for future investigations.

Savoldi and Gubian (2009) presented a review concerning specifically forensic acquisition focusing on security features of the Symbian S60 version of the OS, the ones belonging to previous versions. Similarly to other works, they commented on the functionality complexity of smartphones and the information treasury they might contain. After an introduction to the OS and file system architecture, they dedicated a big part of their work to the security traits of the OS. The security features review was focusing on the different layers of access and the authorization procedures applications had to use in order to gain access to protected areas of the system. In order to bypass those security impediments, the writers proposed modifications on the *swpolicy.ini* file. This way, they would be able to gain access to previously restricted domains, with the less device state alteration as possible. Afterwards, they enumerated the potential acquisition types that included manual, logical and physical methods. Finally, they presented experimental results for a device running the S60 version of the OS, stating that (given the time the research was conducted) only physical acquisition was able to recover a bitwise copy of the flash memory without bypassing any security mechanisms.

Pooters (2010) created Symbian Memory Imaging Tool (SMIT), which is mentioned to be the first on-phone tool to create linear bitwise copies of the internal flash memory. SMIT, designed for Nokia cellphones, is mainly based on a hybrid method consisting of logical acquisition techniques and boot loader class methods. It also makes use of the Symbian OS API in order to gain access to the file system of the device. The researcher claims that the developed tool is able to recover hidden data from slack and unallocated areas of the memory, due to the support provided for low-level system calls. Since SMIT is an application installed in the mobile device and used for live forensic purposes, it alters its original state. As a result, the researcher adopts techniques for the application to be able to comply with the NIST Guidelines on Cell Phone Forensics (Jansen and Ayers, 2007), such as use of MD5 and SHA-1 hash functions and reducing write traces. The application it is claimed to be compatible with Symbian Ver. 8.1 and newer. Tools accessing the lower level API instead of the file server one are automatically acquiring more privileges over drive manipulation and access. It is notable that SMIT acquisition was more efficient on the test devices running Symbian 9.0 than on the 8.1 ones. On 8.1 ones, the only partition recovered concerned system data, while 9.0 devices returned both system and user data partitions.

Pooters claims that further research needs to be conducted for ver. 8.1, but taking into consideration the current market share, priority should be given to other mobile platforms. Rooting, is also present in Symbian cell phones. For installing SMIT properly, it was obligatory to use a capability hack. Bear in mind that capabilities are a mechanism to control the actions an application is allowed to perform on the OS (Pooters, 2010). In this case, Pooters used the HelloOx (Team HelloOx, 2013) modification, a hack destined for rooting Nokia mobile phones. After HelloOx competing drive mapping, it extracts itself directly on the virtual ROM drive, proceeds to root certificates installation and patches memory pages to prevent the capabilities mechanism from functioning. Nevertheless, compliance of HelloOx and other third party rooting applications with forensically sound disciplines is still controversial (Jansen and Ayers, 2007).

Moreover, the author proceeds to a brief presentation concerning analysis techniques of retrieved flash memory images. Then, he exposes data types of forensic interest acquired from the device, varying from application data to to those edited and deleted by the end-user. Developing a tool like SMIT was a big step in the Symbian forensics field, since none of its predecessors, such as MIAT (Distefano and Me, 2008) were capable of retrieving a bit–by–bit copy of the file system. Moreover, it had the unique trait of deleted data retrieval, which is a weak point concerning logical acquisition based tools. Access to the low-level OS API calls allows the advantages of logical acquisition to combine partially with physical acquisition traits without facing the circuit destruction challenge. Due to the hybrid nature of SMIT, Pooters provided a slightly different classification of acquisition methods. He divided physical acquisition in two categories, Chip extraction and JTAG, while adding the level of Bootloaders between physical and Logical extraction. The current research paper is one to also stand against the obstacle of rooting without altering the original state of data on the device, but the use of hash functions preserves data integrity. Further research needs to be conducted for the retrieval of files altered by the OS itself, such as GPS and PIM activity, but also for the ones residing in databases. Expanding functionality of SMIT to other brands apart from Nokia would be a positive outcome as well.

Thing and Tan (Thing and Tan, 2012) went further through the Symbian forensics field. They mainly focused on retrieval of privacy protected data on smartphones running the Symbian OS. The two devices used in their experiments were running OS ver. 9.4 S60 5th edition and 9.3 3rd edition. Their study concerned acquisition of allocated and deleted SMS from the internal memory of the devices. No acquisition type was mentioned, but since their study was concerning protected data, it could be applicable to both physical and logical methods.

At first, they exposed the current situation concerning SMS recovery tools and concluded that their area of influence had been quite limited and effective in certain particles, such as the SIM module.

Next, they claimed that the main obstacle in internal memory acquisition is the existence of default security mechanisms, not only for the Symbian, but for other mobile platforms as well. As versions of OSs evolved, it is becoming more difficult to penetrate the security locks that preserve the integrity and availability of sensitive data. As a result, they implemented a technique to bypass the AllFiles capability that leads to unlimited access to the Symbian filesystem. The researchers managed to create a sub-directory ("\sys\bin") under any directory in the phone, place executable files in it, and then map it to a new drive letter, thus effectively putting these executable files into the valid executable path. They also configured a Symbian Authority Certificate and integrated it in the device with the aid of the mapdrive API, by triggering the *Symbian Certificate Store*. By doing so, they have been capable of allowing their tool to acquire permissions of executing different kinds of commands while leaving other security mechanisms activated and providing them from intervening to its functionality. This is considered a revolutionary technique, since its predecessor capability hacks like HelloOx different versions (Pooters, 2010) totally disabled security mechanisms, making the device a potential target for malicious attacks. Since security locks had been bypassed, they were able to locate the previously invisible path to the folder containing the unallocated and active SMS within the internal memory, which is the \Private\1000484b\Mail2 one. They figured out that each file entry inside the folder referred to an allocated SMS. Moreover, they implemented an algorithm in order to perceive properly the actual packet and message length (Thing and Tan, 2012). Acquiring deleted SMS from the index file was a slightly different procedure. The fact that every deleted message started with a certain sequence of digits and alphanumeric characters facilitated their work. Along with a maximum number of bytes (64) that had been noted for every deleted entry and observation of different kinds of indices, messages up to that bytes limit could be partially or fully recovered. Researchers proposed future expansion of the same experiment to other types of protected data, such as MMS, e-mails notes, etc.

Thing and Chua (2012) developed a physical acquisition tool for Symbian phones, entitled Symbian Acquisition Tool (SAT), written in Symbian C++. The test subject phone was a Nokia N97 running the S60 5th edition. The tool is composed of an *Acquisition Program*, a *Memory Extractor Module*, a *Hash Generation Module* and a *File Compression Module* (Thing and Chua, 2012). The Acquisition Program triggered the Memory Extraction Module, which stored the bitwise copy of the memory image at a removable storage media. The Hash Generation Module was necessary, since it computed the SHA-1 value for future verification tasks. The File Compression Module was used for reducing the image copy size in order to fit to the removable storage media. The next step concerned filling the device memory with different kind of data, deleting and/or adding new ones and extracting the results concerning fragmentation as well. It seemed that fragmentation levels were higher when many file modifications were taking place. Research would be extended with other models except from Nokia ones and other OS versions.

3.8. WebOS Forensics

Casey et al. (2011) examined forensic acquisition related to the webOS OS. Apart from presenting an outline of the technical characteristics, the authors shared the results of two case studies concerning two famous device carriers of the specific OS. The main challenge was spotted during the performance of forensic acquisition on the system partition. Several tools and traditional approaches have been implemented to support forensics on FAT32 filesystems. The researchers proposed two alternative methods of acquiring an image of the system partition, which required a different procedure, since its file system is ext3. Both techniques utilized the popular dd command with some command variations (novaterm and novacom). The former was the prevailing one, since it did not cause data modification. The empirical investigation revealed that a real treasury of existing, but also unallocated data could be retrieved.

3.9. Windows Mobile Forensics

Klaver's work (Klaver, 2010) has been an influence to many future researchers since not only it introduced revolutionary techniques in the MF field, but also discussed the most significant parts of the hardware and software related to them. His work concerned the study of physical acquisition mechanisms on smartphones incorporating the Windows Mobile OS, ver. 6.0. The most significant attributes of forensic importance were the bootloaders and the RAM heap present in all the Windows Mobile devices. The fundamental role of bootloaders is system booting. In the forensic ecosystem, they are also used for extracting a physical binary image of the memory of the device with effective or fruitless outcome. Despite the fact that bootloaders exist in smartphones carrying other kinds of OSs, only Klaver gave a detailed description of their utilities and applications in the forensic science. He also mentioned that if a bootloader is prevented from being accessed, many problems concerning a potential forensic extraction might occur. Moreover, he argued that the RAM heap is a treasury of unallocated data or data deriving from interaction with applications, since most buffers reside there. Just like other researchers, he took highly into consideration and tried his research methods to comply with the admissible forensic practices upon court (Jansen and Ayers, 2007) and assumed that a sound investigation requires inactive connections and heap alternations on the target handset. After categorizing acquisition methods to logical and physical ones, he reached the conclusion that even though physical acquisition is more effective it is dangerous for both the mobile phone and data stored in it.

On the other hand, he claimed that logical acquisition can become insufficient if access to certain areas of the memory is prohibited and forensic soundness can become compromised if bypassing modifications are used. He then presented a pseudo-physical acquisition technique, as a mix of characteristics from both types of acquisitions. The actions set consisted of a bitwise copy of the flash memory image obtained through ActiveSync. This could be achieved through the use of a "dedicated dll loaded into the system under investigation, thus overwriting RAM and possibly flash memory" (Klaver, 2010). If this had been a physical acquisition procedure, the outcome would be at flash hardware level. Since this technique has been followed, the produced outcome would belong to the level of the file system. This attribute prevented unallocated data from being retrieved and contributed to the hybrid nature of the method.

A presentation of the pseudo-physical acquisition method wouldn't be complete without an evaluation of the tools designed for the specific purpose. Both of the tools utilized interacted with the target device in a way that data on it had not been affected. This could be achieved either by remote handling or by the use of removable storage media. An interesting approach was provided when the author presented the file system reconstruction procedure based on memory pages in pseudocode mode. Pseudocode was also for describing acquisition procedures from files with high forensic importance, such as *cemail.vol* and *pim.vol*, with the use of proper libraries and dlls. Yet another unique feature offered in the context of this research, is the use of a Python script, namely *cedbexplorer.py*, as a potential algorithm to retrieve standalone records within a database. Once a record was retrieved, it was decompressed; its MD5 hash was calculated and then compared to the MD5 hash fingerprint deriving from xpdumpcedb.exe. Preliminary efforts were made by the author toward understanding the way a program is interacting with the heap and leaving traces on it. Hence, Klaver implemented another Python script, *heapdigger.py*, which searched the heap marker in an image and decoded the heap headers and subsequent heap items. This research served as food for thought for many future ones and methods employed need to be updated accordingly to accomodate newer versions of the OS.

A more limited version of acquisition and data analysis based on Klaver's work was proposed by (Casey et al., 2010) and applied only to non-password protected devices. The authors adopted the same pseudo-physical acquisition technique as Klaver and made use of one of the tools he proposed. Difficulties in retrieving deleted data were taken highly into consideration, without focusing on the acquisition methods related to each kind. Obstacles that may arise are due to failures in reconstructing the file system, as well as the fact that a certain amount of devices running the Windows Mobile OS replace the content of deleted files with a sequence of 0xFF. Similarly to other researchers in the field, they concluded that forensically valuable information can be retrieved from the cemail.vol and pim.vol files. By presenting a detailed structure of the cemail.vol file, they emphasize on the importance of the data provided within it. They also promoted the use of a hex editor to retrieve hidden deleted data.

Another place susceptible for containing important information is the system registry, since there could be found details of the configuration and use of a device (Casey et al., 2010). The researchers mainly focus on the importance of acquired data during an investigation. An interesting aspect was presented in the end of the research, concerning the remote execution of a piece of code sending data to a third party entity, which observes the user's actions. Such a task may not be visible upon the task manager or perceived by the average user, but it can still run on memory or be detectable by the RAPI tools commands (Klaver, 2010). In this direction, Casey et al. (2010) propose an implementation of discovering such processes by a program entitled MobileSpy, which had been able to detect this kind of changes, but the documentation provided would have been more satisfying if tested on more than one devices. Casey et al. (2010)'s work completes the experimental overview provided in Klaver's and can be further used for a plethora of examples.

The work introduced by Rehault (Rehault, 2010) was a case study of a pseudo-physical acquisition method implemented strictly for the HTC TyTnll device running Windows Mobile ver. 6.0. The author used a modified bootloader and acquired a bitwise copy of the flash memory. As far as we are aware of, this is the only work providing an analytic overview of the components and functionality of a bootloader. Afterwards, the author proceeded with the file system and registry reconstruction, while well known carving tools were used to retrieve the contents of databases, such as *cemail.vol* and unallocated files. The acquisition method proposed showed the major impediment of inflexibility, since devices even from the same vendor cannot be compatible with one unified bootloader type.

Grispos et al. (2011) studied data acquisition concerning the applicable forensic techniques on smartphones running the Windows mobile OS. This work does not propose a new method, but makes a comparison between the ones already developed. The lack of a standardization structure within forensic acquisition processes seemed to be the major challenge, since they had to create a hypothesis and adapt it to the tools they used from scratch. The study began with a classification of acquisition methods to physical and logical and a presentation of the tools used during each one. The test smartphone was running the 6.1 ver. and wasn't supplied with an external storage card, since the authors claimed that it was unnecessary, due to previous studies conducted on the subject. The device succumbed to manual, physical and logical acquisition procedures. Before any kind of acquisition took place, ActiveSync was enabled. As far as it concerns physical acquisition, the researchers used a pseudo-physical technique, implemented by *Cellebrite's Universal Forensics Extraction Device Physical Pro edition, version 1.1.3.8* (Grispos et al., 2011).

The same forensic suite was also used in order to complete the logical examination. Since data extracted from physical acquisition techniques were not in human readable format, tools such as forensic toolkits and file carvers were used to interpret different kinds of files from the binary image. Apart from commercial file carvers, the Klaver (2010) python script, *cedbexplorer.py* was used as a different approach to achieve data extraction from database files, such as cemail.vol. Moreover, string extractors were used for extracting the content of allocated files. On the other hand, logical and manual acquisition extractions needed no further editing. After completing every kind of acquisition, the authors implemented a technique where retrieved data and artifacts were tested through MD-5 hash functions and compared to the original ones. As a result, retrieved pieces of information were classified to four categories, three out of which concerned the relevance between the original and the extracted ones. The last one was used if the artifacts were neither detected nor supported. Data that had not been fully detected were additionally tested by a fuzzy hash, in order to measure its similarity to the original ones. As expected, results deriving from the two categories of acquisition types were different at a major scale. Unallocated files were only retrieved through physical acquisition and the use of WinHex, while it was unable to recover any data relevant to appointments, contacts and call logs, most of which were stored in the embedded databases. Logical acquisition was capable of recovering those data types. Various differences were presented even between retrieved files of the same kind. The researchers concluded that, in order to acquire a fulfilling data set concerning a forensic investigation, a set of acquisition methods and tools should be applied. Also, extra attention has to be paid on the integrity and validity of the tools used.

Satheesh Kumar et al. (2012) proposed an agent-based approach for forensic acquisition on Windows Mobile devices. A unique feature of the specific research was that, apart from a technical framework introduction, the writers dedicated a section describing the most significant parts of a real-time investigation procedure divided into seven phases: Identification, Seizure, Acquisition, Authentication, Analysis, Presentation and Preservation. They claimed that all phases are of equal importance to the investigation results, but two of them, acquisition and analysis, present major technical significance. The authors implemented the tool they proposed on these two aforementioned phases. Firstly, they made a small-scale evaluation of acquisition methods and recommended physical as the high quality and effective one. Due to limitations concerning the division of internal memory types to flash ROM and RAM, they concluded that physical acquisition couldn't be performed in a satisfying scale; logical acquisition on the other hand is the most frequent choice. Meanwhile, they calculated the connectivity requirements needed for a proper setup, such as USB/Bluetooth bridge from the mobile device to a computer and ActiveSync/Windows Mobile Device Center (WMDC) installation on the workstation. Then, they gathered the forensically important data sets, which resided in the file system, databases and registry. The software agent the researchers proposed was an all-in-one tool, combining logical acquisition attributes enabling it to interact with the most important databases (cemail.vol and pim.vol) and pseudo-physical acquisition traits, such as RAPI tools (Casey et al., 2010;

Klaver, 2010). In their research there were no innovative features, but a complete solution other than applying many different techniques at the same time. Finally, they presented the tool infrastructure, acquisition results on a test device and compared it to other commercial forensic tools. Such a comparison though couldn't be considered accurate, since some tools didn't provide simultaneous physical and logical acquisition support.

To the best of our knowledge, the research related to devices running the newer versions of the Windows Mobile OS, Windows Phone 7 and 8, is significantly weak. Kaart et al. (2013) mentioned that databases (EDB) previously present in older versions of the OS, such as *pim.vol* can also be found in Windows Phone devices and that its structure and contents have significant forensic value. Moreover, they pointed out that Klaver (2010) has conducted a high quality research in manipulation of EDB files for forensic purposes. Due to the fact of not being able to track down some reports or research papers describing the structure of the *pim.vol* file, they decided to use reverse-engineering, aiming to acquire as much information as possible. One of the first warnings they addressed to the community was the uncertainty of the accuracy concerning their assumption. They used hex-dumps and structure diagrams for representing the structural elements and other details of the file. The experiments took place in a device running version 6.1 of the OS. The writers proceeded in developing a custom tool that "extracted databases and records from the EDB volumes" (Kaart et al., 2013) into the *Traces* framework. Afterwards, they presented the structure of the EDB file, and analyzed the existing page types and connections between them. One interesting trait discovered, was the relation between unallocated entries and record slots. Specifically, when a change in an entry occurred, a new slot pointer was created. This implied that older record may exist until some other entry occupies their slot. The existence of irrelevant number or letter sequences between records implied the use of *XPRESS* compression. Lastly, the extraction and interpretation results were also compared to the results used in the research paper by Klaver (2010)concluding that the number of exported entries was equal in both cases. Moreover, some databases, that had not been successfully traced by the method in Klaver (2010), such as _ _*sysDeletedRows* and _ _ *sysRowTrack* were exported in the current case. Since the results had been satisfactory enough, the authors proposed the same method to be applied in Windows Phone 7 devices, so as to trigger further research to that direction.

JTAG is one widely used physical acquisition method. Even if the initial use of the JTAG port is to produce test vectors for board examination or debugging in embedded systems (Breeuwsma, 2006). The author took advantage of the formerly mentioned port functions so as to extract a bitwise memory image. The paper was not solely dedicated to the description of the JTAG method, but also to the possible impediments an investigator might have faced, such as the *watchdog timer*, which is responsible for resetting the processor when spotting the absence of normal functioning control signals, or other incidents related to reset. Hence, he proposed solutions to these issues, such as reset prevention by forfeiting the embedded system with a sample vector. Moreover, he provided the potential investigator with a walkthrough on finding and accessing the JTAG port. Finally, the author presented an example image of the JTAG anatomy from a target device used during the experimental phase. Note that this paper has been published in 2006. However, it is included in the current work due to the importance of the findings it presents.

Yates (2010) seems to hold the credits for primary forensic research, while the field of SSDF wasn't fully formed. His study consisted of the main characteristics of the most prevailing OS for mobile devices and how they can be utilized in potential MF investigations. Later on, he exposed the unique features of mobile devices that make the forensics discipline referring to them more complicated than that of desktop computers. Moreover, he presented some commonly used forensic suites and tools as a point of reference.

Mutawa et al. (2012) focused on performing a forensic examination of mobile devices equipped with the most popular OSs, Android (ver. 2.3.3), Blackberry (ver. 6.0) and iOS (ver. 4). More precisely, Facebook, Twitter and MySpace mobile versions of applications were installed, and used in order to provide a satisfying amount of information to be retrieved. After conducting a logical acquisition on the devices, they performed a manual analysis on each of the logical images acquired. The extraction and analysis procedure was fully certified, since they consulted the guidelines provided by NIST (Jansen and Ayers, 2007). They argued that the investigation procedure would be more accurate if browser data were taken into consideration as well, since many users prefer logging in to social networks from a plain browser tab than the application itself. Therefore, there was no data loss, because the investigation was originally restricted to the use of applications only.

Blackberry Desktop Software (BDS) was utilized to perform logical acquisition in the test-subject Blackberry phones, with the sync option disabled. Official backup software prevents from changes taking place to crucial elements of the logical image. Thus, this method is acceptable on court (ACPO, 2007; Jansen and Ayers, 2007). A logical copy was acquired, but no traces of data deriving from social networking applications were found. Since this operation failed, further research should be conducted on the subject. Although the authors claim Zdziarski's method (Zdziarski, 2008; Hoog and Gaffaney, 2009) to be superior and the most accurate iPhone forensics technique, their investigation is limited to the use of iTunes backup utility, with the automatic sync feature powered off. This decision poses the dilemma of: *to jailbreak or not to jailbreak*. Nevertheless, data acquisition was successful, since a great amount of social networking interaction data could be retrieved from all the applications tested. This kind of information includes: Facebook user and friend data, friends with active chat sessions, timestamps, comments posted, all previously logged-in users; Twitter usernames, profile pictures and tweets; Myspace credentials, posts and timestamps. Android examination seems to be the most disputable category, since acquiring data from a non-rooted device eliminates the quantity and quality of useful information to be gathered. On the other hand, rooting a device makes significant changes to the potential admissible evidence upon court. The option of rooting has been chosen, aiming to effectiveness. A standalone application, namely MyBackup was used to extract the logical backup to an external SD card. Analysis of the logical image returned usernames, pictures uploaded and viewed, chat messages and created albums from Facebook; usernames, tweets and device info for Twitter and usernames, passwords, cached files and cookies for MySpace. Although the results from the examination procedure were satisfying enough for two out of the three OS platforms tested, it is hard to reach certain conclusions. There is a need for use of different techniques and approaches, enhanced by the use of different forensics tools, since they all have altering specifications in various data types. Further research should be conducted in order to produce statistics that will lead to strengthening any weak points showing up.

Every research method can contribute some interesting assets to the field, such as knowledge, application of it in practice and experimental results. Research and experiments conducted by (Chun and Park, 2012) belong to that specific category. They studied logical acquisition techniques from two target devices, running the Android OS ver. 2.1 and the Windows Mobile ver. 6.1.

Research was strictly limited to certain types of acquisition and data types. The authors proceeded to logical acquisition with the use of official backup and sync suites for each devicee. ActiveSync was used for the Windows Mobile one, while Kies for Android. ActiveSync is considered a global backing up suite for Windows Mobile handsets; Android devices on the other hand, apart from few exceptions such as the Samsung mobile phones don't have a specific backing up suite baseline (Grover, 2013). This last fact imposes many questions concerning the different interaction of Android devices to that type of data extraction and creates more experimental needs. Since the retrieval procedure only consisted of interaction with official product software, there was no need for interfering with security locks or other trespassing methods, such as rooting. This approach though is not applicable to a real scenario, since the amount of data to be acquired will be relatively smaller compared to the one demanded for fulfilling the needs of a typical investigation.

The data types the authors aspired to extract were also limited to user-edited ones, such as images, SMS and contacts. One severe factor that was not taken into consideration or left intentionally undocumented was the isolation of the devices from any kind of networking source. The only potential precaution for both devices was that they had to be switched on during the acquisition procedure. Official backup suites had the ability to restore deleted files, but there was no further information provided about the quantity or quality of them. Lack of metrics concerning data integrity and availability was also spotted in the results concerning the allocated data acquisition, since the authors had just claimed that some files were impossible to be read. A limited scale research can provide useful information. However, since the research field is small, such kind of studies have to be supported by a critical mass of experimental data for creating detailed metrics and statistics.

The great majority of field studies step on existing methods in order to evolve or even disapprove them. Fewer, on the other hand, develop innovative research elements of mobile devices and how they can become useful to development of new theories. Park et al. (2012) discussed a method of forensic analysis from fragmented memory pages, when the reconstruction of the file system is impossible. Their first goal was to explain the basic characteristics of memory types of smartphones and the mechanisms they use for balancing deletion actions alongside the memory in order to augment the estimated life span of the device. Next, they clarified that data acquisition can be partially easy, since they are only deleted on block level. They then proposed a process model concerning forensic analysis, in which they introduced the flash memory page analysis that can occur whenever the reconstruction of the file system is impossible or for not allocated areas in the file system.

The flowchart proposed in this work can be considered an extended version of the one demonstrated by Simao et al. (2011). By the time the research was undertaken, the process model was applicable only to Android and iOS devices. One of the most interesting features of the process model they proposed was that it concerned a real-time investigation scenario and not separate acquisition techniques. The memory analysis method consisted of four scalable steps, which serve in reducing the estimated size of the memory to be examined. Duplicate entries, metadata and forms of certain data outlined by their size were excluded from the examination. The last step concerns the remaining data classification, according to their format. Data that have been through compression, such as audio, video and documents were named after *random* and their acquisition is considered a difficult task, due to their size and consequently their fragmentation level. This can be easily explained, since the bigger a file is, the more scattered it can be through memory blocks. On the other hand, non-random data are smaller in size, such as Web pages, SQLite databases and text data. Both categories need specialized tools in order to be retrieved and completely restored. For proper documentation, the authors also conducted two case studies in two mobile devices, supporting the YAFFS and the EXT4 file system. The method proposed is still in a primitive phase of development and many features need to be added for providing full functionality, such as simultaneous support of two file types on one memory page.

4. Discussion

As already mentioned, MF is a relatively new discipline. The first guidelines and studies - published around 2007 - were major contributions to the field, since many future research initiatives were influenced by them. A relatively big growth in the number of publications was noticed in 2011 and 2012, when smartphones had become increasingly widespread and the same had occurred to their involvement into crimes.

4.1. Representation of the major contributions in chronological order

Figure 1 represents a timeline of literature concerning the field of MF. As already mentioned, works prior to 2007 are not included, since technology is considered outdated and thus beyond the scope of this survey. The horizontal axis in the bottom of the figure represents the presence, absence or coexistence of low-level modification mechanisms. Studies may concern modified devices, non-modified ones, while in some others, both types can coexist. Different kinds of geometrical shapes refer to OS types (e.g., Circle to Android, diamond to Blackberry, rounded rectangle to theoretical background and standardization and so forth). Shapes are placed within the diagram according to their chronological order. Numbers inside them correspond to the entry number in the References, while letters to the acquisition type presented. Choice of letters is as close to the first letter of each acquisition type as possible. *P* stands for Physical Acquisition, *L* for Logical Acquisition, *A* for all acquisition types, while *N* shows that no techniques were mentioned in the context of the corresponding work. Texture fill is related to the data types each research work examined. Large grid implies all data types, dark vertical (vertical lines) refer to the user data category, dark horizontal (horizontal lines) to user and application data, dark downward diagonal to OS and application data, and a lack of texture indicates absence of data types. Solid lines between two shapes imply influence, while dashed ones imply compliance or reference to theory and regulations. On the other hand, dash dot dot texture represents an implicit reference, i.e. similar points of view that don't refer directly to each other.

As we observe from Figure 1, older contributions affected (and continue to affect) many others in the following years. Therefore, for obtaining a holistic view of the MF field, one has to go back a considerable number of years. This is actually the reason why we decided to take into account 7 years of MF developments, and comes despite the fact that technological advances on smartphone hardware and associated platforms are moving ahead in a very rapid pace. Also, one can argue that since several research works as depicted in Figure 1 are influenced and based on past ones, there is a tendency of researchers improving previous studies than deciding to occupy themselves in the exploration of new research spaces. From a quantitative point of view, almost half of the research works examined (25 out of 53) concerned devices without any low-level modification (root, jailbreak or capability hack). This occurred due to the implicit need for the provision of an original image of the state of the device according to the proposed guidelines (ACPO, 2007; Jansen and Ayers, 2007). The majority of research papers shown in Figure 1 that belong to the non-modified category have been influenced by the guidelines. Notably, the appearance frequency of studies concerning low-level modified devices is augmenting during the last two years. This fact implies the necessity for re-examination of the theoretical background concerning the austerity of admissibility methods, since investigations upon non-modified devices is resulting to a smaller quantity of acquired evidence. Claims mentioned above can also be verified from the fact that recent studies are concerned but not restricted to existing guidelines/specifications. This incident shows an escape tendency. Apart from the theoretical background, assumptions extracted may trigger new research interests towards design and implementation of different hardware architecture for mobile devices, so they can become more forensic-tolerant.

Bear in mind that the more a mobile OS gets popular worldwide and taking into consideration the exceptions we have mentioned previously, the more research focus on it becomes. For instance, from 2009 to 2011, the number of works concerning Android forensics was multiplied by three. Nevertheless, research should not halt for OS types that have been outdated, since they can still contribute a lot to investigations and newer techniques can be effective on older devices and file systems as well.

Thirty nine (29) out of fifty three (53) works included research for all the data types mentioned in Section 2.3. This result signifies that everything is important in an investigation and can serve as possible evidence. There is no disregard to any data type. Therefore, user data are the most highly appreciated, since they are the most important piece of evidence. As a result, almost every research is taking them into consideration. Moreover, calls and SMS data were the ones with the greatest evidential importance in the survey of McMillan et al. (2013). Another data combination appearing frequently is user and application data. It is mostly spotted in studies concerning social networking or Instant Messaging (IM) applications, when a combination of both data types was inevitable. User data are highly appreciated, but, depending on the underlying case, location (GPS) data may also be of high importance (Lessard and Kessler, 2009; Bader and Baggili, 2010; Hoog, 2011; Maus et al., 2011; Simao et al., 2011; Mylonas et al., 2013). As it is known, location (GPS) data are not only limited to a pair of co-ordinates, but also can be related to routes the suspect or the victim has roamed or even specific Points of Interest(POIs) that can shed light to each case. (Berla Corporation, 2010).

4.2. Towards a common framework

Lack of standardization and frequent adaptation is a major issue in the field of MF (Jansen and Delaitre, 2009; Lessard and Kessler, 2009). Rapid changes in technology, variations and gaps among different kinds of mobile devices and OSs are making the procedure of creating a common framework or standardization model a hard but challenging task for organizations and researchers. Nevertheless, a great number of similar attributes are pointing to that direction. The argument can be validated through the admission of the fact that there are many common patterns among acquisition methods and tools designed for each case. Client/server based models in forensic tools implementation is one of the attributes, both present in Blackberry and Windows Mobile devices (Sasidharan and Thomas, 2011). Similar adaptation schemes are present in completely different research work. One of the most vivid examples was the use of approximately the same flowchart concerning acquisition techniques on Android and iOS devices, while there was no direct influence between the two works (Park et al., 2012; Simao et al., 2011). Such contributions can set the base of a generally acceptable model. Moreover, many logical acquisition-oriented research use backup and synchronization suites. Information Systems Analysis Methodology can be applied so as scientists to be able to develop a multi-leveled MF Framework, where the lower (zero) level would consist of common, general conceptions, and the higher level ones would concern the different attributes and actions performed. The *Forensic Spiral* (Jansen and Delaitre, 2009) was an early official attempt for general standardization, concerning forensic tools deployment in the mobile computing ecosystem. The backbone of the model was based on the fact that technology is shape-shifting quickly; so a potential forensic tool should be able to comply with changes, but not be distributed until being officially validated by the corresponding organization. Modification of the device state at the moment of seizing is another issue to be resolved and has a common extent to every kind of device.

There also seems to be some lack of standardization of evidence description, up to a semantic/ontological level. This is already done for other computer forensics (Brinson et al., 2006; Gayed et al., 2012).

Last but not least, we could conclude that we can only consider the *right* and maybe *wrong* MF tool within context, i.e. the first responder needs to take on board the suspect's capabilities (offender profiling) and the particular offense of course, for being able to decide which data are of relevance to the case and which could be discarded. So in order to have a useful and applicable standardization exercise, these aspects need also to be included. In other words, we need to accept that there are no universally accepted *best practices*, but only context dependent ones. The former assumption is strengthened after the very recent NIST draft on MF guidelines by Ayers et al. (2013). The document updates the state-of-the-art and focuses on the variety of forensic tools and methods existing that lead to the conclusion that the appropriate method to be used is relative and based on the incident. As mentioned in (Chabrow, 2013), one of the main characteristics of the new draft guideline, is the absence of a strict acquisition rule; "the guidance is not all-inclusive and does not prescribe how law enforcement and incident response communities should handle mobile devices during their investigations or incidents".

4.3. Future Challenges

Almost every research considered in the context of this work concludes or implies that a forensic investigation is complete when every possible acquisition method is applied. Even though at least three different physical acquisition methods have been spotted, the most widespread is the use of (adapted) bootloaders, no matter the OS of the target device. This happens not only because it is considerably the safest method amongst all but because it is simultaneously cost-effective and providing satisfying results. Although pseudo-physical acquisition is documented enough, dissemination of information and documentation are still poor concerning the other kinds of physical acquisition, such as chip-off and JTAG. When a real-time incident takes place, forensic analysts will need more time and effort for acquiring the amount of information needed in order to perform the other tasks of physical acquisition. Existing tools for most common makes and brands can facilitate the investigation procedure, but a documentation concerning experimental results could make the real difference. Even if experiments in a big scale concerning JTAG and chip-off techniques may be less affordable, they still have to be conducted.

Accessibility of features and parts of the mobile devices that are crucial for forensic investigations, such as bootloaders and the RAM heap should be ensured. A boot loader might lack the functionality to copy memory while the RAM heap might be practically inaccessible.

The great majority of experiments takes place on specific brands of mobile devices and versions of operating systems. It is generally accepted though, that even devices that run the same OS present different behavior. This means that while a MF method may be operational and useful for a certain version of a given mobile platform, it can become obsolete very quickly due, say, to the installation of an OS patch. As a result, brand and model diversity is another factor that needs to be taken into serious consideration.

Many researchers rely solely on commercial forensic tools, taking advantage of their ease-of-use compared to raw acquisition techniques. Even though this approach can be less time-consuming, there is always a possibility that results may not be satisfactory enough. Optimal retrieval quality level can be achieved by the combination of commercial suites and also innovative research methods.

Newer versions of mobile OSs are equipped either with security mechanisms that are hard to bypass, such as Symbian versions after 9.0, or preserve data integrity with encryption methods to avoid possible compromising situations, such as iOS 4.x. These facts impose the challenge of continuously adapting the existing methods to the new standards and experiments, for the researchers to have a complete view of the field. Additionally, effective acquisition methods applied in the past have to be revised, evaluated from the onset and modified (if possible), so as to comply with the new technology attributes, that is, newer OS versions and the embedment of advanced and sophisticated hardware. Hash functions are used in order to preserve the integrity of the acquired data sets. If the fingerprint of the retrieved file or instance is the same as the original, then acquisition procedure has been completed successfully. On the other hand, hash functions may allow collisions, which, at least theoretically, can be used as exploits. An intentionally modified acquired file or instance can bypass hashing control and then be able to obfuscate an investigation procedure, by altering, deleting or modifying existing data on the device.

Since newer research shows escape tendencies from the theoretical background (that is demanding a total absence of altering the devices state after seizure), it is debatable whether and to which degree OS architecture or low-level modification culture should be revisited. A total recall concerning hardware is a difficult and high-cost procedure; it also will not provide any kind of solution for older devices. On the contrary, enabling rooting privileges ever since a device is released and implementing a software-based solution in order to preserve security and prevent from misuse is probably a more decent solution, which can also be applied to older devices by, say, a firmware upgrade.

Also, it appears that due to the nature of smartphones, many MF procedures must inevitably involve live forensics as the device needs to be powered on (traditional dead forensics are almost useless). There is an upcoming trend of comparison between live and dead forensic techniques (Pooters, 2010; Garfinkel, 2010; AlZarouni, 2011; Lai et al., 2011; Sylve et al., 2012).

As such, MF needs to consider approaches followed by triage tools (incident response tools). This is also a very hot and topical research domain. In relation to the above point, network forensics seem also to be closely coupled to MF as smartphone devices are in (virtually) constant and pervasive connectivity with other devices. According to the work proposed by Hoog (2011), modifications taking place on a device, intentional or not, can compromise further acceptance as evidence. One of the ways that prevent this kind of changes is isolation from any network source, which, by default presents a plethora of advantages and disadvantages. There are applications where loss of connectivity triggers destruction of data (i.e., the opposite with the *remote wipe* Blackberry and iOS functionality (Research In Motion, 2011; Apple Inc., 2013). This is for sure to challenge the first responder's judgement and generate the dilemma "*to place the seized device in a faraday bag, or not to..?*". This also shows the importance of Principle 1 and 2 of ACPO (ACPO, 2007).

As already mentioned in section 1, malware especially designed for modern mobile platforms are rapidly becoming a serious threat. In most cases, such malicious software is acounted for compromising the normal behaviour of the device in order to bypass the security controls and execute malicious commands. Mobile devices infected by malware may be also part of a botnet anticipating remote commands to unleash synchronized attacks (Damopoulos et al., 2011).

Without doubt, mobile malware growth is also a great concern in the context of MF. To determine whether a mobile device has been compromised by malware it is necessary to perform either an offline analysis, where a duplicate of the flash memory is examined, or a live examination directly on the device (Casey, 2013). For instance, a customized bootloader that bypasses the OS of the device may be used to obtain the flash memory. After that, one is able to mount the forensic image on a forensic examination system and have it scanned using a proper version of antivirus software. Generally, however, there exist two main types of software analysis methods, namely static and dynamic. According to the first one, the software of interest is directly analyzed using the source code, or if not available, the corresponding binary file by means of reverse-engineering techniques (Schmidt et al., 2009; Egele et al., 2011). In dynamic analysis, on the other hand, the behavior of the software is examined while it is executed by the OS on the host device (Blasing et al., 2010; Damopoulos et al., 2012b).

A recently discovered vulnerability (Forristal, 2013) in every Android version implemented until now is expected not only to provoke certain kind of turmoil to the ecosystem, but also serve as an information extraction means for the MF discipline. It is related to Android application package file (APK) code modification without interfering to any cryptographic mechanisms. Putting it another way, this weakness allows changing an application's code leaving at the same time the cryptographic signature of the application untouched. This way, Android can be fooled into believing the application is original. Thus, any Trojan can penetrate in the system. This kind of attack is not only implemented for the Android OS. Malware injection is rather a growing tendency, especially for jailbreaking kind of attacks in other OSs as well. A recent example that doesn't even impose rooting (at least in a direct way) refers to this attack type with the use of a malicious (disguised) usb charger, developed specifically for the iOS platform (Arnott, 2013). MF scientists can take advantage of this attribute and establish a mutual communication bus between the target device and a forensic workstation.

Very recently, a new category of home entertainment devices, namely SmartTV, have penetrated the market. SmartTV systems have also high forensic value as it can reveal important information about their user. These devices have very similar architecture with that of a mobile device as they afford CPU, memory, network interfaces and custom or legacy OS, including Android and iOS. Thus, MF methods and tools may be well-fitted to those systems as well. However, so far, we are not aware of any MF method being applied in the context of these systems. This is due to the limited number of running services, the lack of rich accessibility interfaces (for low-level interaction with the device), and the restricted OS access permissions. Moreover, until now, no rooting (jailbreaking) methods have arisen to be applicable in SmartTV (Kuipers et al., 2013). Thus, a unified framework needs to consider how the technology will evolve in the years to come.

The emergence of innovative acquisition methods like the FROST one (Muller and Spreitzenbarth, 2013) encourages the further implementation of similar techniques, that seem to be a promising future for the discipline. Nonetheless, the data alteration issue still has to be resolved.

Last but not least, a research timeline has to be updated over the years, in order to preserve the ability of observing the trends within the field. This will provide researchers with quicker and more effective decision making processes.

5. Conclusions

MF is a discipline which presents a steady growth. The research conducted and undergoing standardization attempts indicate that the area is under continuous development. After identifying the challenges, this work provides a comprehensive review and classification of the state-of-the-art research in the field of MF. It therefore contributes in presenting a holistic approach of how MF evolved along the years. Many OSs, acquisition and data types cases were examined and trends deriving from them were observed. As far as we are aware of, this is the first time an exhaustive and detailed survey of this kind is attempted. The current work can be used as a reference to anyone interested in better understanding the facets of this fast evolving area. It is also expected to foster research efforts to the development of fully-fledged solutions that put emphasis mostly to the technological, but also to the standardization aspect.

Acknowledgements

We would like to express our appreciation to the reviewers and especially to Editor in Chief Prof. Eoghan Casey for their valuable and constructive comments so as to ameliorate the quality of the present work.

References

504ensics Labs, 2013. Lime linux memory extractor. http://www.504ENSICS.com.

ACPO, 2007. Good Practice Guide for Computer-Based Electronic Evidence Ver. 4. http://7safe.com/electronic_evidence/ACPO_guidelines_computer_evidence_v4_web.pdf.

AlZarouni, M., 2011. Mobile handset forensic evidence: a challenge for law enforcement. http://ro.ecu.edu.au/adf/24/.

Andriotis, P., Oikonomou, G., Tryfonas, T., 2012. Forensic analysis of wireless networking evidence of android smartphones. In: Information Forensics and Security (WIFS), 2012 IEEE International Workshop on. pp. 109–114.

Apple Inc., 2013. icloud: Erase your device remotely. http://support.apple.com/kb/ph2701.

Arnott, N., 2013. ios malware injecting charger to be presented at black hat. http://http://www.imore.com/malware-injecting-ios-charger-be-presented-black-hat.

Arrifin, A., D'Orazio, C., Choo, K.-K. R., Slay, J., 2013. ios forensics: How can we recover deleted image files with timestamp in a forensically sound manner? In: 2013 International Conference on Availability, Reliability and Security. pp. 375–382.

ASTM International, 2009. Forensic science standards. http://www.astm.org/Standards/forensic-science-standards.html.

Ayers, R., Brothers, S., Jansen, W., 2013. Guidelines on mobile device forensics. http://csrc.nist.gov/publications/drafts/800-101-rev1/draft_sp800_101_r1.pdf.

Bader, M., Baggili, I., 2010. iphone 3gs forensics: Logical analysis using apple itunes backup utility. Small Scale Digital Device Forensics Journal 4 (1), 1–15.

Becker, A., Mladenow, A., Kryvinska, N., Strauss, C., 2012. Aggregated survey of sustainable business models for agile mobile service delivery platforms. Journal of Service Science Research 4, 97–121.

Belenko, A., 2011. Eppb: Now recovering blackberry device passwords. http://blog.crackpassword.com/2011/09/recovering-blackberry-device-passwords/.

Belenko, A., 2012. ios and blackberry forensics. http://www.slideshare.net/andrey.belenko/ios-and-blackberry-forensics.

Berla Corporation, 2010. Gps forensics. http://www.gpsforensics.org/.

Blasing, T., Batyuk, L., Schmidt, A.-D., Camtepe, S., Albayrak, S., 2010. An android application sandbox system for suspicious software detection. In: Malicious and Unwanted Software (MALWARE), 2010 5th International Conference on. pp. 55–62.

Breeuwsma, I. M., 2006. Forensic imaging of embedded systems using {JTAG} (boundary-scan). Digital Investigation 3 (1), 32 – 42.
URL http://www.sciencedirect.com/science/article/pii/S174228760600003X

Breeuwsma, M., Jongh, M. D., Klaver, C., Knijff, R. V. D., Roeloffs, M., 2007. Forensic data recovery from flash memory.

Brinson, A., Robinson, A., Rogers, M., 2006. A cyber forensics ontology: Creating a new approach to studying cyber forensics. Digital Investigation 3, Supplement, The Proceedings of the 6th Annual Digital Forensic Research Workshop (0), 37–43.

Casey, E., 2009. Common pitfalls of forensic processing of blackberry mobile devices. http://computer-forensics.sans.org/blog/2009/06/15/common-pitfalls-of-forensic-processing-of-blackberry-mobile-devices#.

Casey, E., 2011a. Digital Evidence and Computer Crime, 3rd Edition. Academic Press.

Casey, E., 2011b. Digital Evidence and Computer Crime: Forensic Science, Computers, and the Internet. Academic Press.

Casey, E., 2013. Smartphone forensics and mobile malware analysis. http://www.caseite.com/content/smartphone-forensics-and-mobile-malware-analysis.

Casey, E., Bann, M., Doyle, J., 2010. Introduction to windows mobile forensics. Digital Investigation 6, Embedded Systems Forensics: Smart Phones, GPS Devices, and Gaming Consoles, 136–146.

Casey, E., Cheval, A., Lee, J. Y., Oxley, D., Song, Y. J., 2011. Forensic acquisition and analysis of palm webos on mobile devices. Digital Investigation 8 (1), 37–47.

Chabrow, E., 2013. Nist revising mobile forensics guide. http://www.bankinfosecurity.com/nist-drafting-mobile-forensics-guide-a-6048.

Chavez, A., 2008. A jailbroken iphone can be a very powerfull weapon in the hands of an attacker. Tech. rep., Purdue University, Calumets CIT Department.

Chun, W.-S., Park, D.-W., 2012. A study on the forensic data extraction method for sms, photo and mobile image of google android and windows mobile smart phone. In: Convergence and Hybrid Information Technology. Vol. 310 of CCIS. Springer Berlin Heidelberg, pp. 654–663.

Damopoulos, D., Kambourakis, G., Anagnostopoulos, M., Gritzalis, S., Park, J., 2012a. User privacy and modern mobile services: are they on the same path? Personal and Ubiquitous Computing (Online First), 1–12.

Damopoulos, D., Kambourakis, G., Gritzalis, S., 2011. iSAM: An iphone stealth airborne malware. In: Proceedings of the 26th IFIP TC-11 International Information Security Conference, IFIP Advances in Information and Communication Technology - IFIP AICT. Springer, pp. 17–28.

Damopoulos, D., Kambourakis, G., Gritzalis, S., 2013. From keyloggers to touchloggers: Take the rough with the smooth. Computers & Security 32, 102–114.

Damopoulos, D., Kambourakis, G., Gritzalis, S., Park, S., 2012b. Exposing mobile malware from the inside (or what is your mobile app really doing?). Peer-to-Peer Networking and Applications, 1–11.

Distefano, A., Me, G., 2008. An overall assessment of mobile internal acquisition tool. Digital Investigation 5, Supplement, S121–S127.

EDRM LLC, 2013. Electronic Discovery Reference Model Stages. http://www.edrm.net/resources/edrm-stages-explained.

Egele, M., Kruegel, C., Kirda, E., Vigna, G., 2011. Pios: Detecting privacy leaks in ios applications. In: Proceedings of the 18th Annual Network & Distributed System Security Symposium (NDSS).

Fairbanks, K., Atreya, K., Owen, H., 2009. Blackberry ipd parsing for open source forensics. In: Southeastcon, 2009. SOUTHEASTCON '09. IEEE. pp. 195–199.

Fang, J., Jiang, Z., Chow, K.-P., Yiu, S.-M., Hui, L., Zhou, G., 2012a. Mtk-based chinese shanzhai mobile phone forensics. In: Eighth Annual IFIP WG 11.9 International Conference on Digital Forensic. pp. 1–9.

Fang, J., Jiang, Z., Chow, K.-P., Yiu, S.-M., Hui, L., Zhou, G., He, M., Tang, Y., 2012b. Forensic analysis of pirated chinese shanzhai mobile phones. In: Peterson, G., Shenoi, S. (Eds.), Advances in Digital Forensics VIII. Vol. 383 of IFIP Advances in Information and Communication Technology. Springer Berlin Heidelberg, pp. 129–142.

Forensics Wiki, 2012. Blackberry forensics. http://www.forensicswiki.org/wiki/Blackberry_Forensics#Blackberry_BBB_File_Format_.28Mac_OS_X.29_.28.bbb.29.

Forristal, J., 2013. Uncovering android master key that makes 99% of the devices vulnerable. http://bluebox.com/corporate-blog/bluebox-uncovers-android-master-key/.

Garfinkel, S. L., 2010. Digital forensics research: The next 10 years. Digital Investigation 7, Supplement, The Proceedings of the Tenth Annual DFRWS Conference (0), S64–S73.

Gayed, T. F., Lounis, H., Bari, M., 2012. Cyber forensics: Representing and (im)proving the chain of custody using the semantic web. In: Proceedings of the Fourth International Conference on Advanced Cognitive Technologies and Applications. COGNITIVE '12. IARIA, Nice, France.

Grispos, G., Storer, T., Glisson, W. B., 2011. A comparison of forensic evidence recovery techniques for a windows mobile smart phone. Digital Investigation 8 (1), 23–36.

Grover, J., 2013. Android forensics: Automated data collection and reporting from a mobile device. Digital Investigation 10, Supplement (0), S12 – S20, the Proceedings of the Thirteenth Annual {DFRWS} Conference 13th Annual Digital Forensics Research Conference.
URL http://www.sciencedirect.com/science/article/pii/S1742287613000480

Guido, M., Ondricek, J., Grover, J., Wilburn, D., Nguyen, T., Hunt, A., 2013. Automated identification of installed malicious android applications. Digital Investigation 10, Supplement (0), S96 – S104, the Proceedings of the Thirteenth Annual {DFRWS} Conference 13th Annual Digital Forensics Research Conference.

URL http://www.sciencedirect.com/science/article/pii/S1742287613000571

Harrill, Mislan, 2007. A small scale digital device forensics ontology. http://www.ssddfj.org/papers/ssddfj_v1_1_harrill_mislan.pdf.

He, M., Fang, J., Jiang, Z. L., Yiu, S. M., Chow, K. P., Niu, X., 2012. Digital forensic on mtk-based shanzhai mobile phone with nand flash. In: The First International Conference on Digital Forensics and Investigation. pp. 1–10.

Hewlett-Packard Development Company, L.P., 2011. Overview of hp webos. https://developer.palm.com/content/resources/develop/overview_of_webos/overview_of_webos.html.

Hoog, A., 2011. Chapter 6 - android forensic techniques. In: Android Forensics. Syngress, Boston, pp. 195–284.

Hoog, A., Gaffaney, K., 2009. iphone forensics. http://www.mandarino70.it/Documents/iPhone-Forensics-2009.pdf.

Husain, M. I., Baggili, I., Sridhar, R., 2011. A simple cost-effective framework for iphone forensic analysis. In: Digital Forensics and Cyber Crime. Vol. 53 of LNICST. Springer Berlin Heidelberg, pp. 27–37.

ipddump, 2011. Extract your information from a blackberry backup ipd. http://code.google.com/p/ipddump/.

ISO/IEC, 2012. Guidelines for identification, collection, acquisition, and preservation of digital evidence, ISO/IEC 27037:2012, First Edition. http://www.iso27001security.com/html/27037.html.

Jansen, W., Ayers, R., November 2004a. Guidelines on pda forensics. http://csrc.nist.gov/publications/nistpubs/800-72/sp800-72.pdf.

Jansen, W., Ayers, R., August 2004b. Pda forensic tools: An overview and analysis. http://csrc.nist.gov/publications/nistir/nistir-7100-PDAForensics.pdf.

Jansen, W., Ayers, R., 2007. Guidelines on cell phone forensics. http://csrc.nist.gov/publications/nistpubs/800-101/SP800-101.pdf.

Jansen, W., Delaitre, A., October 2009. Mobile forensic reference materials: A methodology and reification. http://csrc.nist.gov/publications/nistir/ir7617/nistir-7617.pdf.

Jung, J., Jeong, C., Byun, K., Lee, S., 2011. Sensitive privacy data acquisition in the iphone for digital forensic analysis. In: Secure and Trust Computing, Data Management and Applications. Vol. 186 of CCIS. Springer Berlin Heidelberg, pp. 172–186.

Kaart, M., Klaver, C., van Baar, R., 2013. Forensic access to windows mobile pim.vol and other embedded database (edb) volumes. Digital Investigation 9 (34), 170 – 192.
URL http://www.sciencedirect.com/science/article/pii/S1742287612000874

Kim, D., Park, J., Lee, K.-g., Lee, S., 2012. Forensic analysis of android phone using ext4 file system journal log. In: J. (Jong Hyuk) Park, J., Leung, V. C., Wang, C.-L., Shon, T. (Eds.), Future Information Technology, Application, and Service. Vol. 164 of LNEE. Springer Netherlands, pp. 435–446.

Klaver, C., 2010. Windows mobile advanced forensics. Digital Investigation 6, Embedded Systems Forensics: Smart Phones, GPS Devices, and Gaming Consoles (3-4), 147–167.

Kuipers, R., Starck, E., Heikkinen, H., 2013. Jailbreak/apple tv. http://theiphonewiki.com/wiki/Jailbreak/Apple_TV.

Lai, Y., Yang, C., Lin, C., Ahn, T., 2011. Design and implementation of mobile forensic tool for android smart phone through cloud computing. In: Lee, G., Howard, D., Slezak, D. (Eds.), Convergence and Hybrid Information Technology. Vol. 206 of CCIS. Springer Berlin Heidelberg, pp. 196–203.

Lessard, J., Kessler, G. C., 2009. Android forensics: Simplifying cell phone examinations. http://www.ssddfj.org/papers/SSDDFJ_V4_1_Lessard_Kessler.pdf.

Lohrum, M., 2012. Forensic extractions of data from the nokia n900. In: Gladyshev, P., Rogers, M. (Eds.), Digital Forensics and Cyber Crime. Vol. 88 of LNICST. Springer Berlin Heidelberg, pp. 89–103.

Marshall, A. M., 2011. Standards, regulation and quality in digital investigations: The state we are in. Digital Investigation 8, Standards, professionalization and quality in digital forensics (2), 141–144.

Marzougy, M., Baggili, I., Marrington, A., 2013. Blackberry playbook backup forensic analysis. In: Rogers, M., Seigfried-Spellar, K. (Eds.), Digital Forensics and Cyber Crime. Vol. 114 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Berlin Heidelberg, pp. 239–252.
URL http://dx.doi.org/10.1007/978-3-642-39891-9_15

Maus, S., Hofken, H., Schuba, M., 2011. Forensic analysis of geodata in android smartphones.
URL http://www.schuba.fh-aachen.de/papers/11-cyberforensics.pdf

McKemmish, R., 1999. What is Forensic Computing? Trends & issues in crime and criminal justice. Australian Institute of Criminology.
URL http://books.google.pt/books?id=NoqGmgEACAAJ

McMillan, J., Glisson, W., Bromby, M., January 2013. Investigating the increase in mobile phone evidence in criminal activities. In: HICSS-46.
URL http://eprints.gla.ac.uk/69395/

Mokhonoana, P. M., Olivier, M. S., 2007. Acquisition of a symbian smart phone's content with an on-phone forensic tool. In: Southern African Telecommunication Networks and Applications Conference 2007 (SATNAC 2007) Proceedings.

Morris, B., 2006. Symbian OS Architecture Sourcebook. John Wiley & Sons.

Morrissey, S., 2010. iOS Forensic Analysis: for iPhone, iPad, and iPod touch, 1st Edition. Apress, Berkely, CA, USA.

Muller, T., Spreitzenbarth, M., 2013. Frost. In: Jacobson, M., Locasto, M., Mohassel, P., Safavi-Naini, R. (Eds.), Applied Cryptography and Network Security. Vol. 7954 of Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 373–388.
URL http://dx.doi.org/10.1007/978-3-642-38980-1_23

Mutawa, N. A., Baggili, I., Marrington, A., 2012. Forensic analysis of social networking applications on mobile devices. Digital Investigation 9, Supplement (0), S24–S33.

Mylonas, A., Meletiadis, V., Mitrou, L., Gritzalis, D., 2013. Smartphone sensor data as digital evidence. Computers and Security [in press Elsevier]. URL http://www.sciencedirect.com/science/article/pii/S0167404813000527

Park, J., Chung, H., Lee, S., 2012. Forensic analysis techniques for fragmented flash memory pages in smartphones. Digital Investigation 9 (2), 109–118.

Pooters, I., 2010. Full user data acquisition from symbian smart phones. Digital Investigation 6, Embedded Systems Forensics: Smart Phones, GPS Devices, and Gaming Consoles (3-4), 125–135.

Quick, D., Alzaabi, M., 2011. Forensic analysis of the android file system yaffs2. In: Proceedings of the 9th Australian Digital Forensics Conference, Edith Cowan University, Perth Western Australia.

Racioppo, C., Murthy, N., 2012. Android forensics: A case study of the htc incredible phone. http://csis.pace.edu/~ctappert/srd2012/b6.pdf.

Rehault, F., 2010. Windows mobile advanced forensics: An alternative to existing tools. Digital Investigation 7, 38–47.

Research In Motion, 2011. Remote wipe reset to factory defaults it policy rule. http://docs.blackberry.com/en/admin/deliverables/4222/Remote_Wipe_Reset_to_Factory_Defaults_250402_11.jsp.

Sasidharan, S., Thomas, K., 2011. Blackberry forensics: An agent based approach for database acquisition. In: Abraham, A., Lloret Mauri, J., Buford, J., Suzuki, J., Thampi, S. (Eds.), Advances in Computing and Communications. Vol. 190 of CCIS. Springer Berlin Heidelberg, pp. 552–561.

Satheesh Kumar, S., Thomas, B., Thomas, K., 2012. An agent based tool for windows mobile forensics. In: Gladyshev, P., Rogers, M. (Eds.), Digital Forensics and Cyber Crime. Vol. 88 of LNICST. Springer Berlin Heidelberg, pp. 77–88.

Savoldi, A., Gubian, P., 2009. Issues in symbian s60 platform forensics. Journal of Communication and Computer 6.

Schmidt, A.-D., Bye, R., Schmidt, H.-G., Clausen, J., Kiraz, O., Yüksel, K. A., Camtepe, S. A., Albayrak, S., 2009. Static analysis of executables for collaborative malware detection on android. In: Proceedings of the 2009 IEEE international conference on Communications. ICC'09. IEEE Press, Piscataway, NJ, USA, pp. 631–635.

Simao, A., Sicoli, F., Melo, L., Deus, F., Sousa, J. R., 2011. Acquisition and analysis of digital evidence in android smartphones. The International Journal of Forensic Computer Science 6 (1), 28–43.

Son, N., Lee, Y., Kim, D., James, J. I., Lee, S., Lee, K., 2013. A study of user data integrity during acquisition of android devices. Digital Investigation 10, Supplement (0), S3 – S11, the Proceedings of the Thirteenth Annual {DFRWS} Conference 13th Annual Digital Forensics Research Conference.
URL http://www.sciencedirect.com/science/article/pii/S1742287613000479

SWGDE, 2009. Swgde best practices for mobile phone examinations v1.0. https://www.swgde.org/documents/Current%20Documents/2009-05-21%20SWGDE%20Best%20Practices%20for%20Mobile%20Phone%20Examinations%20v1.0.

Sylve, J., Case, A., Marziale, L., Richard, G. G., 2012. Acquisition and analysis of volatile memory from android devices. Digital Investigation 8 (3-4), 175–184.

Team HelloOx, 2013. HelloOx. http://www.helloox2.com.

Thing, V., Chua, T.-W., 2012. Symbian smartphone forensics: Linear bitwise data acquisition and fragmentation analysis. In: Computer Applications for Security, Control and System Engineering. CCIS. Springer Berlin Heidelberg, pp. 62–69.

Thing, V., Tan, D., 2012. Symbian smartphone forensics and security: Recovery of privacy-protected deleted data. In: Information and Communications Security. Vol. 7618 of LNCS. Springer Berlin Heidelberg, pp. 240–251.

Tso, Y.-C., Wang, S.-J., Huang, C.-T., Wang, W.-J., 2012. iphone social networking for evidence investigations using itunes forensics. In: Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication. ICUIMC '12. ACM. Article 62, pp. 1–7.

Vidas, T., Zhang, C., Christin, N., 2011. Toward a general collection methodology for android devices. Digital Investigation 8, S14–S24.

Volatile Systems, 2011. The volatility framework: Volatile memory artifact extraction utility framework. https://www.volatilesystems.com/default/volatility.

Yates, M., 2010. Practical investigations of digital forensics tools for mobile devices. In: 2010 Information Security Curriculum Development Conference. InfoSecCD '10. ACM, New York, NY, USA, pp. 156–162.

Yu, X., Jiang, L.-H., Shu, H., Yin, Q., Liu, T.-M., 2009. A process model for forensic analysis of symbian smart phones. In: Advances in Software Engineering. Vol. 59 of CCIS. Springer Berlin Heidelberg, pp. 86–93.

Zdziarski, J., 2008. iPhone Forensics. Recovering Evidence, Personal Data, and Corporate Assets. O'Reilly Media.

Zimmermann, C., Spreitzenbarth, M., Schmitt, S., Freiling, F. C., 2012. Forensic analysis of yaffs2. In: Suri, N., Waidner, M. (Eds.), Sicherheit 2012: Sicherheit, Schutz und Zuverlassigkeit. Vol. 195 of LNI. GI, pp. 59–69.

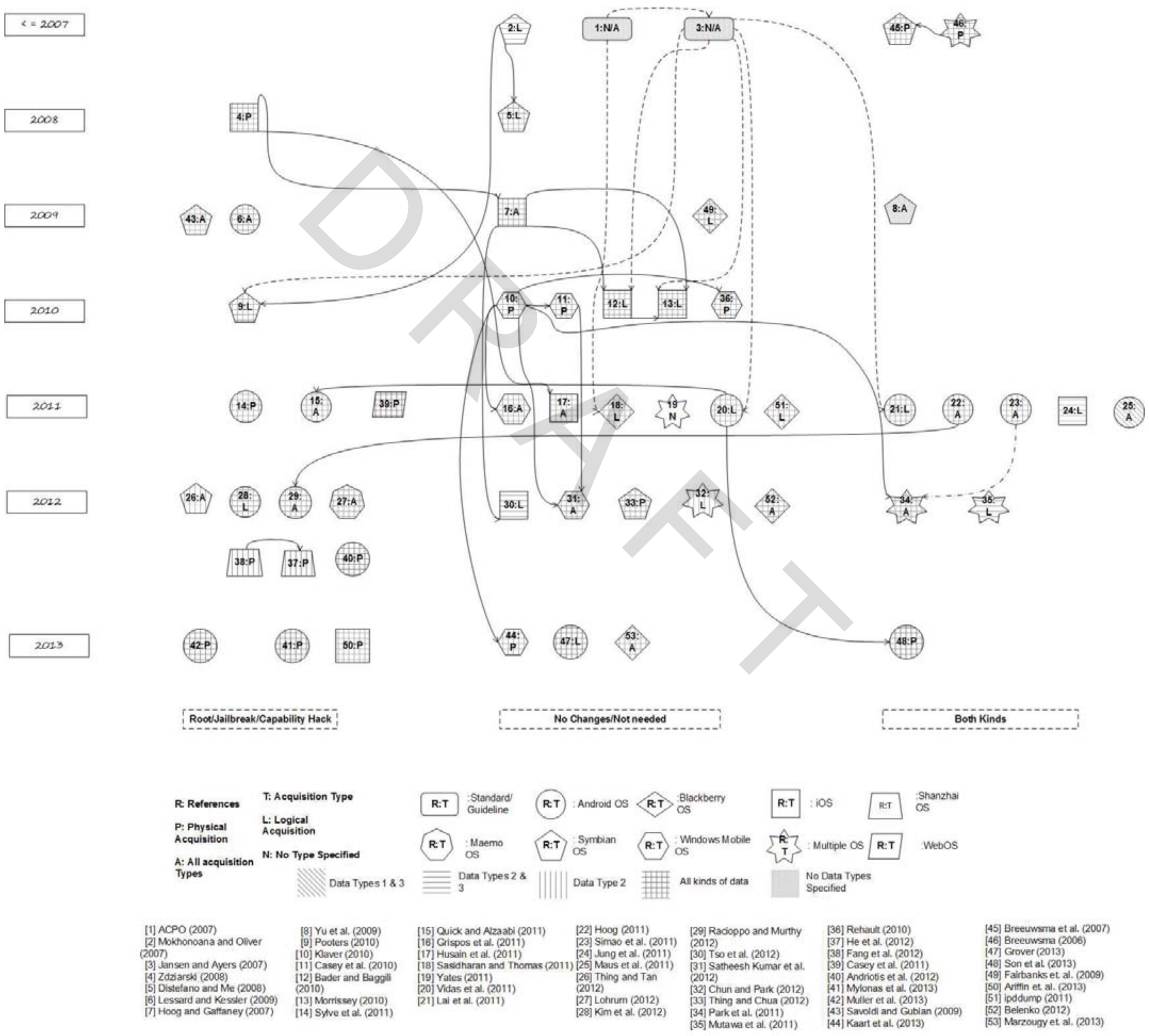

Figure 1: Major smartphone forensics approaches in chronological order. The arrows indicate interrelation between proposals ((i.e., [b] has been influenced by [a])