UNIVERSITY OF PIRAEUS
DEPARTMENT OF DIGITAL SYSTEMS

MASTER PROGRAM IN
DIGITAL SYSTEMS SECURITY

Mobile Device Forensics: Guidelines, Analysis and Tools

By

Konstantinos Georgokitsos

Supervisor: Dr. Christoforos Dadoyan

Master Thesis submitted to the Department of Digital Systems of the University of Piraeus in partial fulfillment of the requirements for the degree of Master of Science in Digital Systems Security

Piraeus, Greece, March 2018

Disclaimer

This document reflects the results of a study that has been prepared on behalf of the Postgraduate Programme «Digital Systems Security» at University of Piraeus. The information and conclusions contained in this thesis express the author’s personal opinion and arguments, and therefore should not be interpreted that they represent the official concepts of University of Piraeus.

Abstract

Mobile device forensics is the science of recovering digital evidence from mobile device under forensically sound conditions using accepted methods. Mobile device forensics is an evolving specialty in the field of digital forensics and there is an increase in the number of mobile device forensics (MoDeFo) tools for proper recovery and speedy analysis of data present on mobile devices. Scope of this thesis is to provide an in-depth look into the technologies involved and their relationship to mobile device forensic procedures, the challenges associated while carrying forensic analysis and to elaborate various forensic analysis techniques and tools. This document also discusses procedures for the preservation, acquisition, examination, analysis, and reporting of digital information on mobile devices as part of forensic analysis procedures.

Keywords: Analysis, Digital Evidence, Tools, Forensic, Preservation, Data Acquisition, Instant Messaging, Android Forensics, Instant Messenger Forensics.

Contents

List of Figures

List of Tables

1 Introduction

Mobile devices, such as smart phones and tablets, have become an integral part of people’s daily lives. The global adoption of such devices has been growing faster than any other consumer technology in history. These small factor devices introduce a new processing and communication paradigm, enabling end-users to access and manage a broad set of data and services, while on the move. They combine advanced computing capability, such as internet communication, information retrieval, video, e-commerce and other features, defining a personal accessory that allows constant social connection. This is the fact which makes the device one of the necessities for many people.

The success of the mobile phone can be attributed to the fact that it helps satisfy the human need for instant gratification. Waiting is inconvenient, which is why the majority of technology in the consumer market is marketed around providing more, better, and faster functionality. To materialize this, a wide range of mobile applications have been developed, which are extending from entertainment and gaming to critical mobile banking and proprietary enterprise applications for accessing corporate resources. Although compact, these handheld devices can contain personal information including call history, text messages, digital photographs, videos, passwords, credit card numbers, as well as corporate information contained in e-mails, calendar items, memos and address books.

The contemporary mobile software that provides sophisticated communication services over the internet, allowing users to exchange textual messages, as well as audio, video, and image files, has changed the way people interact among them. The usage of these services, broadly named *instant messaging* (IM), has undoubtedly exploded in the past few years, thanks to the extended use of smartphones that provide quite sophisticated IM applications. Instant messaging facilitates a more streamlined flow of communication. While phone calls or emails can be time consuming, instant messaging allows people to address issues in a live and prompt manner with minimum disruptions.

In addition to legitimate uses, however, IM applications are increasingly being used to carry out illicit activities. Similar to other popular consumer technologies, IM services have been exploited by criminals to commit frauds, dissemination of malware, personal data theft or even more extreme criminal activities such as grooming children online with the purpose of sexual exploitation or establishing communication channels between terrorists. Therefore, it was imperative to develop technics and mechanisms to explore in depth the Instant Messaging Applications and be able to extract information by analyzing data contained in devices' file system. This requirement led to the birth of mobile digital forensics, a branch of digital forensics that deals with the recovery of digital evidence or data from mobile devices, under forensically sound conditions. Digital evidence, which is defined as *"Digital evidence or electronic evidence is any probative information stored or transmitted digitally and a party to a judicial dispute in court can use the same during the trial"*, can be found in memory modules and data storage areas of mobile telephones. This evidence can prove an important part of a criminal or civil prosecution. Deleted text messages can be recovered, which can reveal not only purposes and objectives but also suspect's plan of action. As different mobile devices are built differently, specialized forensic techniques are required to ensure that mobile telephone forensics assessments conducted are done so in a forensically sound mode and that the information extracted will endure the inquiry of a court of law[1].

[1] Archit Goel, Anurag Tyagi, Ankit Agarwal, *Smartphone Forensic Investigation Process Model*, International Journal of Computer Science & Security (IJCSS), Volume (6) : Issue (5) : 2012.

2 Background

2.1 Android Operation System & Security

In this thesis we will focus on IM Applications installed on devices running Android operation system (Android OS) despite the fact that those apps are also available for iOS operation system as well. Android is a Linux-based operating system designed, primarily, for touch screen mobile devices such as smart phones and tablet computers. Since its appearance, Android followed an upward trajectory and wide acceptance. The main reason is that it is an open-source software which can be customized, configured or rebuilt and still operates functionally on the same devices. The applications can be developed by companies or individuals and be compatible with all different android devices that run a specific android version or more. Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007. Google actively develops the Android platform but gives a portion of it for free to hardware manufacturers and phone carriers who want to use Android on their devices.

The most important thing about a device that fulfills multiple purposes and therefore keeps personal data of the owner, is security. Over the years, mobile devices adopt the principal of Security by Design and implements protection mechanisms that involve security hardware, like biometric sensors and Trusted Platform Modules (TPMs). Although, an extra layer of security is implemented by the operating system itself. Android uses a permission-based system to further enhance security on the device. When an Android application is to be installed, it first requests a list of permissions that it requires which the user must then accept to install the app. These permissions include access to device resources such as Internet, Bluetooth and External Storage. If an application requires access to these resources, the developer must state so in the application's "AndroidManifest.xml" file located within the Application Package (APK) for the application. Android applications are generally written in the Java programming language and converted into a format (DEX) compatible with the virtual machine used in the Android OS, known as the Dalvik Virtual Machine (Dalvik). In addition to permissions that normal applications (applications that are installed into the "/data/app" directory) may use, there are system level permissions that may only be used by applications which are installed in the "/system/app" directory and signed with the developer key that was used to sign the device's operating system. These applications are known as system applications. The "data" and "system" directories, along with several others, are mount points for partitions that Android uses in order to further compartmentalize the system.[2]

2.2 Mobile Device Forensics History and Application Fields

Digital forensics is a branch of forensic science encompassing the recovery and investigation of material found in digital devices, often in relation to computer crime. Digital forensics investigations have a variety of applications. The most common is to support or refute a hypothesis before criminal or civil courts. Forensics may also feature in the private sector, such as during internal corporate investigations or intrusion investigation.[3] Mobile device forensics is a branch of digital forensics relating to recovery of digital evidence or data from a mobile device under forensically sound conditions. Although the phrase mobile device usually refers to mobile phones, however, it can also relate to any digital device that has both internal memory and communication ability, including PDA devices, GPS devices and tablet computers. The use of mobile devices in crime was widely recognized for some years, but the forensic study of mobile devices is a relatively new field, dating from the early 2000s and late 1990s. A proliferation of smartphones and other digital devices on the consumer market caused a demand for forensic examination of the devices, which could not be met by existing computer forensics techniques.[4]

As the world of mobile devices continues to evolve, so has their ability to create, store, and transmit electronic communications and other forms of electronically stored information. It is becoming increasingly common for corporate employees at all levels to rely on a tablet, smartphone, or other mobile device as their primary means of corporate communications. As a result, the field of digital forensics has needed to keep pace with this rapid evolution of mobile devices, cloud services, and mobile applications. Mobile device security, preserving and collecting evidence from devices, and the basics of electronically stored information analysis are critical components in any mobile device forensic investigation.[5]

[2] Ben Martini, Quang Do, Kim-Kwang Raymond Choo, *Conceptual Evidence Collection and Analysis Methodology for Android Devices*, Information Assurance Research Group, University of South Australia, 2015

[3] Wikipedia, The free encyclopedia, *Digital forensics*, https://en.wikipedia.org/wiki/Digital_forensics

[4] Wikipedia, The free encyclopedia, *Mobile device forensics*, https://en.wikipedia.org/wiki/Mobile_device_forensics

[5] Brandon Letha, Arnold Garcia, *Mobile Device Forensics: The New Frontier*, iDiscovery Solutions, The Metropolitan Corporate Counsel, February 2014, Page 20.

Mobile devices do not just operate as stand-alone data sources. They can constantly synchronize with other devices and applications, either directly or via the cloud. This means data may exist in places where investigators might not think to look. Not only does this constant synchronization affect collection, but it also impacts preservation. When dealing with mobile devices, forensic teams need to consider the requirements of the matter at hand. This includes the specific devices and potential security obstacles, along with other software and applications that may be part of the synchronization process, separate memory sources and volatile data.

2.3 Mobile Device Forensics Policies and Set of Controls

Digital Evidence is "*information of probative value that is stored or transmitted in binary form*". Thus any useful information stored or transferred in digital mode is evidence regardless of the devices or interfaces used to store or transfer it. Therefore, smart phones are a promising site for collecting such evidence[6]. The fundamental principles of mobile device forensics can be thought of as rules governing the way in which digital evidence is handled which allow such evidence to be admissible in court. Immediately we can see that any attempt to define these principles is made difficult by the fact that legislation concerning digital evidence differs from country to country. Mathew Braid defined five rules for evidence in order to be considered useful: it must be admissible, authentic, complete, reliable and believable[7]. Courts demand stringent requirements on the admissibility of digital evidence at trial. Admissibility is a legal concept that prescribes requirements to judge the acceptance of any kind of evidence – be it digital or not[8]. As a result, digital evidence creates a necessity for appropriate methods in order to prove and preserve its significance and authenticity[9].

The requirement of authentication or identification as a condition precedent to admissibility is satisfied by evidence sufficient to support a finding that the matter in question is what its proponent claims. However, digital evidence in particular must be regarded skeptically since it

[6] Archit Goel, Anurag Tyagi, Ankit Agarwal, *Smartphone Forensic Investigation Process Model*, International Journal of Computer Science & Security (IJCSS), Volume (6) : Issue (5) : 2012.

[7] Matthew Braid, *Collecting electronic evidence after a system compromise*, 2001.

[8] Rafael Accorsi, *Safekeeping Digital Evidence with Secure Logging Protocols: State of the Art and Challenges*, Department of Telematics, University of Freiburg, Germany.

[9] Jennifer Richter, Nicolai Kuntze, Carsten Rudolph, *Securing Digital Evidence*.

only exists in a binary representation. Because digital information can be created easily and without any verifiable record of who did so, and it can be changed, often without detection, courts are grappling with ways to authenticate digital evidence under these circumstances[10]. Without adequate protection mechanisms such data can be easily manipulated without leaving any traces[11]. In order to demonstrate that digital evidence is what the proponent claims it to be, the foundation must take into account not only the legal requirements of procedure and evidence (addressing relevance, reliability, accuracy, authentication and related issues), but must also include an evaluation of each of the components of the information system from which the evidence was generated.

Therefore, attempts have been made to standardize principles on an international basis and the following are commonly agreed upon:

- The act of collecting digital evidence should not result in any alteration of the data in question, wherever this is possible.
- All handling of digital evidence (from collection through to preservation and analysis) must be fully documented.
- Access to original digital evidence should be restricted to those deemed "forensically competent".

Each of the above principles requires more detailed explanation to be properly appreciated and understood.[12]

As it is described in the “Overview of Digital Forensics” which was published by ISACA in 2015, the enterprise cybersecurity program should have policies that address all forensics considerations, such as contacting law enforcement, monitoring, and conducting regular reviews of forensics policies, guidelines and procedures. Good practice requires that policies are part of an overall governance and management framework, such as COBIT 5, from ISACA, which provides a hierarchical structure into which all policies should fit and link clearly to the

[10] Lucy L. Thomson, *Human Rights Electronic Evidence Study: Admissibility of electronic documentation as evidence in U.S. Courts*, December 1, 2011.

[11] Jennifer Richter, Nicolai Kuntze, Carsten Rudolph, *Securing Digital Evidence*.

[12] Forensic Focus, *Principles of computer forensics*, http://www.forensicfocus.com/principles-of-computer-forensics.

underlying principles. Mobile devices present greater challenges in handling due to memory volatility, so proper handling procedures must be followed to protect digital data.

Most mobile devices have a basic set of comparable features and capabilities. They house a microprocessor, read-only memory (ROM), random access memory (RAM), a radio module, a digital signal processor, a microphone and speaker, a variety of hardware keys and interfaces, and a liquid crystal display (LCD). The operating system of a mobile device may be stored in either NAND or NOR memory, while code execution typically occurs in RAM. Generally, the information collected comes from internal memory (flash memory) or external memory (subscriber identity module [SIM], Secure Digital [SD], Multi Media Card [MMC], Compact Flash [CF] cards or memory sticks). Call records and mobile backups can also be obtained through carriers, which provide other information that is useful in developing evidence, especially in cases of encryption.

2.4 Mobile Device Forensics Scientific Process

Ken Zatyko, the former director of the Defense Computer Forensics Laboratory, defined the following eight-step digital forensics scientific process:[13]

1. **Obtain search authority:** In a legal investigation, legal authority is required to conduct a search or seizure of data.
2. **Document chain of custody:** In legal contexts, chronological documentation of evidence handling is required to avoid allegations of evidence tampering or misconduct.
3. **Image and hash:** When digital evidence is found, it should be carefully duplicated and then hashed to validate the integrity of the copy.
4. **Validate tools:** When possible, tools that are used for forensics should be validated to ensure reliability and correctness.
5. **Analyze:** Forensic analysis is the execution of investigative and analytical techniques to examine the evidence.
6. **Repeat and reproduce (quality assurance):** The procedures and conclusions of forensic analysis should be repeatable and reproducible by the same or other forensic analysts.

[13] Zatyko Ken, *Commentary: Defining Digital Forensics*, Forensic Magazine, January 2007.

7. **Report:** The forensic analyst must document his/her analytical procedure and conclusions for use by others.
8. **Possibly present expert testimony:** In some cases, the forensic analyst will present his/her findings and conclusions to a court or another audience.

The process involves more than intrusion-related security incidents. Zatyko defines scientific digital forensics as:

"The application of computer science and investigative procedures for a legal purpose involving the analysis of digital evidence after proper search authority, chain of custody, validation with mathematics, use of validated tools, repeatability, reporting, and possible expert presentation."

Digital forensics follows a rigorous scientific process to present findings of fact to prove or disprove a hypothesis in a court of law, civil proceeding or another action. Zatyko's eight-step process can be broadly divided into four stages[14]: *Preservation, Acquisition, Examination, and Analysis and Reporting,* which are analyzed in the following sections, where the connection between the procedure and the investigation stages is presented.

2.5 Preservation

The stages of forensic analysis are the most important part of the investigation, to recover data and acquire information, in order to be able to use it in the court of law. But before all that, it is necessary for the investigation to gather all the evidence in a secure manner and ensure its integrity and good condition. This is where preservation takes place, providing a strict framework to avoid any alteration of the digital evidence.

Evidence preservation is the process of securely maintaining custody of property without altering or changing the contents of data that reside on devices and removable media. It is the very first step in digital evidence recovery. Preservation involves the search, recognition, documentation, and collection of electronic based evidence. Failure to preserve evidence in its original state could jeopardize an entire investigation, potentially losing valuable case-related information.

[14] Rick Ayers, Sam Brothers, Wayne Jansen, *Guidelines on Mobile Device Forensics*, NIST Special Publication 800-101 Rev. 1, May 2014.

2.5.1 Securing and Evaluating the Scene

Incorrect procedures or improper handling of a mobile device during seizure may cause loss of digital data. Moreover, traditional forensic measures, such as fingerprints or DNA testing, may need to be applied to establish a link between a mobile device and its owner or user. If the device is not handled properly, physical evidence may be contaminated and rendered useless. A Mobile Station and is partitioned into two distinct components: the Universal Integrated Circuit Card (UICC) and the Mobile Equipment (ME). Alertness to mobile device characteristics and issues (e.g., memory volatility) and familiarity with tangential equipment (e.g., media, cables, and power adapters) are essential. For mobile devices, sources of evidence include the device, UICC and associated media. Associated peripherals, cables, power adapters, and other accessories are also of interest. All areas of the scene should be searched thoroughly ensuring related evidence is not overlooked.

Equipment associated with the mobile device, such as removable media, UICCs, or personal computers, may prove more valuable than the mobile device itself. Removable media varies in size and can be easily hidden and difficult to find. Most often, removable memory cards are identifiable by their distinctive shape and the presence of electrical contacts located on their bodies that are used to establish an interface with the device. Personal computers may be particularly useful in later accessing a locked mobile device, if the personal computer has established a trusted relationship with it. While securing a mobile device, caution should be taken when an individual is allowed to handle the mobile device. Many mobile devices have master reset codes that clear the contents of the device to original factory conditions. Master resets may be performed remotely requiring proper precautions such as network isolation to ensure that evidence is not modified or destroyed.

Mobile devices and associated media may be found in a damaged state, caused by accidental or deliberate action. Devices or media with visible external damage do not necessarily prevent the extraction of data. Damaged equipment should be taken back to the lab for closer inspection. Repairing damaged components on a mobile device and restoring the device to working order for examination and analysis may be possible. Undamaged memory components may also be removed from a damaged device and their contents recovered independently. This method should be used with caution, as it is not possible with all devices.

2.5.2 Documenting the Scene

Evidence must be accurately identified and accounted for. Non-electronic materials such as invoices, manuals, and packaging material may provide useful information about the capabilities of the device, the network used, account information, and unlocking codes for the PIN. Photographing the crime scene in conjunction with documenting a report on the state of each digital device and all computers encountered may be helpful in the investigation, if questions arise later about the environment. A record of all visible data should be created. All digital devices, including mobile devices, which may store data, should be photographed along with all peripherals cables, power connectors, removable media, and connections. The investigator should avoid touching or contaminating the mobile device when photographing it and the environment where found. If the device's display is in a viewable state, the screen's contents should be photographed and, if necessary, recorded manually, capturing the time, service status, battery level, and other displayed icons.

2.5.3 Isolation

Many mobile devices offer the user with the ability to perform either a remote lock or remote wipe by simply sending a command (e.g., text message) to the mobile device. Additional reasons for disabling network connectivity include incoming data (e.g., calls or text messages) that may modify the current state of the data stored on the mobile device. Outgoing data may also be undesirable as the current GPS location may be delivered to an advisory providing the geographic location of the forensic examiner. Therefore, forensic examiners need to be aware and take precautions when securing mobile devices mitigating the chance of data modification. The Scientific Working Group on Digital Evidence's (SWGDE) "Best Practices for Mobile Phone Forensics" document covers best practice for the proper isolation of mobile devices [SWG13]. Some key implications for proper collection are summarized below.

Isolating the mobile device from other devices used for data synchronization is important to keep new data from contaminating existing data. If the device is found in a cradle or connected with a personal computer, pulling the plug from the back of the personal computer eliminates data transfer or synchronization overwrites. It is recommended that a capture of the personal computer's memory be extracted before disconnecting the device, as memory acquired generally proves to be of significant forensic value. The mobile device should be seized along with associated hardware. Media cards, UICCs, and other hardware residing in the mobile device should not be removed. Also, seizing the computer that was connected to the mobile device provides the ability to acquire synchronized data from the hard disk that might not be obtained from the device. Any associated hardware such as media cards, UICCs, power adapters, device sleeves, or peripherals, should be seized along with related materials such as product manuals, packaging, and software.

Isolating a mobile device from all radio networks (e.g. WiFi, Cellular and Bluetooth) is important to keep new traffic, such as SMS messages, from overwriting existing data. Besides the risk of overwriting potential evidence, the question may arise whether data received on the mobile device after seizure is within the scope of the original authority granted. Vulnerabilities may exist that may exploit a weaknesses related to software vulnerabilities from the web browser and OS, SMS, MMS, third-party applications and WiFi networks. The possibility of such vulnerabilities being exploited may permit the argument that data may have been modified during the forensic examination.

Finally, in order to completely isolate the device from any external network or electromagnetic field that may affect the data, specialists use specific equipment to block any physical interference. A couple of examples of this kind of equipment, are Radio Isolation Containers, Cellular Network Isolation Cards, Jammers and Spoofing Devices.

2.5.4 Packaging, Transporting and Storing Evidence

Once the mobile device is ready to be seized, the forensic specialist should seal the device in an appropriate container and label it appropriately according to agency specifications. Due to the volatile nature of some mobile devices, they should immediately be checked into a forensic laboratory for processing and the power requirements should be discussed with the evidence custodian. Battery powered devices held in storage for more than a day risk power depletion and data loss, unless a process is in place to avoid this outcome. Storage facilities that hold evidence should provide a cool, dry environment appropriate for valuable electronic equipment. All evidence should be in sealed containers in a secure area with controlled access.

2.5.5 On-Site Triage Processing

Currently many organizations are challenged with large backlogs of digital forensics casework. An on-site triage solution is being employed more and more world-wide to accommodate for this exponential growth in digital forensic caseload. Triaging involves performing a data extraction (i.e., Manual or Logical) on-scene followed immediately by a preliminary analysis of the data extracted. Logical extraction tools are providing additional capabilities to use keywords and specific known hashes alerting the on-scene examiner immediately to potential issues that need to be addressed. Where possible, devices supporting encryption, such as Android and iOS devices, should be triage processed at the scene if they are found in an unlocked state, as the data may no longer be available to an investigator once the device's screen is locked, or if the battery exhausts. Deploying the use of field forensics tools to either acquire the device, or establish a trusted relationship with the device, will ensure that the data can be accessed at a later time, after the device has locked.

2.5.6 Generic On-Site Decision Tree

Generic on-site decision tree might be used as a general guideline for organizations and agencies to help with the prioritization of on-site triage examinations via the description of some of the actions and decision points. This provides a starting point intended for customization allowing alignment with existing policies and procedures.

The following list describes some of the actions and decision points contained within the tree.

- *Unlocked/Undamaged* – Is the device in an unlocked state and functional permitting a manual or logical data extraction?
- *Urgen*t – Do circumstances exist such that data extraction is required on site?
- *Lab less than 2 hours away* – Can the mobile device be transported to a forensics laboratory in less than 2 hours?
- *Tool/ Training* – Is the device supported by the tool and has the examiner received proper training?
- *Contact Expert* – The on-site examiner should contact an expert for additional assistance and guidance.
- *Battery More than 50%* – Does the device show that it has more than 50% remaining battery power?

- *Need More Data* - After the extraction is successful and the examiner has reviewed the results, is additional information or analysis required?[15]

The following diagram illustrates an example of an on-site decision tree:

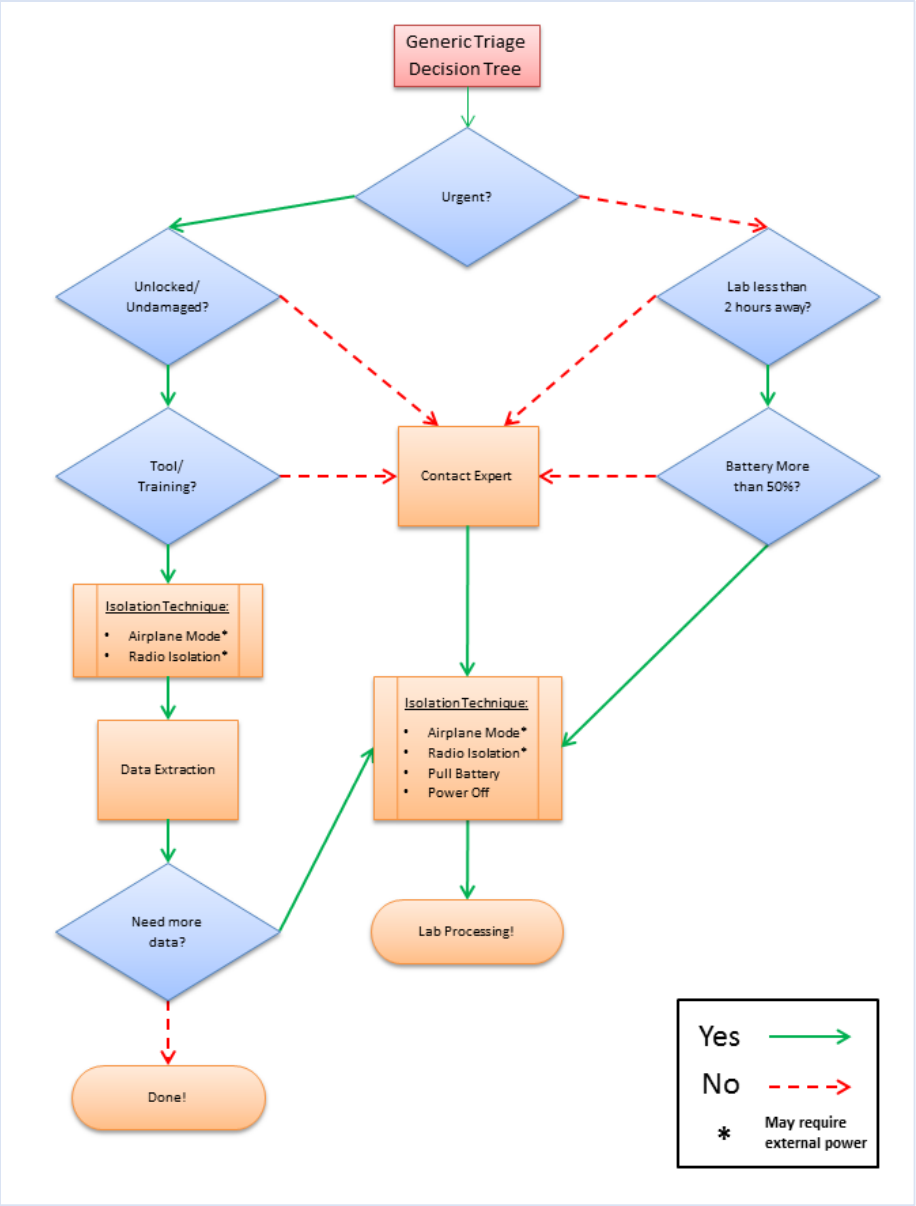


Figure 1: Generic Triage Decision Tree

[15] Rick Ayers, Sam Brothers, Wayne Jansen, *Guidelines on Mobile Device Forensics*, NIST Special Publication 800-101 Rev. 1, May 2014.

2.6 Acquisition

Acquisition is the process of imaging or otherwise obtaining information from a mobile device and its associated media. Performing an acquisition at the scene has the advantage that loss of information due to battery depletion, damage, etc. during transportation and storage is avoided. Off-site acquisitions unlike a laboratory setting may be challenging in finding a controlled setting in which to work with the appropriate equipment while satisfying additional prerequisites.

The forensic examination begins with the identification of the mobile device. The type of mobile device, its operating system, and other characteristics determine the route to take in creating a forensic copy of the contents of the device. The type of mobile device and data to be extracted generally dictates which tools and techniques should be used in an investigation.

2.6.1 Mobile Device Identification

To proceed effectively, mobile devices need to be identified by the make, model, and service provider. If the mobile device is not identifiable, photographing the front, back and sides of the device may be useful in identifying the make, model and current state (e.g., screen lock) at a later time. Individuals may attempt to thwart specialists by altering the mobile device to conceal its true identity. Device alteration may range from removing manufacturer labels to filing off logos. In addition, the operating system and applications may be modified or in rare situations completely replaced, and appear differently as well as behave differently than expected. These modifications should be taken into consideration on a case-by-case basis.

Other clues that allow identification of a mobile device include such things as: manufacturer logos, serial numbers, or design characteristics (e.g. candy bar, clam shell). Overall, knowing the make and model helps to limit the potential service providers, by differentiating the type of network the device operates over (i.e., GSM, non-GSM), and vice versa. Synchronization software discovered on an associated computer may also help to differentiate among operating system families. Further means of identification include the following:

- Device Characteristic
- Device Interface
- Device Label

- Carrier Identification
- Reverse Lookup.

2.6.2 Tool Selection and Expectations

Once the make and model of the mobile device are known, available manuals should be retrieved and studied. The manufacturer's web site is a good place to begin. Typing the model number into a search engine may also reveal a significant amount of information about the mobile device. As mentioned earlier, the device being acquired largely dictates the choice of forensic tools. The following criteria have been suggested as a fundamental set of requirements for forensic tools, and should be considered when a choice of tools is available:

Usability – the ability to present data in a form that is useful to an investigator

Comprehensive – the ability to present all data to an investigator so that both inculpatory and exculpatory evidence can be identified

Accuracy – the quality of the output of the tool has been verified

Deterministic – the ability for the tool to produce the same output when given the same set of instructions and input data

Verifiable – the ability to ensure accuracy of the output by having access to intermediate translation and presentation results

Tested – the ability to determine if known data present within the mobile device internal memory is not modified and reported accurately by the tool

Experimenting with various tools on test devices to determine which acquisition tools work efficiently with specific mobile device types is highly recommended. Besides gaining familiarity with the capabilities of the tool, experimentation allows special purpose search filters and custom configurations to be setup before use in an actual case.

2.6.3 Mobile Device Memory Acquisition

Mobile devices are often submitted for laboratory processing with only specific items requested for recovery, such as call logs or graphics. If any doubt or concerns exist about the requested data, contacting the submitter for clarification is recommended. Though it is not always necessary to recover all available data, a complete acquisition avoids having to redo the process later if additional data is requested. For examinations involving a limited scope search warrant (e.g., only text messages), a full memory data extraction may be completed but care should be taken to only report items covered by the warrant.

To acquire data from a mobile device, a connection must be established to the device from the forensic workstation. Before performing an acquisition, the version of the tool or device being used should be documented, along with any applicable patches or errata from the manufacturer applied to the tool. As mentioned earlier, caution should be taken to avoid altering the state of a mobile device when handling it, for example, by pressing keys that may corrupt or erase data. Once the connection has been established, the forensic software suite or device may proceed to acquire data from the mobile device.

The date and time maintained on the mobile device is an important piece of information. The date and time may have been obtained from the network or manually set by the user. Owners may manually set the day or time to different values from the actual ones yielding misleading values in the call and message records found on the mobile device. If the device was on when seized, the date and time maintained and differences from a reference clock should have already been recorded. Nevertheless, confirmation at the time acquisition may prove useful. If the mobile device was off when seized, the date and time maintained and differences from a reference clock should be recorded immediately when first powered on. Actions taken during acquisition, such as removal of the battery to view the device label, may affect the time and date values. Mobile devices may provide the user with an interface for a memory card.

Mobile device forensic tools that acquire the contents of a resident memory card normally perform a logical acquisition. If the device is found in an active state, the mobile device internal memory should be acquired before removing and performing a physical acquisition of the associated media (e.g., microSD Card). Otherwise, if the device is found in a power off state, a physical acquisition of the removable media should be performed before the internal handset memory of the mobile device is acquired. With either type of acquisition, the forensic tool may or may not have the capability to decode recovered data stored on the card (e.g., SMS text messages), requiring additional manual steps to be taken.

After an acquisition is finished, the forensic specialist should confirm that the contents of a device were captured correctly. On occasion, a tool may fail without any error notification and require the specialist to reattempt acquisition. It is advisable to have multiple tools available and be prepared to switch to another if difficulties occur with the initial tool.

Invariably, not all relevant data viewable on a mobile device using the available menus may be acquired and decoded through a logical acquisition. Manually scrutinizing the contents via the device interface menus while video recording the process not only allows such items to be captured and reported, but also confirms that the contents reported by the tool are consistent with observable data. Manual extraction must always be done with care, preserving the integrity of the device in case further, more elaborate acquisitions are necessary

The contents of a mobile device's memory often contain information, such as deleted data, that is not recoverable through either a logical or manual extractions. Lacking a software tool able to perform a physical acquisition, it may be necessary to turn to hardware-based techniques. Two techniques commonly used are acquisition through a standardized JTAG test interface, if supported on the device, and acquisition by directly reading memory that has been removed from the device.

2.6.4 GSM Mobile Device Considerations

Mobile devices that do not require a UICC are relatively straightforward as the acquisition entails a single device. Mobile devices requiring UICCs are more complex. There are two items that must be examined: the handset and the UICC. Depending on the state of the mobile device (i.e., active, inactive) the handset and UICC may be acquired jointly or separately. It is generally accepted to process the UICC first while the device is in an inactive state.

If the mobile device is active, a joint acquisition of the handset and UICC contents should be acquired first. A direct acquisition recovers deleted messages present on a UICC, while an indirect acquisition via the handset does not. The UICC must be removed from the mobile device and inserted into an appropriate reader for direct acquisition.

A well-known forensic issue that arises when performing a joint acquisition is that the status of unread text messages change between acquisitions. The first acquisition may alter the status flag of an unread message to read. Reading an unread text message from a UICC indirectly through the handset causes the operating system of the device to change the status flags. UICCs that are read directly by a tool do not make these modifications. One way to avoid this issue is to omit selecting the recovery of UICC memory when performing the joint acquisition (if the tool allows such an option).

If the mobile device is inactive, the contents of the UICC may be acquired independently before that of the handset. The UICC acquisition should be done directly through a PC/SC reader. The handset acquisition should be attempted without the UICC present. Many devices permit an acquisition under such conditions, allowing PIN entry for the UICC to be bypassed, if it were enabled. If the acquisition attempt is unsuccessful, the UICC may be reinserted and a second attempt made. Performing separate independent acquisitions (i.e., acquiring the UICC before acquiring the contents of the handset) avoids any operating system related forensic issues associated with an indirect read of UICC data. However, removing the SIM can reportedly cause data to be deleted on some mobile devices.

2.6.5 UICC Considerations

Similar to a mobile device, to acquire data from a UICC, a connection must be established from the forensic workstation to the UICC, using a PC/SC reader. As before, the version of the tool being used should be documented, along with any applicable patches or errata from the manufacturer applied to the tool. Once the connection has been established, the forensic software tool may proceed to acquire data from the UICC.

Capturing a direct image of the UICC data is not possible because of the protection mechanisms built into the module. Instead, forensic tools send command directives called Application Protocol Data Units (APDUs) to the UICC to extract data logically, without modification, from each elementary data file of the file system. The APDU protocol is a simple command-response exchange. Each element of the file system defined in the GSM standards has a unique numeric identifier assigned, which can be used to walk through the file system and recover data by referencing an element and performing some operation, such as reading its contents.

Because UICCs are highly standardized devices, few issues exist with regard to a logical acquisition. The main consideration is selecting a tool that reports the status of any PINs and recovers the data of interest. Vast differences exist in the data recovered by UICC tools, with some recovering only the data thought to have the highest relevance in a typical investigation, and others performing a complete recovery of all data, even though much of it is network related with little investigative value.

2.6.7 Tangential Equipment

Tangential equipment includes devices that contain memory and are associated with a mobile device. The three main categories are memory cards, host computers to which a mobile device has synchronized its contents and cloud-based storage.

Smartphones may provide an interface that supports removable media (e.g., microSD or MMC), which may contain significant amounts of data. Memory cards are typically flash memory, used as auxiliary user file storage, or as a means to convey files to and from the device. Data may be acquired with the use of a write-blocked media reader and a forensic application.

The data contained on a mobile device is often present on a personal computer, due to the capability of mobile devices to synchronize or otherwise share information among one or more host computers. Such personal computers or workstations are referred to as synched devices. Because of synchronization, a significant amount of data on a mobile device may be present on the owner's laptop or personal computer and recovered using a conventional computer forensic tool for hard drive acquisition and examination

2.6.8 Synched Devices

Synchronization refers to the process of resolving differences in certain classes of data, such as e-mail residing on two devices (i.e., a mobile phone and a personal computer), to obtain a version that reflects any actions taken by the user (e.g., deletions or additions) on one device or the other. Synchronization of information may occur at either the record level or the file level. When done at the file level, any discrepancies from the last synchronization date and time result in the latest version automatically replacing the older version. Occasionally manual intervention may be needed if both versions were modified independently since the last synchronization occurred. Record level synchronization is done similarly, but with more granularity, whereby only out-of-date parts of a file are resolved and replaced.

Mobile devices are typically populated with data from the personal computer during the synchronization process. A significant amount of informative data may reside locally on a personal computer. Data from the mobile device may also be synchronized to the computer, through user-defined preferences in the synchronization software. Because the synchronized contents of a mobile device and personal computer tend to diverge quickly over time, additional information may be found in one device or the other.

The synchronization software and the device type determine where mobile device files are stored on the PC. Each synchronization protocol has a default installation directory, but the location may be user specified.

2.6.9 Memory Cards

Memory card storage capacity ranges from 128MB and up. As technological advances are made, such media becomes physically smaller and offers larger storage densities. Removable media extends the storage capacity of mobile devices allowing individuals to store additional files beyond the device's built-in capacity and to share data between compatible devices.

Some forensics tools are able to acquire the contents of memory cards; many are not. If the acquisition is logical, deleted data present on the card is not recovered. Fortunately, such media can be treated similarly to a removable disk drive and imaged and analyzed using conventional forensic tools with the use of an external media reader.

A physical acquisition of data present on removable media provides the examiner the potential to search the contents of the media and potentially recover deleted files. One drawback is that mobile device data, such as SMS text messages may require manual decoding or a separate decoding tool to interpret. A more serious issue is that content protection features incorporated into the card may block the recovery of data. For instance, BlackBerry™ devices provide the user with the ability to encrypt data contained on the removable media associated with the mobile device.

2.7 Examination and Analysis

The examination process uncovers digital evidence, including that which may be hidden or obscured. The results are gained through applying established scientifically based methods and should describe the content and state of the data fully, including the source and the potential significance. Data reduction, separating relevant from irrelevant information, occurs once the data is exposed. The analysis process differs from examination in that it looks at the results of the examination for its direct significance and probative value to the case. Examination is a technical process that is the province of a forensic specialist. However, analysis may be done by roles other than a specialist, such as the investigator or the forensic examiner.

The examination process begins with a copy of the evidence acquired from the mobile device. Fortunately, compared with classical examination of personal computers or network servers, the amount of acquired data to examine is much smaller with mobile devices. Because of the prevalence of proprietary case file formats, the forensic toolkit used for acquisition will typically be the one used for examination and analysis. While interoperability among the acquisition and examination facilities of different tools is possible, only a few tools support this feature. Examination and analysis using 3rd party tools are generally accomplished by importing a generated mobile device memory dump into a mobile forensics tool that supports 3rd party mobile device images.

After the duplicate image of the evidence is created, analysis can begin on the image. The digital forensic analyst may use specialized tools to uncover deleted or hidden material. Depending on the forensic request, the analyst can report findings about numerous types of information, e.g., email, chat logs, images, hacking software, documents and Internet history. After evidence is collected and analyzed, it is assembled to reconstruct events or actions and provide facts to the requesting party. These facts may identify people, places, items and events and determine how they are related so that a conclusion can be reached. This effort can include correlating data among multiple sources. In some environments, early case assessment (ECA) provides immediate review for the requesting parties, at which time they can ask for more advanced analysis. ECA typically involves imaging, indexing, archiving and an internal reporting mechanism for the requesting party to quickly access needed reconnaissance. ECA typically saves time and is often preferred over analysis.

2.7.1 Potential Evidence

Mobile device manufacturers typically offer a similar set of information handling features and capabilities, including Personal Information Management (PIM) applications, messaging and e-mail, and web browsing. The set of features and capabilities vary based on the era in which the device was manufactured, the version of firmware running, modifications made for a particular service provider, and any modifications or applications installed by the user. The potential evidence on these devices may include the following items:

- Subscriber and equipment identifiers
- Date/time, language, and other settings
- Phonebook/Contact information
- Calendar information
- Text messages
- Outgoing, incoming, and missed call logs
- Electronic mail
- Photos
- Audio and video recordings
- Multi-media messages
- Instant messaging
- Web browsing activities
- Electronic documents
- Social media related data
- Application related data
- Location information
- Geolocation data.

The items present on a device are dependent not only on the features and capabilities of the mobile device, but also on the voice and data services subscribed to by the user. For example, prepaid phone service may rule out the possibility for multi-media messaging, electronic mail, and web browsing. Similarly, a contract subscription may selectively exclude certain types of service, though the phone itself may support them.

Two types of computer forensic investigations generally take place. The first type is where an incident has occurred but the identity of the offender is unknown (e.g., a hacking incident). The second is where the suspect and the incident are both known (e.g., a child-porn investigation). Prepared with the background of the incident, the forensic examiner and analyst may proceed toward accomplishing the following objectives:

- Gather information about the individual(s) involved {who}.
- Determine the exact nature of the events that occurred {what}.
- Construct a timeline of events {when}.
- Uncover information that explains the motivation for the offense {why}.
- Discover what tools or exploits were used {how}.

In many instances the data is peripheral to an investigation or useful in substantiating or refuting the claims of an individual about some incident. On occasion, direct knowledge, motivation, and intention may be established. Most of the evidence sources from mobile devices are: contact data, call data, messaging, pictures, video, social media, or Internet-related information. User applications potentially provide other evidence sources. User files placed on the device for rendering, viewing, or editing are other important evidence sources. Besides graphic files, other relevant file content includes audio and video recordings, spreadsheets, presentation slides, and other similar electronic documents.

2.7.2 Applying Mobile Device Forensic Tools

Once a copy of the acquisition results is available, the next steps involve searching the data, identifying evidence, creating bookmarks, and developing the contents of a final report. Knowledge and experience with the tools used for examination are extremely valuable, since proficient use of the available features and capabilities of a forensic tool can greatly speed the examination process.

It is important to note that forensic tools have the potential to contain some degree of error in their operation. For example, the implementation of the tool may have a programming error; the specification of a file structure used by the tool to translate bits into data comprehensible by the examiner may be inaccurate or out of date; or the file structure generated by another program as input may be incorrect, causing the tool to function improperly. Experiments conducted with mobile device forensic tools indicate a prevalence of errors in the formatting and display of data. Therefore, having a high degree of trust and understanding of the tool’s ability to perform its function properly is essential.

To uncover evidence, specialists should gain a background of the suspect, offense and determine a set of terms for the examination. Search expressions should be developed in a systematic fashion, such as using contact names that may be relevant. By proceeding systematically, the specialist creates a profile for potential leads that may unveil valuable findings. Suggestions for the analysis of extracted data, as NIST describes in its publication are the following ones:

- **Ownership and possession** – Identify the individuals who created, modified, or accessed a file, and the ownership and possession of questioned data by placing the subject with the device at a particular time and date, locating files of interest in nondefault locations, recovering passwords that indicate possession or ownership, and identifying contents of files that are specific to a user.
- **Application and file analysis** – Identify information relevant to the investigation by examining file content, correlating files to installed applications, identifying relationships between files (e.g., e-mail files to e-mail attachments), determining the significance of unknown file types, examining system configuration settings, and examining file metadata (e.g., documents containing authorship identification).
- **Timeframe analysis** – Determine when events occurred on the system to associate usage with an individual by reviewing any logs present and the date/time stamps in the file system, such as the last modified time. Besides call logs, the date/time and content of messages and e-mail can prove useful. Such data can also be corroborated with billing and subscriber records kept by the service provider.
- **Data hiding analysis** – Detect and recover hidden data that may indicate knowledge, ownership, or intent by correlating file headers to file extensions to show intentional obfuscation; gaining access to password-protected, encrypted, and compressed files; gaining access to steganographic information detected in images; and gaining access to reserved areas of data storage outside the normal file system.

Searching data for information on incriminating or exculpatory evidence takes patience and can be time consuming. Some tools have a simple search engine that matches an input text string exactly, allowing only for elementary searches to be performed. Other tools incorporate more intelligent and feature rich search engines, allowing for generalized regular expression patterns (grep) type searches, including wildcard matches, filtering of files by extension, directory and batch scripts that search for specific types of content (e.g., e-mail addresses, URLs). The greater the tool’s capabilities, the more the forensic examiner benefits from experience with and knowledge of the tool.

2.7.3 Call and Subscriber Records

Records maintained by the service provider capture information needed to accurately bill a subscriber or, in the case of a prepaid service plan, debit the balance. The records collected are referred to as call detail records (CDRs), which are generated by the switch handling an originating call or SMS message from a mobile device. For some service providers, the records may also include fixed line, international gateway, and voice over IP transaction information. While the content and format of these records differ widely from one service provider to another, the fundamental data needed to identify the subscriber/device initiating the call, the initial cell servicing the call, the number dialed, and the duration of the call is captured. Detailed information such as the identifier of the cell (i.e., the BTS) and the sector involved are often included. Besides call detail records, subscriber records maintained by a service provider can provide data useful in an investigation. For example, for GSM systems, the database usually contains the following information about each customer:

- Customer name and address
- Billing name and address (if other than customer)
- User name and address (if other than customer)
- Billing account details
- Telephone number (MSISDN)
- IMSI
- UICC serial number (ICCID)
- PIN/PUK for the UICC
- Services allowed.

Communication confidentiality is, however, a very sensitive topic in Greece, to a point that the Greek constitution makes reference to its safeguarding. Hellenic Authority for Communication Security and Privacy (ADAE) has taken up the responsibility of issuing regulation that electronic Communication Service Providers (CSPs) need to adopt in order to ensure the continuity, confidentiality and availability of their network and services to acceptable levels.

2.8 Reporting

After the analysis is complete, a report of the findings is developed, which outlines findings and methodologies. The provided exhibits may include attribution of file ownership, chat logs, images and emails, detailed login / logoff times, entry into facility logs and anything that places the suspect at the device at the same time and location of an event. The findings can be used to confirm or disprove alibis and provided statements. Digital evidence can also be used to prove intent. The completed report is given to the investigator, who is usually from law enforcement in a criminal matter or a designated senior manager in a civil action. Further actions are determined after the report is reviewed. Digital forensic analysts provide facts and impart knowledge to give expert opinion only when they are required to do so in court. They never seek to aid or blame. Instead, analysts provide a scientific basis so that the court, company or other requesting party may use the unbiased evidence and gain a better understanding of events.

Digital evidence, as well as the tools, techniques and methodologies used in an examination is subject to being challenged in a court of law or other formal proceedings. Therefore, proper documentation is essential in providing individuals the ability to re-create the process from beginning to end. As part of the reporting process, making a copy of the software used and including it with the output produced is advisable when custom tools are used for examination or analysis, should it become necessary to reproduce forensic processing results.

3 Mobile Device Forensic Tools and Equipment

3.1 Description of requirements

Computer forensic specialists either deal with the private or the public sector. With the public sector, their work is usually to support or refute a hypothesis before criminal or civil courts. The art of Cyber forensic investigation is quite complex and requires rigorous precision in following every investigative step from Acquisition to Analysis & Reporting. Experts now face the need for dependable tools that help them to do so, from the beginning. Every investigation requires usage of multiple tools, dependence on a sole tool causes the investigation to lose its flexibility and makes it prone towards ambiguity. In this document, we will focus on tools available through General Public License or evaluation versions of professional ones.

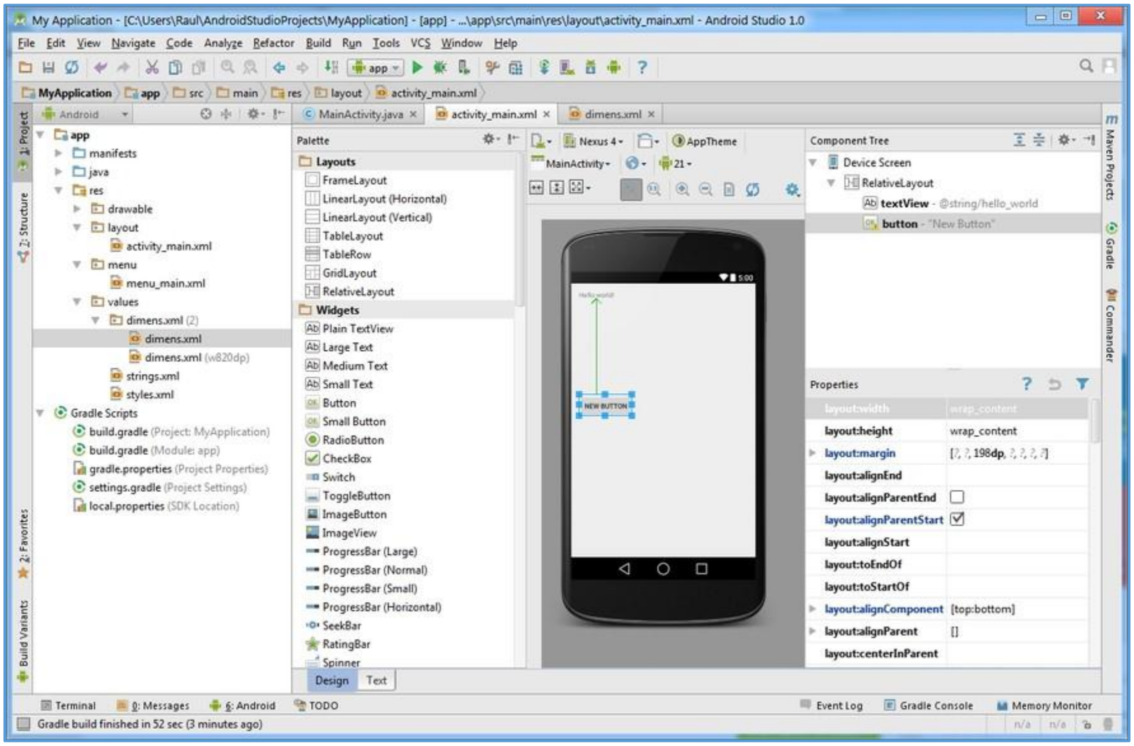
3.1.1 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.[16] Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. The current stable version is 3.0 released in October 2017.

The Android Studio IDE is free to download and use. It has a rich UI development environment with templates to give new developers a launching pad into Android development. Developers will find that Studio gives them the tools to build phone and tablet solutions. Android Studio is intended to be used by development teams as small as one person or as large as global teams. The Android Studio IDE can be linked to larger teams with GIT or similar version control services for larger teams. Mature Android developers will find tools that are necessary for large teams to deliver solutions rapidly to their customers. Android solutions can be developed using either Java or C++ in Android Studio. The workflow for Android Studio is built around the

[16] Wikipedia, The free encyclopedia, *Android Studio*, https://en.wikipedia.org/wiki/Android_Studio.

concept of continuous integration. Continuous Integration allows for teams to test their code each and every time a developer checks in their work. Issues can be captured and reported to the team immediately. The concept of continuously checking code provides actionable feedback to the developers with the goal of releasing versions of a mobile solution faster to the Google Play App Store. To this end, there is rigorous support for LINT tools, Pro-Guard and App Signing tools.


My Application - [C:\Users\Raul\AndroidStudioProjects\MyApplication] - [app] - ...\app\src\main\res\layout\activity_main.xml - Android Studio 1.0

Component Tree
- Device Screen
  - RelativeLayout
    - textView - @string/hello_world
    - button - "New Button"

Properties

layout:width	wrap_content
layout:height	wrap_content
layout:margin	[?, ?, 198dp, ?, ?, ?, ?]
layout:alignEnd	
layout:alignParentEnd	□
layout:alignParentStart	☑
layout:alignStart	
layout:toEndOf	
layout:toStartOf	
layout:alignComponent	[top:bottom]
layout:alignParent	[]
layout:centerInParent	

Gradle build finished in 52 sec (3 minutes ago)

Figure 2: Android Studio GUI

Performance tools provide access to view how well an Android application package file (APK) is going. The performance and profiling tools display a color-coded image to show how often the same pixel is drawn on a screen to reduce rendering overhead. The GPU rendering shows how well your app does in maintaining Google's 16-ms-per-frame benchmark. Memory tools visualize where and when your app will use too much system RAM and when Garbage collection occurs, Battery Analysis tools present how much drain you're placing on a device.[17] Android Studio supports Google App Engine for quick cloud integration of new APIs and features. You will find support for many APIs directly in Android Studio such as Google Play,

[17] Matthew David, Kimberly Clarke, "*Learn more about the Android Studio IDE from Google*" Tech Target, http://searchsoftwarequality.techtarget.com/feature/Learn-more-about-the-Android-Studio-IDE-from-Google.

Android Pay and Health. There is support for all platforms of Android starting with Android 1.6 and later. There are variants of Android that are significantly different to the Google Android version. The most popular is Amazon's Fire OS. Android Studio can be used to build Amazon Fire OS APKs using these guidelines. Android Studio is replacing Google's support for Eclipse ADT.

A very useful feature of Android Studio is the ability to create virtual android devices. The Android Emulator simulates various Android phone, tablet, Android Wear, and Android TV devices. It comes with predefined configurations for popular device types and can transfer data faster than a device connected over USB. The Android Emulator provides almost all the capabilities of a real Android device. The developer can simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors and access the Google Play Store. This is very useful for mobile forensics investigators to test forensic techniques, understand how Android operating system works, identify the essential directories that contain valuable data and finally develop mechanisms and create procedures that help forensics investigation.

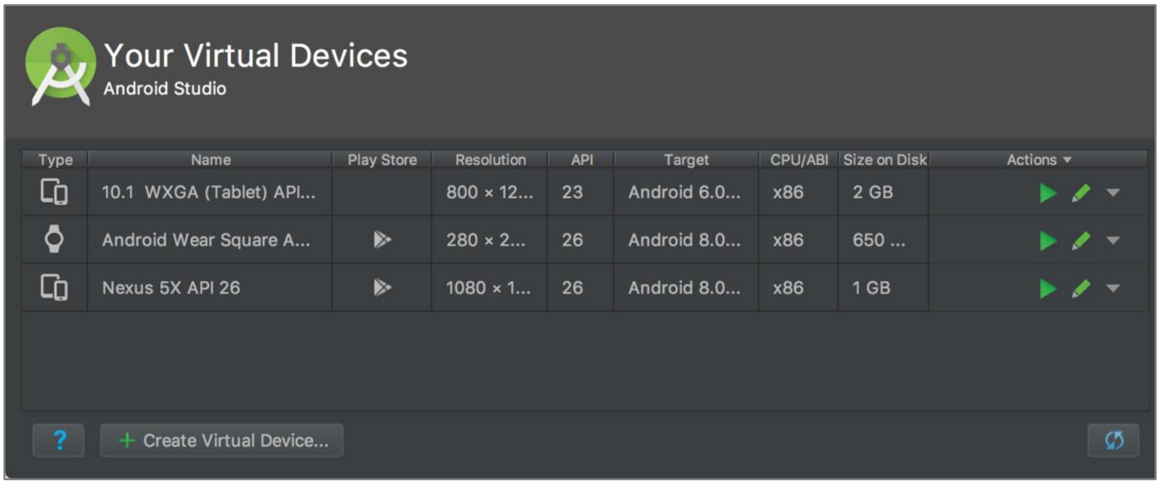
3.1.2 Android Debug Bridge

Android Debug Bridge (ADB)[18] is a versatile command-line tool, included in the Android SDK Platform-Tools package, that lets communicate with a device. The ADB command facilitates a variety of device actions, such as installing and debugging apps, and it provides access to a UNIX shell that you can use to run a variety of commands on a device. It is a client-server program that includes three components:

- A **client**, which sends commands. The client runs on the development machine. Client can be invoked from a command-line terminal by issuing an ADB command.
- A **daemon** (ADBd), which runs commands on a device. The daemon runs as a background process on each device.
- A **server**, which manages communication between the client and the daemon. The server runs as a background process on your development machine.

ADB is included in the Android SDK Platform-Tools package.

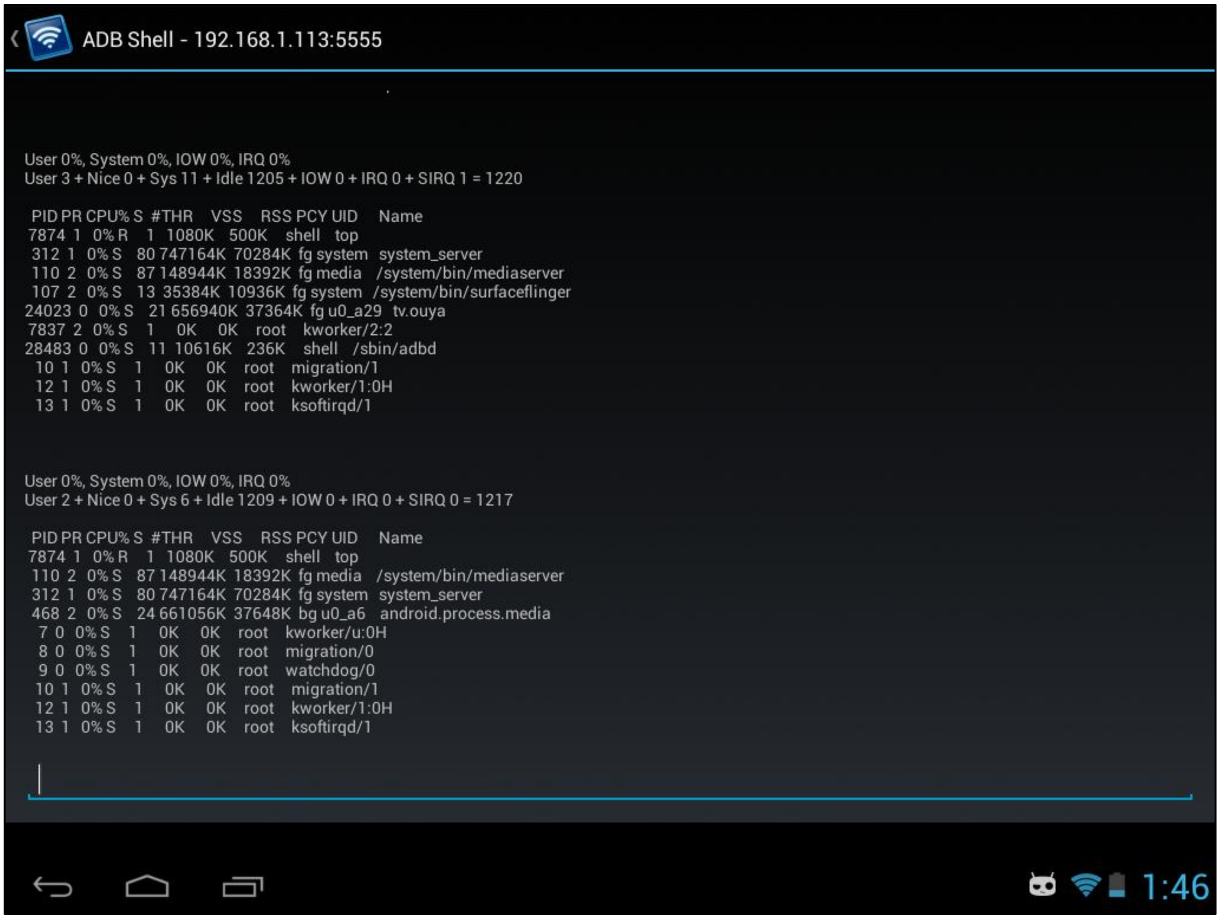
[18] https://developer.android.com/studio/command-line/adb.html.



Your Virtual Devices
Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	10.1 WXGA (Tablet) API...		800 × 12...	23	Android 6.0...	x86	2 GB	▶ ✎ ▾
	Android Wear Square A...	▶	280 × 2...	26	Android 8.0...	x86	650 ...	▶ ✎ ▾
	Nexus 5X API 26	▶	1080 × 1...	26	Android 8.0...	x86	1 GB	▶ ✎ ▾

Figure 3: ADB Virtual Device Manager



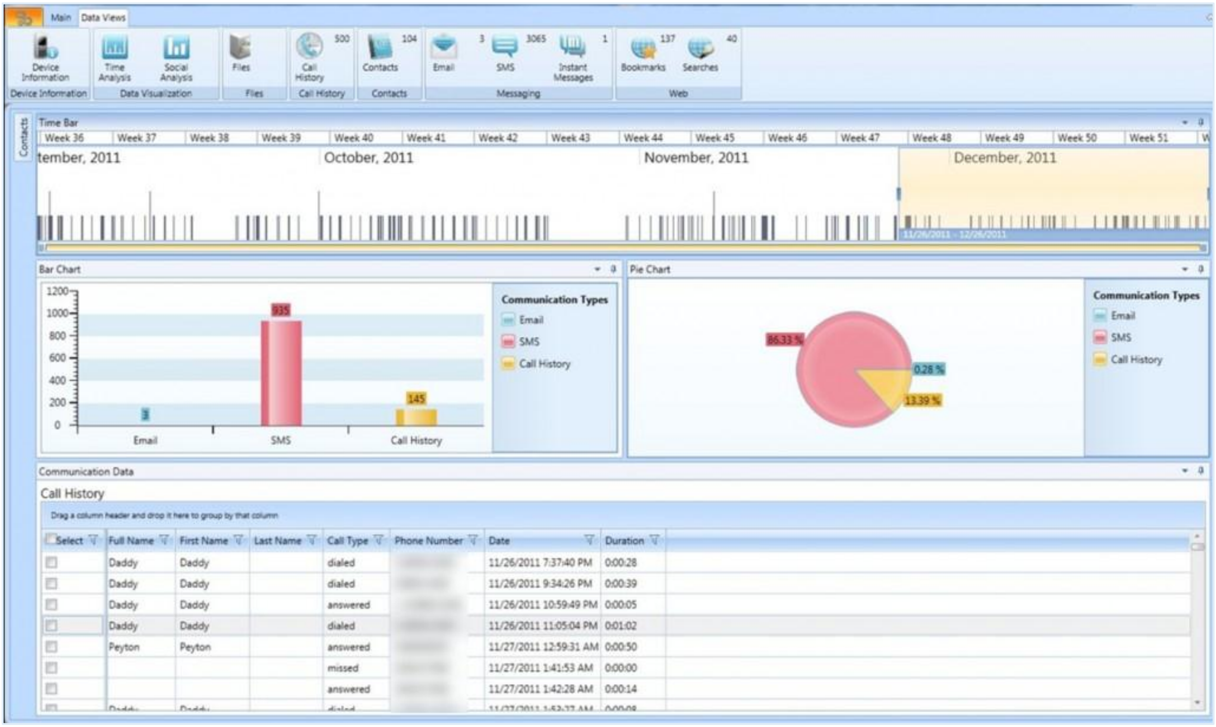
ADB Shell - 192.168.1.113:5555

```
User 0%, System 0%, IOW 0%, IRQ 0%
User 3 + Nice 0 + Sys 11 + Idle 1205 + IOW 0 + IRQ 0 + SIRQ 1 = 1220

 PID PR CPU% S  #THR     VSS     RSS PCY UID      Name
7874  1   0% R     1   1080K    500K     shell    top
 312  1   0% S    80 747164K  70284K  fg system   system_server
 110  2   0% S    87 148944K  18392K  fg media    /system/bin/mediaserver
 107  2   0% S    13  35384K  10936K  fg system   /system/bin/surfaceflinger
24023 0   0% S    21 656940K  37364K  fg u0_a29   tv.ouya
7837  2   0% S     1      0K      0K     root     kworker/2:2
28483 0   0% S    11  10616K    236K     shell    /sbin/adbd
  10  1   0% S     1      0K      0K     root     migration/1
  12  1   0% S     1      0K      0K     root     kworker/1:0H
  13  1   0% S     1      0K      0K     root     ksoftirqd/1


User 0%, System 0%, IOW 0%, IRQ 0%
User 2 + Nice 0 + Sys 6 + Idle 1209 + IOW 0 + IRQ 0 + SIRQ 0 = 1217

 PID PR CPU% S  #THR     VSS     RSS PCY UID      Name
7874  1   0% R     1   1080K    500K     shell    top
 110  2   0% S    87 148944K  18392K  fg media    /system/bin/mediaserver
 312  1   0% S    80 747164K  70284K  fg system   system_server
 468  2   0% S    24 661056K  37648K  bg u0_a6    android.process.media
   7  0   0% S     1      0K      0K     root     kworker/u:0H
   8  0   0% S     1      0K      0K     root     migration/0
   9  0   0% S     1      0K      0K     root     watchdog/0
  10  1   0% S     1      0K      0K     root     migration/1
  12  1   0% S     1      0K      0K     root     kworker/1:0H
  13  1   0% S     1      0K      0K     root     ksoftirqd/1
```

Figure 4: ADB Shell

3.1.3 Mobile Phone Examiner Plus

Mobile Phone Examiner Plus (MPE)[19] is a stand-alone mobile device investigation solution that includes enhanced smart device acquisition and analysis capabilities. With a different approach to digital mobile forensics, MPE allows mobile forensic examiners to take control of the investigation by providing them with unique tools necessary to quickly collect, easily identify and effectively obtain the key data other solutions miss.



Call History

Drag a column header and drop it here to group by that column

Select	Full Name	First Name	Last Name	Call Type	Phone Number	Date	Duration
☐	Daddy	Daddy		dialed	[redacted]	11/26/2011 7:37:40 PM	0:00:28
☐	Daddy	Daddy		dialed	[redacted]	11/26/2011 9:34:26 PM	0:00:39
☐	Daddy	Daddy		answered	[redacted]	11/26/2011 10:59:49 PM	0:00:05
☐	Daddy	Daddy		dialed	[redacted]	11/26/2011 11:05:04 PM	0:01:02
☐	Peyton	Peyton		answered	[redacted]	11/27/2011 12:59:31 AM	0:00:50
☐				missed	[redacted]	11/27/2011 1:41:53 AM	0:00:00
☐				answered	[redacted]	11/27/2011 1:42:28 AM	0:00:14
☐	Daddy	Daddy		dialed	[redacted]	11/27/2011 1:53:27 AM	0:00:08

Figure 5: Mobile Phone Examiner Plus GUI

3.1.4 XRY

XRY[20] is a digital forensics and mobile device forensics used to analyse and recover information from mobile devices such as mobile phones, smartphones, GPS navigation tools and tablet computers. It consists of a hardware device with which to connect phones to a PC and software to extract the data.

[19] https://www.forensicstore.com/product/mobile-phone-examiner-plus/.

[20] Wikipedia, *XRY(software)*, https://en.wikipedia.org/wiki/XRY_(software).

XRY is designed to recover the contents of a device in a forensic manner so that the contents of the data can be relied upon by the user. Typically it is used in civil/ criminal investigations, intelligence operations, data compliance and electronic discovery cases. The software is available to law enforcement, military and intelligence agencies. It has become well known in the digital forensics community as one of their common tools for this type of work.

There are many more complex challenges when examining mobile phones in comparison to the forensic examination of normal computers. Many mobile phones have their own proprietary operating systems, which makes reverse engineering of such devices a very complex operation. The speed of the mobile device market also means that there are many more new devices being manufactured on a regular basis, so a mobile forensics tool must deal with all of these issues before being suitable for the task.

The XRY system allows for both logical examinations (direct communication with the device operating system) and also physical examinations (bypassing the operating system and dumping available memory). Whilst the logical recovery of data is generally better supported for more devices, physical examination offers the ability to recover more deleted information such as SMS text messages, images and call records etc. Because of the complexities of the topic, specialist training is usually recommended to operate the software.

The latest versions include support to recover data from smartphone apps such as the Android, iPhone and Blackberry devices. Data recovered by XRY has been used successfully in various court systems around the world. XRY has been tested by a number of different government organizations as suitable for their needs and is now in worldwide use

3.1.5 Cellebrite UFED

Cellebrite Universal Forensic Extraction Device' (UFED)21 is a mobile forensics product with the ability to extract both physical and logical data from mobile devices such as cellular phones and other hand-held mobile devices, including the ability to recover deleted data and decipher encrypted and password protected information. It extracts mobile device data directly onto an SD card or USB flash drive.

[21] Wikipedia, *Cellebrite,* https://en.wikipedia.org/wiki/Cellebrite.

The UFED is able to extract, decrypt, parse and analyze phonebook contacts, all types of multimedia content, SMS and MMS messages, call logs, electronic serial numbers (ESN), International Mobile Equipment Identity (IMEI) and SIM location information from both non-volatile memory and volatile storage alike. The UFED supports all cellular protocols including CDMA, GSM, IDEN, and TDMA, and can also interface with different operating systems' file systems such as iOS, Android OS, BlackBerry, Symbian, Windows Mobile and Palm as well as legacy and feature cell phones' operating systems. The UFED enables the retrieval of subject data via logical, file system, or physical. Physical extraction enables it to recover deleted information, decipher encrypted data, and acquire information from password-protected mobile applications such as Facebook, Skype, WhatsApp and browser-saved passwords. The UFED's physical extraction functionality can also overcome devices' password locks, as well as SIM PIN numbers.

It is claimed that the UFED maintains the integrity of digital evidence:

- All cable connectors from subject (source) side act as a write blocker, being read-only via the on-board hardware chip set.
- Although a Faraday shielded bag, included in all ruggedized UFED kits, blocks external electromagnetic fields and wireless radio signals, the UFED has a SIM card cloning capability which also isolates the phone from the wireless network.
- Read-only boot loaders keep data from being altered or deleted during a physical extraction.

3.1.6 The Volatility Framework

Volatility[22] is a collection of tools designed to be used as part of incident response and forensic analysis efforts where analyzing volatile memory is necessary or desired. It is free, open source, and written in the Python scripting language. It provides a platform to analyze and extract objects from memory dumps, and supports several operating systems including Linux, OSX 10.5, and Windows.

The Volatility framework supports a wide variety of commands, including commands that list open network connections, print a list of open DLL files, print out the memory map associated

[22] Kristine Amari, *Techniques and Tools for Recovering and Analyzing Data from Volatile Memory*, 26 March 2009.

with the memory dump being analyzed, print a list of the open files associated with a process, and much more. One of its exciting features (particularly for malware analysts) is its ability to reconstruct and write out an executable sample from its associated process.

The Volatility framework is very easy to install and run; simply unpack it onto a system that has Python installed on it and run the command "python volatility". All that is needed to run volatility is a memory dump image, which can be obtained using many different tools, both open source and commercial.

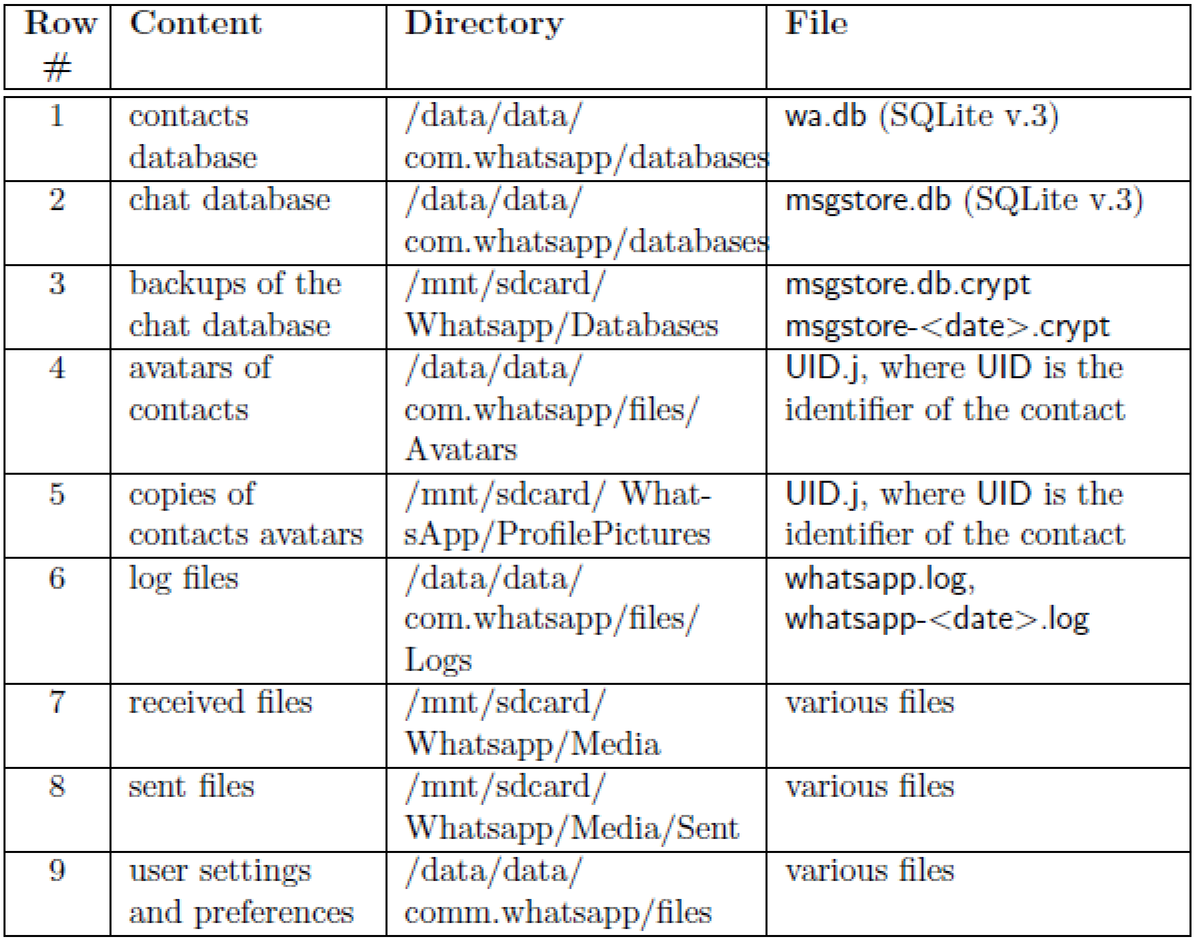
4 Case Study

4.1 Forensic Investigation of WhatsApp Instant Messenger

WhatsApp provides its users with various forms of communications, namely user-to-user communications, broadcast messages, and group chats. When communicating, users may exchange plain text messages, as well as multimedia files (containing images, audio, and video), contact cards, and geolocation information. Each user is associated with its profile, a set of information that includes his/her WhatsApp name, status line, and avatar. The profile of each user is stored on a central system, from which it is downloaded by other WhatsApp users that include that user in their contacts. The central systems provides also other services, like user registration, authentication, and message relay. As reported in the artifacts generated by WhatsApp Messenger on an Android device are stored into a set of files, whose name, location, and contents are listed in **Table 1.**

4.2 Analysis of contact information

The evidentiary value of contact information is notorious, as it allows an investigator to determine who the user was in contact with. In this section we first describe the information that are stored in the contacts database, and then we discuss how this information can be analyzed to determine (a) the list of contacts, (b) when a contact has been added to the database, (c) whether and when a given contact has been blocked and, finally, we show how deleted contacts can be dealt with.[23]



Row #	Content	Directory	File
1	contacts database	/data/data/ com.whatsapp/databases	wa.db (SQLite v.3)
2	chat database	/data/data/ com.whatsapp/databases	msgstore.db (SQLite v.3)
3	backups of the chat database	/mnt/sdcard/ Whatsapp/Databases	msgstore.db.crypt msgstore-<date>.crypt
4	avatars of contacts	/data/data/ com.whatsapp/files/ Avatars	UID.j, where UID is the identifier of the contact
5	copies of contacts avatars	/mnt/sdcard/ WhatsApp/ProfilePictures	UID.j, where UID is the identifier of the contact
6	log files	/data/data/ com.whatsapp/files/ Logs	whatsapp.log, whatsapp-<date>.log
7	received files	/mnt/sdcard/ Whatsapp/Media	various files
8	sent files	/mnt/sdcard/ Whatsapp/Media/Sent	various files
9	user settings and preferences	/data/data/ comm.whatsapp/files	various files

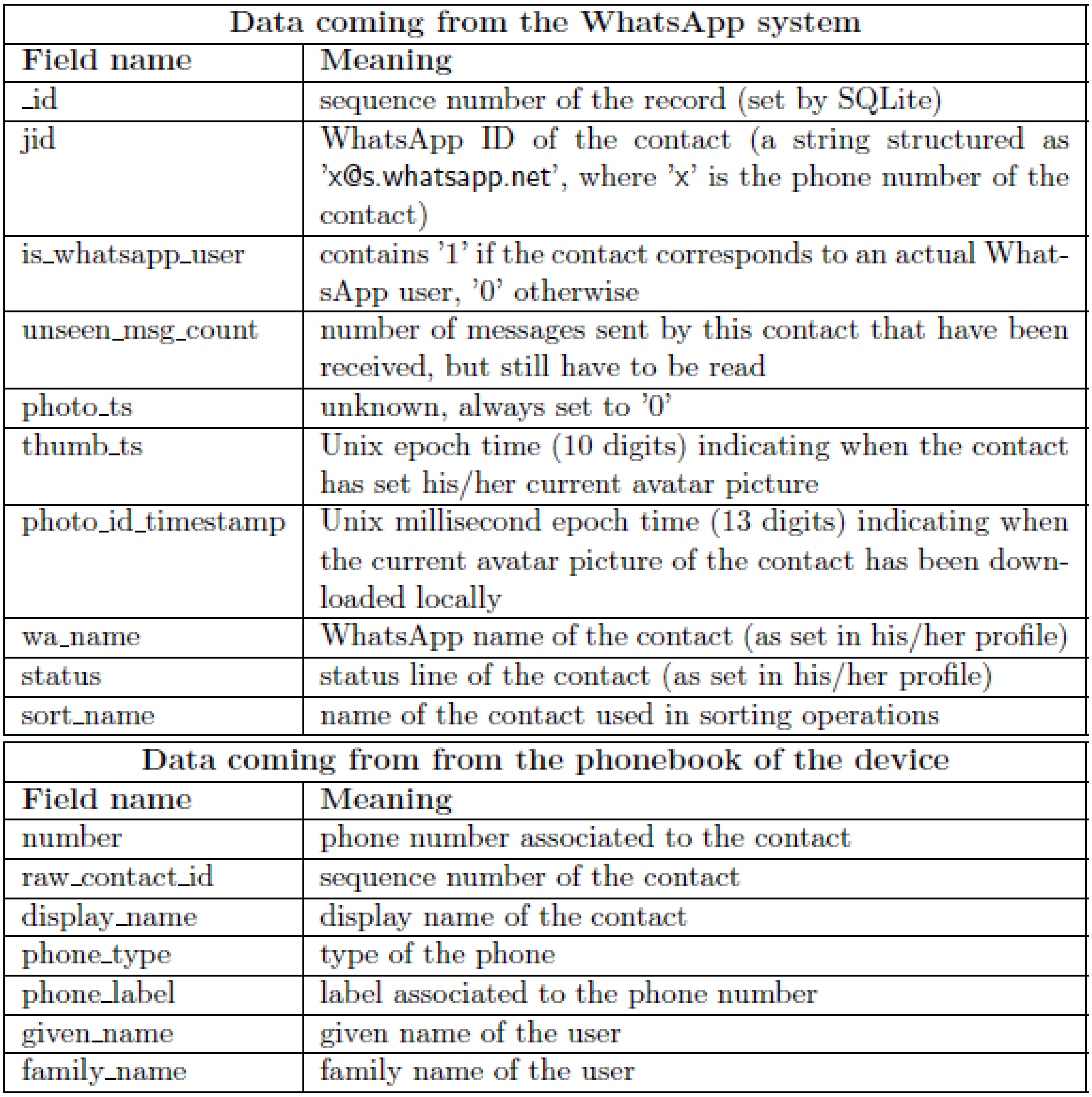
Table 1: WhatsApp Messenger Artifacts

4.2.1 Retrieving contact information

The contacts database wa.db contains three tables, namely wa_contacts, that stores a record for each contact, android metadata, and sqlite sequence, both storing housekeeping information having no evidentiary value. The structure of the records in wa_contacts is shown in **Table 2**, where we distinguish the fields containing data obtained from the WhatsApp system from those storing data extracted from the phonebook of the device.

[23] Cosimo Anglano, "*Forensic Analysis of WhatsApp Messenger on Android Smartphones*", DiSIT - Computer Science Institute, Universita del Piemonte Orientale, Alessandria, Italy, September 2014

As can be observed from this table, each record stores the WhatsApp ID of the contact, a string structured as "x@s.whatsapp.net", where "x" is the phone number of that contact. Furthermore, each record stores the profile name (field wa_name), and the status string (field status) of the corresponding contact. Field is whatsapp user is instead used to differentiate actual WhatsApp users from unreal ones: WhatsApp Messenger indeed adds to the contact database a record for each phone number found in the phonebook of the device, even if the corresponding user is not registered with the WhatsApp system.m the WhatsApp system from those storing data extracted from the phonebook of the device.



Data coming from the WhatsApp system	
Field name	**Meaning**
_id	sequence number of the record (set by SQLite)
jid	WhatsApp ID of the contact (a string structured as 'x@s.whatsapp.net', where 'x' is the phone number of the contact)
is_whatsapp_user	contains '1' if the contact corresponds to an actual WhatsApp user, '0' otherwise
unseen_msg_count	number of messages sent by this contact that have been received, but still have to be read
photo_ts	unknown, always set to '0'
thumb_ts	Unix epoch time (10 digits) indicating when the contact has set his/her current avatar picture
photo_id_timestamp	Unix millisecond epoch time (13 digits) indicating when the current avatar picture of the contact has been downloaded locally
wa_name	WhatsApp name of the contact (as set in his/her profile)
status	status line of the contact (as set in his/her profile)
sort_name	name of the contact used in sorting operations

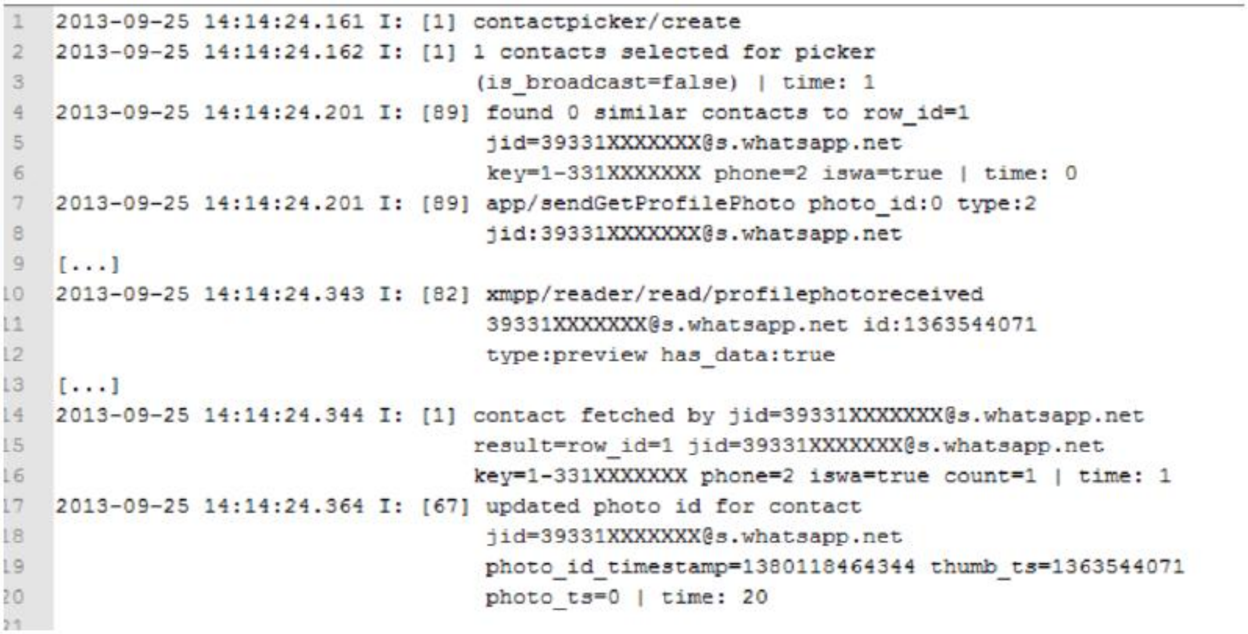
Data coming from from the phonebook of the device	
Field name	**Meaning**
number	phone number associated to the contact
raw_contact_id	sequence number of the contact
display_name	display name of the contact
phone_type	type of the phone
phone_label	label associated to the phone number
given_name	given name of the user
family_name	family name of the user

Table 2: Structure of the wa_contacts table

Avatar pictures may have evidentiary value as well: they can be indeed used to link a WhatsApp account to the real identity of the person using it. The avatar picture of a contact **x@s.whatsapp.net** is stored, as a JPEG file named **x@s.whatsapp.net.j**, in the directories listed in Table 1, rows no. 4 and 5. The timestamps stored in the thumb ts and photo id timestamp field indicate when the contacts has set his/her current avatar, and when that avatar has been downloaded locally, respectively.

4.2.2 Determining when a contact has been added

In some investigations, it may be necessary to determine when a given user has been added to the contacts database. User contacts are automatically added to the contacts database by WhatsApp Messenger that, each time is started or when the user starts a new conversation, inspects the phonebook of the device and adds all the phone numbers that are not stored there yet. This information is not stored in the wa_contacts table, but can be deduced from the analysis of the log files generated by WhatsApp Messenger.



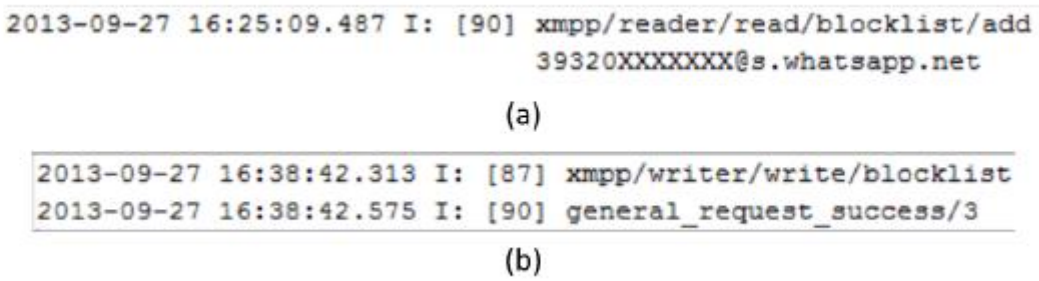
```
2013-09-25 14:14:24.161 I: [1] contactpicker/create
2013-09-25 14:14:24.162 I: [1] 1 contacts selected for picker
                               (is_broadcast=false) | time: 1
2013-09-25 14:14:24.201 I: [89] found 0 similar contacts to row_id=1
                                jid=39331XXXXXXX@s.whatsapp.net
                                key=1-331XXXXXXX phone=2 iswa=true | time: 0
2013-09-25 14:14:24.201 I: [89] app/sendGetProfilePhoto photo_id:0 type:2
                                jid:39331XXXXXXX@s.whatsapp.net
[...]
2013-09-25 14:14:24.343 I: [82] xmpp/reader/read/profilephotoreceived
                                39331XXXXXXX@s.whatsapp.net id:1363544071
                                type:preview has_data:true
[...]
2013-09-25 14:14:24.344 I: [1] contact fetched by jid=39331XXXXXXX@s.whatsapp.net
                               result=row_id=1 jid=39331XXXXXXX@s.whatsapp.net
                               key=1-331XXXXXXX phone=2 iswa=true count=1 | time: 1
2013-09-25 14:14:24.364 I: [67] updated photo id for contact
                                jid=39331XXXXXXX@s.whatsapp.net
                                photo_id_timestamp=1380118464344 thumb_ts=1363544071
                                photo_ts=0 | time: 20
```

Figure 6: Events logged when a user is added to the contacts table

When a contact is added to the wa.db database, WhatsApp Messenger logs several events that are tagged with their time of occurrence and with the WhatsApp ID of the involved user. Examples of these events, corresponding to the addition of user 39331xxxxxxx, are reported in **Figure 6**, from which we note that the following events are logged each time a new user is added: (a) the discovery that the user is not present yet in the contacts database (line no. 4), (b) the queries to the central system to fetch various information about the contact (lines no. 7,10, and 14), and (c) the completion of the download of the corresponding avatar picture (line no. 17). From these events, we can determine when the user has been added to the contacts database.

4.2.3 Blocked Contacts Handling

WhatsApp Messenger enables the user to block anyone of his/her contacts, thus preventing any communication with him/her until the block is removed. In an investigation it can be important to determine whether a contact was blocked or not at a given time, in order to confirm or to exclude the reception of a message sent at that time. The information concerning blocked users is stored neither in the contacts database, nor elsewhere on the memory of the device. Blocked users can be however identified, under some circumstances, by analyzing log files. When a contact is blocked, an event, reporting the WhatsApp ID of that contact and the time of occurrence of the operation, is indeed recorded into the log file (**Figure 7a**). Unfortunately, when a contact is unblocked, the event that is logged (**Figure 7b**) does not report the WhatsApp IDs of the involved contact, and it is cumulative (i.e., it may refer to a set of contacts being unblocked simultaneously).



```
2013-09-27 16:25:09.487 I: [90] xmpp/reader/read/blocklist/add
                                39320XXXXXXX@s.whatsapp.net
```

(a)

```
2013-09-27 16:38:42.313 I: [87] xmpp/writer/write/blocklist
2013-09-27 16:38:42.575 I: [90] general_request_success/3
```

(b)

Figure 7: Events in the log file corresponding to (a) the blocking, (b) the unblocking of user 39320xxxxxxx

4.2.4 Dealing with deleted contacts

In the attempt to hide past interactions, the user may delete a contact, thus causing the removal of the corresponding record from the wa_contacts table. In some cases, if the SQLite engine has not vacuumed the above table yet, it may be possible to recover deleted records by means of suitable techniques. This may be accomplished using Oxygen Forensic SQLite Viewer, indicating indeed that deleted contact records may be recovered. However, if at the moment of the analysis the deleted records have been vacuumed, they cannot be recovered. In these situations, it may be still possible to determine the set of deleted contacts by reconstructing the list of contacts that have been added in the past and then by comparing this list with the contents of the wa_contacts table. The contacts in the list that are not in the database are those that have been deleted. Note that this procedure works only if the log file reporting the addition of a contact of interest is still available when the analysis is performed.

Unfortunately, by proceeding as above, it is not possible to determine when a given contact has been deleted, since deletions give rise to log events that do not reference the WhatsApp ID of the contact being deleted.

4.3 Analysis of exchanged messages

WhatsApp Messenger stores all the messages that have been sent or received into the chat database **msgstore.db** (located in the directory listed in **Table 1**, row 2), whose analysis makes it possible to reconstruct the chronology of exchanged messages, namely to determine when a message has been exchanged, the data it carried, the set of users involved in the conversation, and whether and when it has actually been received by its recipients.

In the following we discuss each one of the above steps separately: we start with the description of the structure of the chat database, and then we explain how to (a) reconstruct the chat history, (b) determine and extract the content of a message, (c) determine the status of a message, (d) determine the set of users among which each message has been exchanged and finally, (e) deal with deleted messages.

4.3.1 The structure of the chat database

The **msgstore.db** database contains the following three tables:

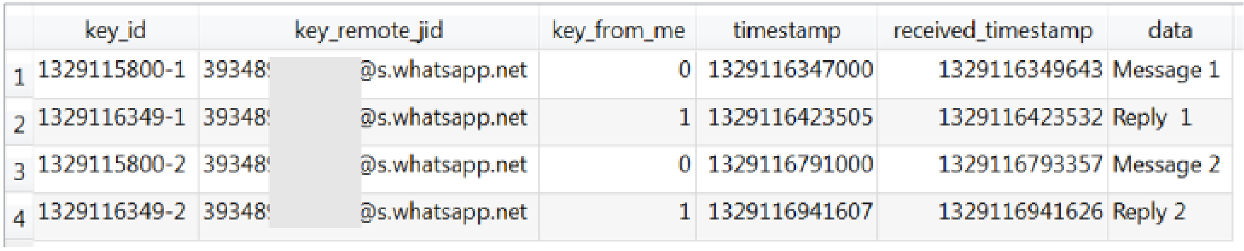
- Messages, which contains a record for each message that has been sent or received by the user. To ease understanding, we classify the fields of these records in two distinct categories: those storing attributes of the message (listed in **Table 3**), and those storing the contents of the message and the corresponding metadata (listed in **Table 4**).

- Chat list, that contains a record for each conversation held by the user (a conversation consists into the set of messages exchanged with a particular contact), whose fields are described in **Table 5.**
- SQLite sequence, which stores housekeeping data used internally by WhatsApp Messenger, whose structure is not reported here since it does not have any evidentiary value.

As reported in "Forensic Analysis of WhatsApp on Android Smartphones" (Thakur, 2013)[24], WhatsApp Messenger usually generates various backup copies of the msgstore.db database, that are stored in the directory listed in **Table 1** row no. 3. These backups are full copies of the **msgstore.db** database, and are not kept synchronized with it. Therefore, they are particularly important from an investigative standpoint, since they may store messages that have been deleted from the main chat database. Backups are encrypted with the AES 192 algorithm, but they can be easily decrypted since, the same encryption key is used on all devices (Cortjens et al., 2011).[25]

4.3.2 Reconstruction of the chat history

To reconstruct the chronology of the messages exchanged by the user, the records stored in the messages table must be extracted and decoded as discussed below. To elucidate, let us consider **Figure 8**, that shows four records corresponding to a conversation between the device owner and the user 39348xxxxxxx(actually, only the fields listed in **Table 3** are displayed).



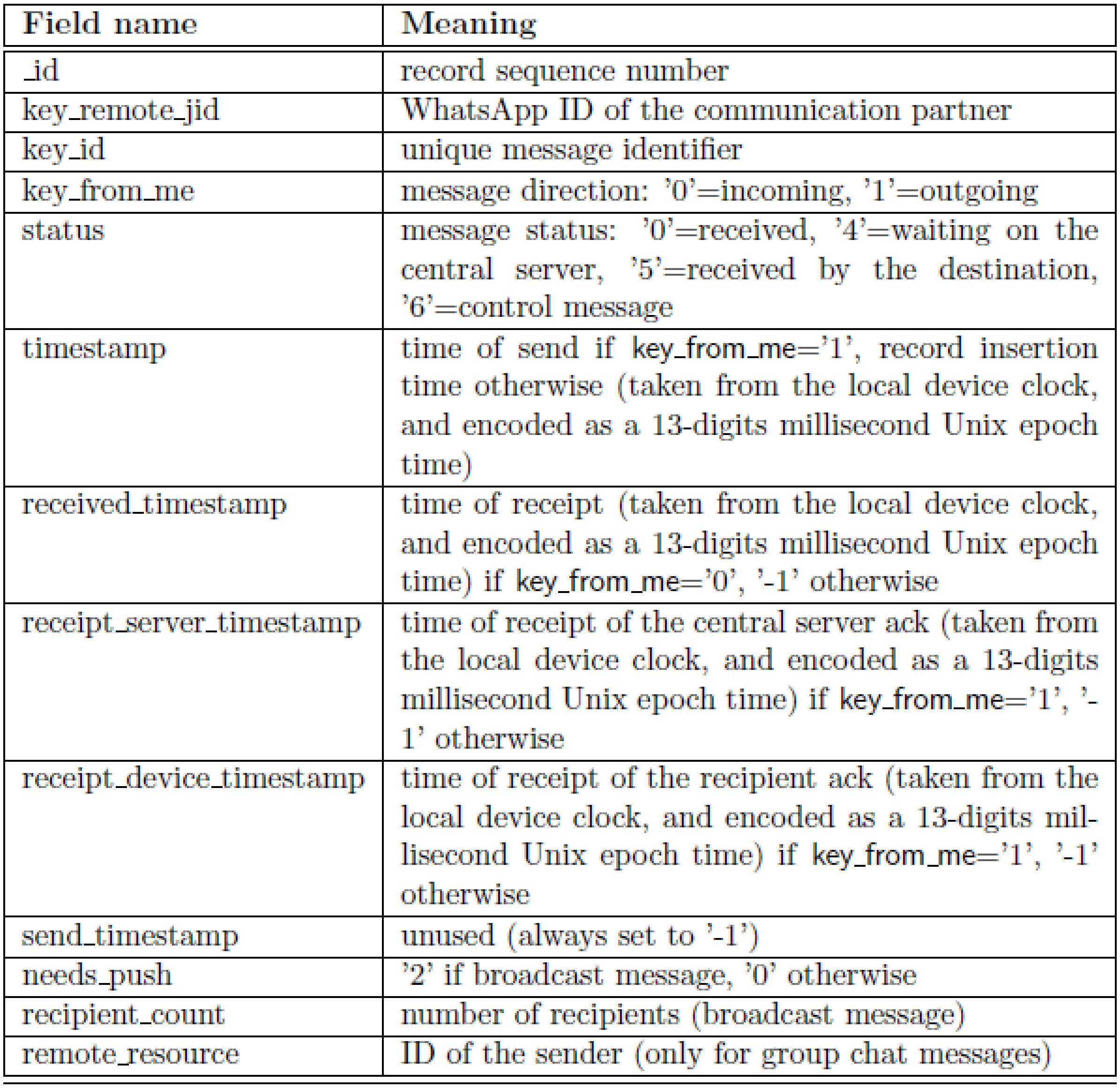
	key_id	key_remote_jid	key_from_me	timestamp	received_timestamp	data
1	1329115800-1	39348[redacted]@s.whatsapp.net	0	1329116347000	1329116349643	Message 1
2	1329116349-1	39348[redacted]@s.whatsapp.net	1	1329116423505	1329116423532	Reply 1
3	1329115800-2	39348[redacted]@s.whatsapp.net	0	1329116791000	1329116793357	Message 2
4	1329116349-2	39348[redacted]@s.whatsapp.net	1	1329116941607	1329116941626	Reply 2

Figure 8: Reconstruction of the chat history. Phone numbers have been grayed out to ensure the privacy of the owner

[24] N.S. Thakur. "*Forensic Analysis of WhatsApp on Android Smartphones*", Master's thesis, University of New Orleans, 2013. Paper 1706

[25] D. Cortjens, A. Spruyt, W.F.C. Wieringa, "WhatsApp Database Encryption Project Report", Technical report, 2011, https://www.os3.nl/media/2011-2012/students/ssn project report.pdf

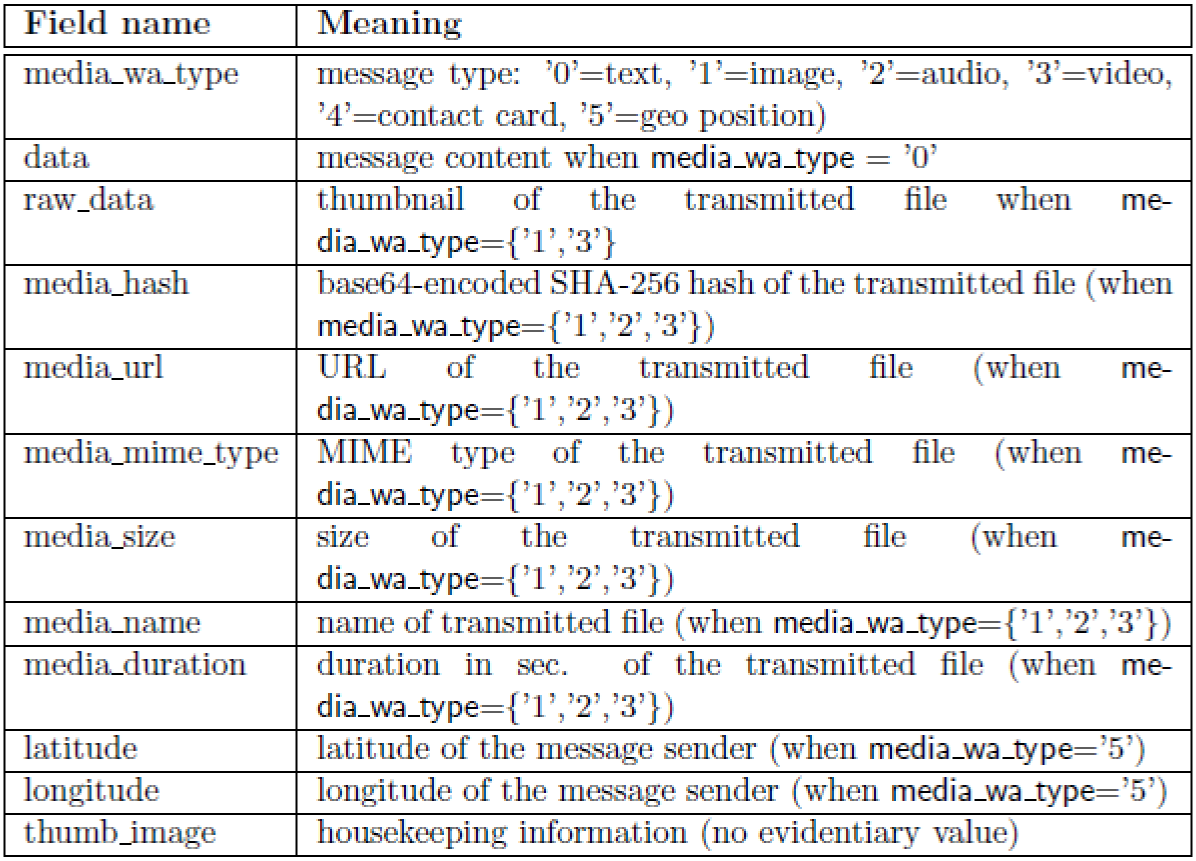
By examining these records, we note that (a) all the messages have been exchanged with the same contact 39348xxxxxxx (they all store the same WhatsApp ID in the key remote field), (b) the conversation has been started by that contact (key from me = '0' in record no.1) with a textual message whose content was \Message 1" (field data) on Feb. 13th, 2012 06:59:09 (field received timestamp), and (c) the device owner replied at 07:00:23 of the same day (field timestamp) with the message corresponding to record no. 2 (key from me='1') with content \Reply 1" (field data).



Field name	Meaning
_id	record sequence number
key_remote_jid	WhatsApp ID of the communication partner
key_id	unique message identifier
key_from_me	message direction: '0'=incoming, '1'=outgoing
status	message status: '0'=received, '4'=waiting on the central server, '5'=received by the destination, '6'=control message
timestamp	time of send if key_from_me='1', record insertion time otherwise (taken from the local device clock, and encoded as a 13-digits millisecond Unix epoch time)
received_timestamp	time of receipt (taken from the local device clock, and encoded as a 13-digits millisecond Unix epoch time) if key_from_me='0', '-1' otherwise
receipt_server_timestamp	time of receipt of the central server ack (taken from the local device clock, and encoded as a 13-digits millisecond Unix epoch time) if key_from_me='1', '-1' otherwise
receipt_device_timestamp	time of receipt of the recipient ack (taken from the local device clock, and encoded as a 13-digits millisecond Unix epoch time) if key_from_me='1', '-1' otherwise
send_timestamp	unused (always set to '-1')
needs_push	'2' if broadcast message, '0' otherwise
recipient_count	number of recipients (broadcast message)
remote_resource	ID of the sender (only for group chat messages)

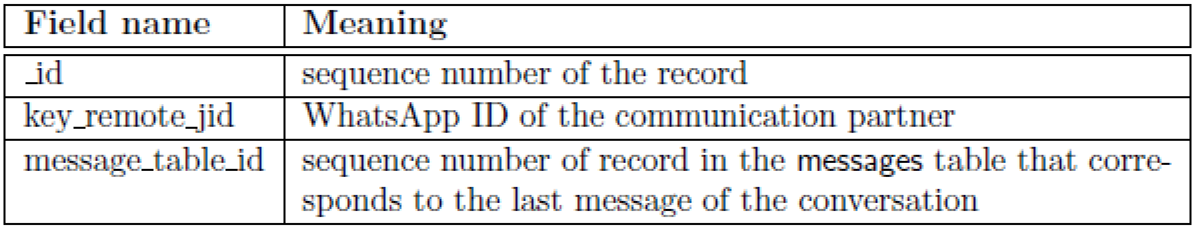
Table 3: Structure of the messages table - fields storing message attributes

The conversation then continued with another message-reply exchange. From these records, we also note that each message carries its own unique identifier in the key id field: this value, set by the sender, is obtained by concatenating the timestamp corresponding to the last start time of WhatsApp Messenger (on the sender's device) with a progressive number (indicating the number of messages sent from that moment), and is used also by the recipient to denote that message. Therefore, by using this value, it is possible to correlate the records of the sender's and recipient's databases corresponding to the same message.



Field name	Meaning
media_wa_type	message type: ’0’=text, ’1’=image, ’2’=audio, ’3’=video, ’4’=contact card, ’5’=geo position)
data	message content when media_wa_type = ’0’
raw_data	thumbnail of the transmitted file when media_wa_type={’1’,’3’}
media_hash	base64-encoded SHA-256 hash of the transmitted file (when media_wa_type={’1’,’2’,’3’})
media_url	URL of the transmitted file (when media_wa_type={’1’,’2’,’3’})
media_mime_type	MIME type of the transmitted file (when media_wa_type={’1’,’2’,’3’})
media_size	size of the transmitted file (when media_wa_type={’1’,’2’,’3’})
media_name	name of transmitted file (when media_wa_type={’1’,’2’,’3’})
media_duration	duration in sec. of the transmitted file (when media_wa_type={’1’,’2’,’3’})
latitude	latitude of the message sender (when media_wa_type=’5’)
longitude	longitude of the message sender (when media_wa_type=’5’)
thumb_image	housekeeping information (no evidentiary value)

Table 4: Structure of the messages table - fields storing information concerning message contents



Field name	Meaning
_id	sequence number of the record
key_remote_jid	WhatsApp ID of the communication partner
message_table_id	sequence number of record in the messages table that corresponds to the last message of the conversation

Table 5: Structure of the chat list table.

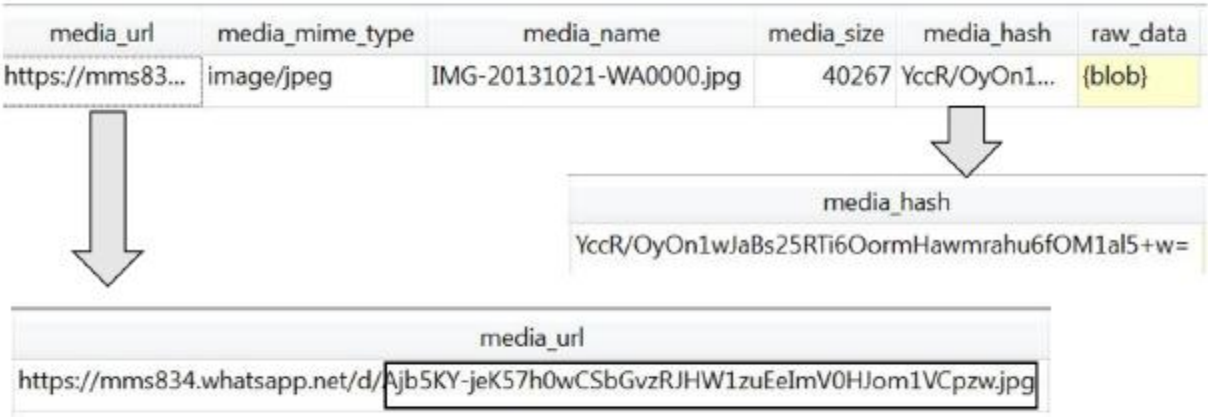
4.3.3 Extracting the contents of a message

In addition to plain text messages, WhatsApp allows its users to exchange messages containing data of various types, namely multimedia files (storing images, audio, and video), contact cards, and geolocation information. The type of data transmitted with a message is indicated (as reported in **Table 4**) by the media_wa_type field, while the information concerning message content is spread, for non-textual messages, over several fields (depending on the specific data type). As a matter of fact, while the content of textual messages (media_wa_type='0') is stored in the data field, for the other types of contents the situation is more involved, as discussed below.

Multimedia files

When the user sends a multimedia file, several activities take place automatically. First, WhatsApp Messenger copies the file into the folder listed in **Table 1**, row 8. Then, it uploads the file to the WhatsApp server, which sends back the URL of the corresponding location. Finally, the sender sends to the recipient a message containing this URL and, upon receiving this message, the recipient sends an acknowledgment back to the sender.

When these steps have been completed, the sender stores into his/her messages table a record like the one shown in Figure 9.



media_url	media_mime_type	media_name	media_size	media_hash	raw_data
https://mms83...	image/jpeg	IMG-20131021-WA0000.jpg	40267	YccR/OyOn1...	{blob}

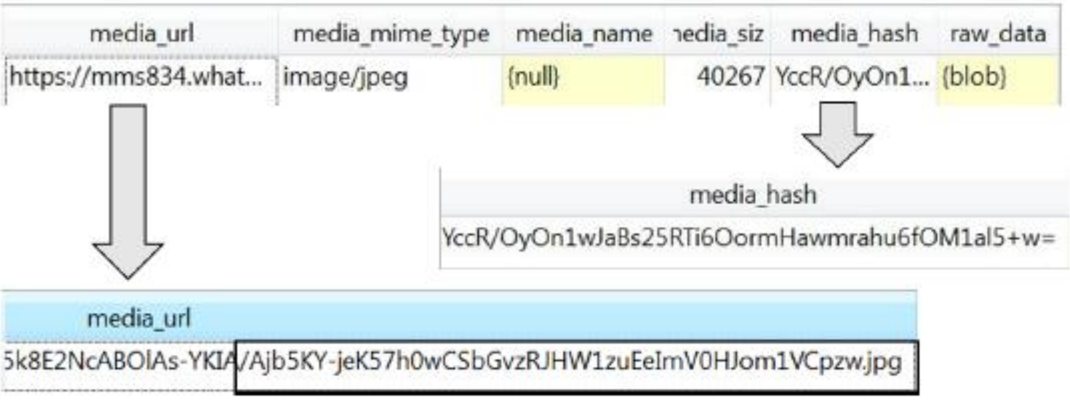
media_hash
YccR/OyOn1wJaBs25RTi6OormHawmrahu6fOM1al5+w=

media_url
https://mms834.whatsapp.net/d/Ajb5KY-jeK57h0wCSbGvzRJHW1zuEeImV0HJom1VCpzw.jpg

Figure 9: Multimedia file exchange - sender side

As can be seen from the above figure, the type of the file is indicated by the media_mime_type field ('image/jpeg' in the example).Its name is instead stored in the media_name field (IMG-20131021-WA0000.jpg in the example), its size in bytes by media_size (40267 in the example), and its thumbnail in the raw_data field (as a blob, i.e. a binary large object). Furthermore, the media_url field stores the URL of the location on the central server where the file has been temporarily stored, whose last part (highlighted in **Figure 9** by framing it) corresponds to the name given by the server to that file. Finally, the base64-encoded SHA-256 hash of the transmitted file is stored in the media hash_field.

On the recipient side, after message reception, the transmitted thumbnail of the file is displayed by WhatsApp Messenger. The actual file is instead downloaded at a later time only if the recipient explicitly requests it. Upon message reception, the recipient stores in his/her messages table a record like the one shown in **Figure 10**.



media_url	media_mime_type	media_name	ıedia_siz	media_hash	raw_data
https://mms834.what...	image/jpeg	{null}	40267	YccR/OyOn1...	{blob}

media_hash
YccR/OyOn1wJaBs25RTi6OormHawmrahu6fOM1al5+w=

media_url
5k8E2NcABOlAs-YKIA/Ajb5KY-jeK57h0wCSbGvzRJHW1zuEeImV0HJom1VCpzw.jpg

Figure 10: Multimedia file exchange - recipient side

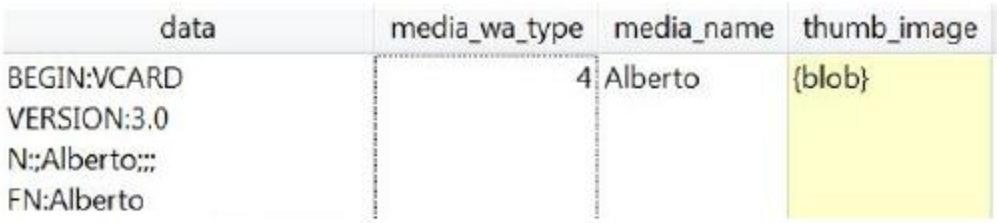
From this figure, we see that most fields are identical to those stored by the sender (in particular, wa_media_type, media_mime_type,

media_size, raw_data, and media_hash). Conversely, the contents of media url is different, except for the name given to the file by the server (highlighted in Figure 10 by framing it). Unlike the sender, however, the media name field is empty, so the local name given by WhatsApp Messenger to that file is unknown. The file can be however identified by comparing the SHA-256 hash stored into the corresponding record with that of all the files that have been received (that are stored in the folder reported in **Table 1**, row 7.

Finally, we note that the file sent by the sender and the one received by the recipient can be correlated by comparing both the file name given by the WhatsApp server and the SHA-256 hash to these files (that are stored, as discussed above, in the media_url and media_hash fields of the corresponding records).

Contact cards

Messages carrying contacts cards (extracted from the phonebook of the sender) correspond, both on the sender and on the recipient side, to messages record that store the transmitted information (in VCARD format) into the data field, and the name given by the sender to that contact in the media_name field. An example of such a record is shown in **Figure 11**.

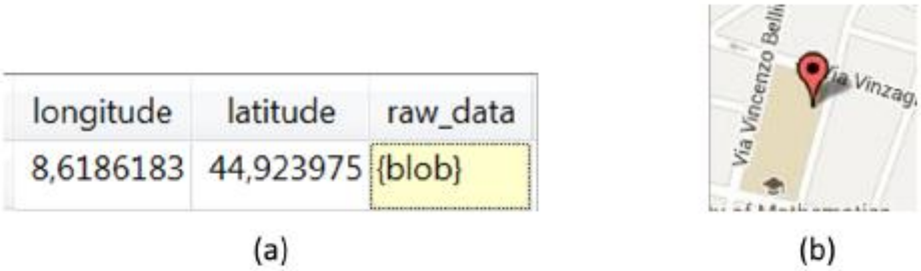


data	media_wa_type	media_name	thumb_image
BEGIN:VCARD VERSION:3.0 N:;Alberto;;; FN:Alberto	4	Alberto	{blob}

Figure 11: Fields containing contact card information

Geolocation coordinates

WhatsApp Messenger enables users to send the geographic coordinates of their current location that are obtained from the Android Location Services running on the device. Messages carrying geographic coordinates correspond, both on the sender and on the recipient side, to messages records that store the latitude and the longitude values into the latitude, longitude fields, and a JPEG thumbnail of the Google Map displaying the above coordinates in the raw data field. An example of such a record is shown in **Figure 12.**



longitude	latitude	raw_data
8,6186183	44,923975	{blob}

Figure 12: Geolocation message:(a) data stored in the database, (b) Google map extracted from the raw data field

4.4.4 Determining the state of the message

In WhatsApp, messages are not exchanged directly among communicating users, but they are first sent to the central server, that forwards them to the respective recipients if they are on-line, and stores them locally until they can be delivered otherwise. This implies that the presence of a record in the messages table does not necessarily mean that an outgoing message has been actually delivered to its recipients. As a matter of fact, after the user has pushed the "send" button of WhatsApp Messenger, the message can be in one of the following three states: (a) waiting on the local device to be transmitted to the central server, or (b) stored on the central server but waiting to be transmitted to its recipient(s), or (c) delivered to its recipient(s).

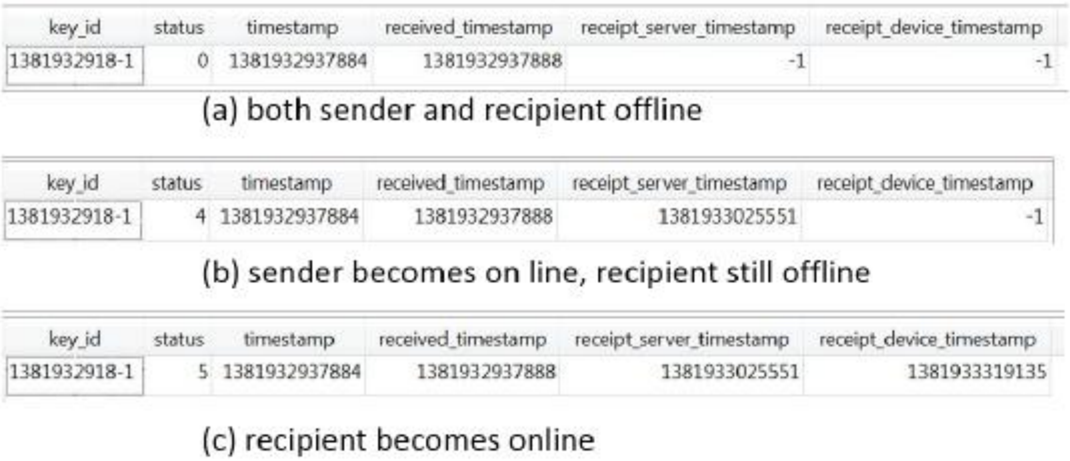
The ability to distinguish the various states of a message may be crucial in an investigation where it must be ascertained whether a message has been actually delivered or not to its destinations, and when such a delivery has taken place.

The current state of a message, as well as the times of its state changes, can be understood by correlating the values contained in several fields of the corresponding record of the sender database 3, namely status, timestamp, received_timestamp, receipt_server_timestamp, and receipt_device_timestamp.

To explain, let us consider a scenario in which a user sends a message when both him/her and the recipient are off-line (**Figure 13**(a)), then the sender gets reconnected to the network while the recipient is still offine (**Figure 13** (b)), and then, finally, also the recipient gets connected (**Figure 13** (c)).

When the message is sent, a record is stored in the messages table of the sender, even if the central server is unreachable. In this case, as shown in **Figure 13**(a), in this record we have that status='0', timestamp='x', and re- ceived timestamp='y', where 'x' and 'y' correspond to when the user has sent the message and when the record has been added to the chat database, respectively.

Later, when the sender returns on-line, the message is forwarded to the central server that replies with an ack. When this ack is received, the sender updates the corresponding record as shown in **Figure 13**(b) by setting status='4', and the value of receipt server timestamp to the reception time of the ack.



key_id	status	timestamp	received_timestamp	receipt_server_timestamp	receipt_device_timestamp
1381932918-1	0	1381932937884	1381932937888	-1	-1

(a) both sender and recipient offline

key_id	status	timestamp	received_timestamp	receipt_server_timestamp	receipt_device_timestamp
1381932918-1	4	1381932937884	1381932937888	1381933025551	-1

(b) sender becomes on line, recipient still offline

key_id	status	timestamp	received_timestamp	receipt_server_timestamp	receipt_device_timestamp
1381932918-1	5	1381932937884	1381932937888	1381933025551	1381933319135

(c) recipient becomes online

Figure 13: Sender side - record updates for a message while in transit

Finally, when the recipient returns on line, it receives the message from the central server, and sends an ack to the sender. Upon receiving this ack, the sender updates again the record corresponding to that message (as shown in **Figure 13**(c)) by setting status='5', and the value of receipt device timestamp to the reception time of the ack.

4.4.5 Determining the partners of a message

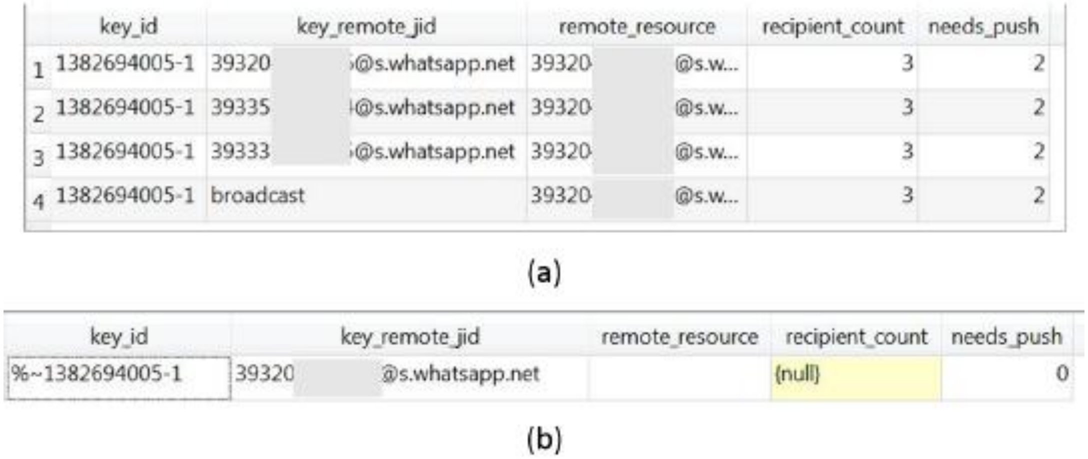
In addition to user-to-user communication, WhatsApp provides its users with two forms of collective communications, namely:

- Broadcast (i.e, one-to-many) communication, whereby a user (the source user) sends the same message to a group of other users (the destination users) that are not aware of each other and whose possible replies are sent to the source user only
- Group chats, providing a many-to-many communication service, whereby each message sent by any user belonging to a chat is received by all the users belonging to that chat

While the WhatsApp ID of the communication partner in a user-to-user communication is easily retrieved from the key_remote_jid field, to determine the set of users involved into a broadcast or a group chat message various fields have to be correlated, as discussed below.

Broadcast messages

When a user sends a broadcast message, a distinct record is created in his/her messages table for each one of the recipients, plus one for itself, as reported in **Figure 14**(a), that shows the records generated by a broadcast message sent to users 39320xxxxxxx, 39335xxxxxxx, and 39333xxxxxxx.



	key_id	key_remote_jid	remote_resource	recipient_count	needs_push
1	1382694005-1	39320[redacted]@s.whatsapp.net	39320[redacted]@s.w...	3	2
2	1382694005-1	39335[redacted]@s.whatsapp.net	39320[redacted]@s.w...	3	2
3	1382694005-1	39333[redacted]@s.whatsapp.net	39320[redacted]@s.w...	3	2
4	1382694005-1	broadcast	39320[redacted]@s.w...	3	2

(a)

key_id	key_remote_jid	remote_resource	recipient_count	needs_push
%~1382694005-1	39320[redacted]@s.whatsapp.net		(null)	0

(b)

Figure 14: Records generated for a broadcast message sent to three recipients on: (a) the sender, (b) one of the recipients. Only the fields that contribute to the identification of the partners are displayed

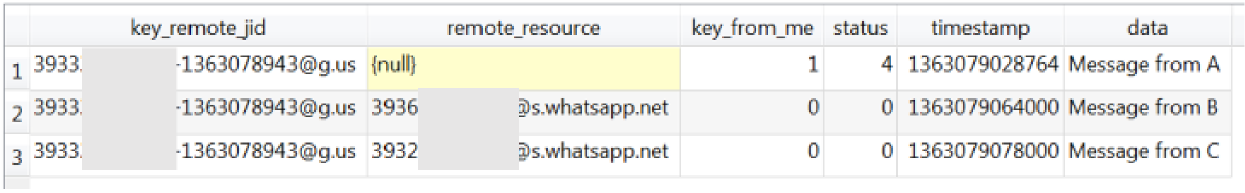
As shown in this figure, all the records corresponding to the same broadcast message have the same message identifier (stored in the key id field), so they can be easily identified. Each one of these records stores in the key_remote_jid field the WhatsApp ID of the recipient (the sender uses the keyword broadcast to denote itself as a recipient), while the remote resource and the recipient count fields store the WhatsApp IDs of the set of destinations and how many they are, respectively (field needs push instead always stores the value '2').

The situation on each one of the destinations is instead different (**Figure 14**(b)), since each one of them stores, in his/her messages table, only a single record that is generated when it receives the broadcast message. This record can be distinguished from those corresponding to non-broadcast messages by look- ing at the value stored in its key id field, that consists in the concatenation of the %~ characters with the message identifier set by the sender.

Group chat communication

When a message is sent within a group chat, a record is generated in the messages table of all the members of that group (including the sender). Each one of these records stores, in the key_remote_jid field, the identifier of the group (the group_id), a string formatted as {creator's phone number}-{creation time}@g.us (where creation time is encoded as a UNIX epoch time).

To illustrate, consider a group chat consisting of three members, namely 3933xxxxxxx, 3936xxxxxxx, and 3932xxxxxx (in the following denoted as A, B, and C, respectively, for brevity), where each user sends in turn to the group a message with textual content 'Message from X' (where 'X' is the name of the user). Let us focus on the records stored in the messages table of user A at the end of this exchange, that are shown in **Figure 15** (the situation for the other users is identical).



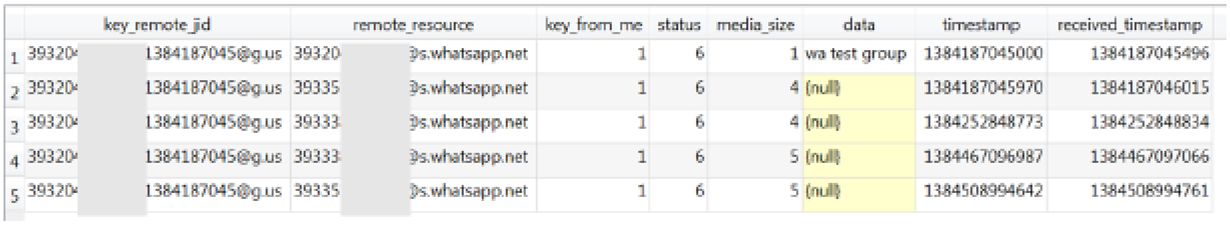
	key_remote_jid	remote_resource	key_from_me	status	timestamp	data
1	3933[redacted]-1363078943@g.us	{null}	1	4	1363079028764	Message from A
2	3933[redacted]-1363078943@g.us	3936[redacted]@s.whatsapp.net	0	0	1363079064000	Message from B
3	3933[redacted]-1363078943@g.us	3932[redacted]@s.whatsapp.net	0	0	1363079078000	Message from C

Figure 15: Records corresponding to three messages exchanged within a group chat.

As can be seen from this figure, all these records store the same group id 3933xxxxxxx-1363078943@g.us in the key_remote_id field. From this value, we can determine the creator of the group (user A) and the date and hour of group creation (March 12, 2013 at 09:02:23). Furthermore, the WhatsApp ID of the message originator is stored into the remote_resource field, while the time of message receipt is stored into the timestamp field. Note that A stores also the record corresponding to the message that (s)he has sent to the group (record no. 1 in the figure). Records like this one can be easily identified by looking at the contents of their status and remote_resource fields, that store the values '4' and 'null', respectively.

Note also that the set of recipients, i.e. of the set of group members at the time of the sending, is not stored anywhere on the record. However, it can be indirectly determined by examining the records corresponding to the control messages that are automatically exchanged by the various group members each time a user joins or leaves the group. These messages, also stored in the messages table, always contain the value '6' in the status field, and encode in the media_size field the specific operation corresponding to the message (in particular, the values '1', '4', and '5' indicate group creation, join, and leave, respectively).

To illustrate, let us consider a scenario in which user 39320xxxxxxx (D, for brevity) creates a group on Nov. 11, 2013 at 16:24:05, and immediately adds user 39335xxxxxxx (E, for brevity) to the group. Then, D adds user 39333xxxxxxx (F, for brevity) on Nov. 12, 2013 at 10:40:48. The records generated by these operations in the chat database of user D are shown in Figure 16 (the same situation occurs on all the other group members).



	key_remote_jid	remote_resource	key_from_me	status	media_size	data	timestamp	received_timestamp
1	39320[redacted]1384187045@g.us	39320[redacted]@s.whatsapp.net	1	6	1	wa test group	1384187045000	1384187045496
2	39320[redacted]1384187045@g.us	39335[redacted]@s.whatsapp.net	1	6	4	(null)	1384187045970	1384187046015
3	39320[redacted]1384187045@g.us	39333[redacted]@s.whatsapp.net	1	6	4	(null)	1384252848773	1384252848834
4	39320[redacted]1384187045@g.us	39333[redacted]@s.whatsapp.net	1	6	5	(null)	1384467096987	1384467097066
5	39320[redacted]1384187045@g.us	39335[redacted]@s.whatsapp.net	1	6	5	(null)	1384508994642	1384508994761

Figure 16 Group management records stored in the msgstore database of user D. For other users we have the same situation, with the exception of record no. 1

Group creation corresponds to record no. 1, as can be seen from status='6' and media size='1'. The time of group creation can be ascertained (besides from the group_id) from the value stored in the timestamp field, while the field data stores the name given to the group (wa test group).

The addition of user E corresponds instead to record no. 2: the specific operation (join) and the identity of the user joining the group (E) can be deduced from fields media_size and remote_resource field, while the time of occurrence is stored in the timestamp field. A similar situation occurs with the addition of user F, whose control message corresponds to record no. 3.

Now, suppose that at a later time, namely on Nov. 14, 2013 at 22:11:36, user F leaves the group. This operation corresponds to record no. 4 in **Figure 16** (media_size='5' indicates a group leave), that reports the identity of the user leaving the group and the time of leave in the remote_resource and the timestamp field, respectively. Finally, when user E leaves the group on Nov. 15, 2013 at 09:49:54, record no. 5 is added to the messages table.

By using the information discussed above, the composition of the group over time can be reconstructed by chronologically sorting the various con- trol messages corresponding to join (status='6' and media size='4') and leave (status='6' and media size='5') of a given group (identified by the contents of the key_remote_jid field), as shown in **Figure 17**.

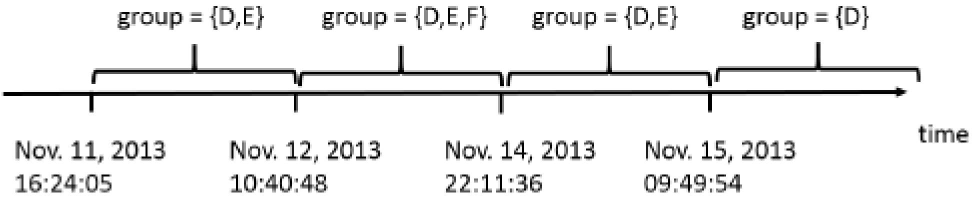


Figure 17: Timeline of group composition variations

From this information, it can be inferred whether a user belonged or not to a group when a specific message was sent to that group.

4.4.6 Dealing with deleted messages

In WhatsApp Messenger, the user may delete the records stored in the msgstore.db database in two different ways, namely:

- Deletion of an individual message: in this case, the corresponding record is deleted from the messages table
- Deletion of all the records belonging to a one-to-one, broadcast, or group chat conversation: in this case, all the records corresponding to the messages exchanged in that conversation are deleted from the messages table, as well as the record of the chat list table corresponding to that conversation

As discussed before, it is sometimes possible to recover deleted SQLite records, and in these cases the analysis techniques discussed in the previous sections can be applied. However, when such a recovery is not feasible, it may be still possible to determine many of the information regarding a deleted message by analyzing the log files generated by WhatsApp Messenger. In particular, as discussed below, it is possible to determine which messages have been deleted and when, when a deleted message has been sent or received and its state, as well as the users involved in the conversation. The same holds true for group control messages, so the analysis of log files also makes it possible to track the evolution of each group over time. In other words, only the contents of a deleted message cannot be recovered anymore.

When a message is deleted, WhatsApp Messenger records into the log file that indicates both the type of operation (msgstore/delete) and the identifier of the deleted message (1363253484-1), as well as the time of deletion. Each time a user-to-user, broadcast, or group chat message is sent/received, WhatsApp Messenger logs the time of the send/receive operation, the identifiers of the involved users, and the identifier of the message. Therefore, by searching into the log file the events corresponding to exchanges of deleted messages, it is possible to ascertain when those messages have been sent or received. Finally, WhatsApp Messenger logs also the events corresponding to reception of the acknowledgment messages sent back by the central server and by the recipient, from which it is possible to determine the state of a message, as well as the times of its state changes.

4.4 Analysis of settings and preferences

WhatsApp Messenger stores various information of potential evidentiary value in several files, located in the directories listed in Table 1 row no. 9. In particular, the file **me** stores (as ASCII text) the phone number registered with WhatsApp (i.e., the number used to create the corresponding WhatsApp ID). The relevance of this information derives from the fact that the SIM card currently used with the smartphone may not be the one used to register the user with WhatsApp. It is indeed possible to replace the latter SIM card with a new one, and to use the WhatsApp ID corresponding to the phone number of the old SIM card. Thus, a user A may impersonate a different user B, as long as A has used B's SIM card during registration, or he/she is using B's smartphone with a different SIM card. By comparing the phone number of the SIM inserted into a smartphone with the phone number stored in the **me** file, it is possible to determine whether this is the case or not.

Furthermore, the file me.jpg stores the currently-used avatar picture of the user. Given that the avatar pictures of all contacts are downloaded locally by WhatsApp Messenger (as discussed in Section 4.2.1), the me.jpg file can be used to understand that the user of the device under examination has been in contact with another user even if the latter one has deleted from its contacts database the record corresponding to the former one. As a matter of fact, the deletion of a record from the contacts database does not cause the deletion of his/her downloaded avatar picture.

5 Conclusion and Future Work

Mobile devices can be a source of digital evidence, containing personally identifiable information, including photographs, passwords, and other useful data, or indicating individuals' location at a specific time, interactions, and content communication, as well. The information they contain can also be an instrumentality of a crime.

In recent years, mobile device forensic analysis has also posed great challenges in terms of providing reliable proofs in court. With the growing demand for examination of mobile devices, a need for the solid development of process guidelines for the examination of these devices has been identified. While the specific details of the examination of each device may differ, the adoption of consistent examination processes will assist the imvestigator in ensuring that the evidence extracted from each device is well documented and that the results are repeatable and defensible in court. Research conducted and undergoing standardization attempts indicate that the mobile device forensics is under continuous development.

References

[1] Timoney, N., 2014. Consumer Contact: Job Advertising Fraud. WABI TV5, http://wabi.tv/2014/05/12/consumer-contact-job-advertising-fraud/

[2] Ntantogian C, Apostolopoulos D, Marinakis G, Xenakis C. "*Evaluating the privacy of Android mobile applications under forensic analysis*", Computers & Security, Elsevier Science, January 2014.

[3] Ben Martini, Quang Do, Kim-Kwang Raymond Choo, "*Conceptual Evidence Collection and Analysis Methodology for Android Devices"*, Information Assurance Research Group, University of South Australia, 2015.

[4] Brandon Letha, Arnold Garcia, "*Mobile Device Forensics: The New Frontier*", iDiscovery Solutions, The Metropolitan Corporate Counsel, February 2014, Page 20.

[5] Scott Polus, Mobile Device Forensics, Law Technology Today, http://www.lawtechnologytoday.org/2016/12/mobile-device-forensics, December 2016.

[6] ISACA, "*Overview of Digital Forensics"*, Cybersecurity Nexus, 2015.

[7] Zatyko Ken, "*Commentary: Defining Digital Forensics*", Forensic Magazine, 2 January 2007, www.forensicmag.com/articles/2007/01/commentary-defining-digital-forensics

[8] Rick Ayers, Sam Brothers, Wayne Jansen, *"Guidelines on Mobile Device Forensics"*, NIST Special Publication 800-101 Rev. 1, May 2014,https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-101r1.pdf

[9] Matthew David, Kimberly Clarke, "*Learn more about the Android Studio IDE from Google*" Tech Target, http://searchsoftwarequality.techtarget.com/feature/Learn-more-about-the-Android-Studio-IDE-from-Google

[10] Amjad Zareen, Dr Shamim Baig, *"Mobile Phone Forensics Challenges, Analysis and Tools Classification"*, 2010 Fifth International Workshop on Systematic Approaches to Digital Forensic Engineering.

[11] Det. Cynthia A. Murphy, *Developing Process for Mobile Device Forensics*.

[12] Cosimo Anglano, "Forensic Analysis of WhatsApp Messenger on Android Smartphones", DiSIT - Computer Science Institute, Universita del Piemonte Orientale, Alessandria, Italy, September 2014.

[13] Aditya Mahajan, M.S. Dahiya, H.P. Sanghvi, "Forensic Analysis of Instant Messenger Applications on Android Devices", International Journal of Computer Applications, April 2013.

[14] Neha S. Thakur. "Forensic Analysis of WhatsApp on Android Smartphones", Master's thesis, University of New Orleans, 2013. Paper 1706.

[15] D. Cortjens, A. Spruyt, W.F.C. Wieringa, "WhatsApp Database Encryption Project Report", Technical report, 2011, https://www.os3.nl/media/2011-2012/students/ssn_project report.pdf.

[16] S. Jeon, J. Bang, K. Byun, and S. Lee. "A recovery method of deleted records for SQLite database", Personal and Ubiquotous Computing, 16, 2012.

[17] Steven M. Bellovin, Matt Blaze, Sandy Clark, Susan Landau, "Lawful Hacking: Using Existing Vulnerabilities for Wiretapping on the Internet", In Proc. of Privacy Legal Scholars Conference, June 2013, http://dx.doi.org/10.2139/ssrn.2312107.

[18] Petr Vanek, Kamil Les, "Sqlite Databases Made Easy", 2013, http://sqliteman.com.

[19] Rolfe Winkler, "WhatsApp Hits 400 Million Users, Wants to Stay Independent", The Wall Street Journal - Digits, Oct. 2013,http://blogs.wsj.com/digits/2013/12/19/whatsapp-hits-400-million-users-wants-to-stay-independent.

[20] WhatsApp Inc. WhatsApp, 2013,http://www.whatsapp.com.

[21] Oxygen Forensics, Inc. SQLite Viewer, 2013, http://www.oxygen-forensic.com/en/features/analyst/data-viewers/sqlite-viewer.

[22] Cosimo Anglano, Massimo Canonico, Marco Guazzone, "Forensic analysis of the ChatSecure instant messaging application on android smartphones", DiSIT - Computer Science Institute, University of Piemonte Orientale, Alessandria, Italy, Elsevier , October 2016

[23] Songyang Wu, Yong Zhang, Xupeng Wang, Xiong Xiong, Lin Du, "Forensic analysis of WeChat on Android smartphones", The Third Research Institute of Ministry of Public Security, Shanghai 201204, China, Elsevier, November 2016

[24] Timothy Vidas, Chengye Zhang, Nicolas Christin, "Towards a General Collection Methodology for Android Devices", Digital Investigation, Aug 2011.

[25] Micro Systemation, XRY, 2013, http://www.msab.com/xry/xry-current-version.

[26] Cellebrite LTD., "Cellebrite Android Forensics", 2013, http://www.cellebrite.com/mobile-forensics/capabilities/android-forensics.

[27] Willassen, S., "Forensic analysis of mobile phone internal memory", In: Pollitt, M., Shenoi, S. (eds.) Advances in Digital Forensics. IFIP, vol. 194, pp. 191–204, Springer, Boston (2006)

[28] AccessData Corporation, FTK Imager, 2013, http://www.accessdata.com/support/product-downloads.

[29] Thing, V.L.L., Chua, T.-W., "Symbian smartphone forensics: Linear bitwise data acquisition and fragmentation analysis", International Conference on Security Technology, November 2012