applied sciences

Article

Methodology for Forensics Data Reconstruction on Mobile Devices with Android Operating System Applying In-System Programming and Combination Firmware

Claudinei Morin da Silveira [1,†], Rafael T. de Sousa Jr. [1,†], Robson de Oliveira Albuquerque [1,2,†], Georges D. Amvame Nze [1,†], Gildásio Antonio de Oliveira Júnior [1,†] and Ana Lucila Sandoval Orozco [1,2,†] and Luis Javier García Villalba [2,*,†]

[1] Cyber Security INCT Unit 6, Laboratory for Decision-Making Technologies (LATITUDE), Department of Electrical Engineering (ENE), Technology College, University of Brasilia (UnB), Brasilia-DF 70910-900, Brazil; claudineimorin@gmail.com (C.M.d.S.); desousa@unb.br (R.T.d.S.J.); robson@redes.unb.br (R.d.O.A.); georges@unb.br (G.D.A.N.); jrgildasio@gmail.com (G.A.d.O.J.); asandoval@fdi.ucm.es (A.L.S.O.)

[2] Group of Analysis, Security and Systems (GASS), Department of Software Engineering and Artificial Intelligence (DISIA), Faculty of Computer Science and Engineering, Office 431, Universidad Complutense de Madrid (UCM), Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, Spain

* Correspondence: javiergv@fdi.ucm.es

† These authors contributed equally to this work.

Received: 16 May 2020; Accepted: 12 June 2020; Published: 20 June 2020

Abstract: This paper proposes a new forensic analysis methodology that combines processes, techniques, and tools for physical and logical data acquisition from mobile devices. The proposed methodology allows an overview of the use of the In-System Programming (ISP) technique with the usage of Combination Firmware, aligned with specific collection and analysis processes. The carried out experiments show that the proposed methodology is convenient and practical and provides new possibilities for data acquisition on devices that run the Android Operating System with advanced protection mechanisms. The methodology is also feasible in devices compatible with the usage of Joint Test Action Group (JTAG) techniques and which use Embedded Multimedia Card (eMMC) or Embedded Multi-Chip Package (eMCP) as main memory. The techniques included in the methodology are effective on encrypted devices, in which the JTAG and Chip-Off techniques prove to be ineffective, especially on those that have an unauthorized access protection mechanism enabled, such as lock screen password, blocked bootloader, and Factory Reset Protection (FRP) active. Studies also demonstrate that data preservation and integrity are maintained, which is critical to a digital forensic process.

Keywords: Mobile Device Forensics; In-System Programming; Combination Firmware; physical/logical data acquisition; android data extraction

1. Introduction

In recent years, our society has experienced an accelerated process of computerization, thus becoming highly interconnected. In May 2019, in Brazil alone, public records pointed to over 230 million smartphones in use. If we add up laptops and tablets to that number, there were then 324 million portable devices, i.e., approximately 1.6 portable devices per inhabitant [1].

With the evolution of mobile devices, a large part of the population has been replacing the computer by the smartphone to perform most tasks involving technology, for example, exchanging messages, sending e-mails, making online payments, etc. According to the StatCounter (Web traffic analysis website) [2], smartphones account for 51.69% of computing devices (smartphones, tablets, and computers) in use worldwide.

Mobile devices are used for a variety of tasks, such as sending or receiving text or voice messages, images, chatting by voice or video calls, watching videos, sending or receiving e-mail, making bank transactions, capturing and manipulating images, using applications for health care, relationships, social networks, monitoring of physical activities, mobility and traffic and even to store and display digital documents. In Brazil, we can mention as an example some government services applications of interest to the citizen, such as the Voter Card, the Vehicle Registration and Licensing Certificate, and the National Driver's License.

With technological advances, it has become common for mobile devices to receive constant hardware and software improvements. With each new device launched, a new System on Chip (SoC) is employed, which is more modern, faster, and more energy-efficient, in addition to expanding storage capacity. New mechanisms to protect against unauthorized access are also frequently implemented or improved.

Numerous manufacturers already produced cell phones and marketed them globally before the appearance of the Android operating system (OS). After that, they started to invest in hardware development so that their devices started to run the new OS. New manufacturers have also emerged, focusing on the global market or the Asian market, whose consumer public has quite significant numbers. South Korean Samsung is the manufacturer that holds the largest share of the global mobile device market [3] and runs the Android OS, making it the most widely used OS worldwide [4]. Even before the Android OS became the most used OS, this system has been considered an important source of information. As remarked by Simão et al. [5], Android OS constitute a large repository of information in a forensic perspective, both on-site as well as provided remotely.

Modern mobile devices are a rich source of data that can be forensic evidence in investigation processes and security analysis. Such devices increasingly have more storage space, connectivity options, and multitask ability, making them the first option for countless users. Currently, devices that run the Android OS and its variations are more accessible in the global market. That turns them into natural candidates in forensic analysis and investigation processes.

When it comes to security and digital forensics, analysts strive to acquire and analyze data from mobile devices. The diversity of technology used on mobile devices, the accumulation of digital evidence, the lack of standardized extraction methodologies, and the lack of necessary training characterize challenges that hinder or prevent data acquisition [6].

Companies that develop technologies for digital forensics are also affected by the diversity of components and manufacturers of mobile devices, as they cannot be effective and efficient in all models or in all manufacturers. The gaps left by such companies are also a barrier to law enforcement forces.

Files stored in digital format are considered digital evidence, including audio, video, and image files, and even the software or the hardware itself. It is noteworthy that such files and devices can be part of the investigation of most crimes in the digital environment. Therefore, they must be adequately investigated and protected against alteration, maintaining protection under the chain of custody.

The Mobile Device Forensics is characterized by difficulties in accessing device data, the effort to unlock or bypass security mechanisms and obtain data, and often, by the inability to obtain the entire volume of data contained in the device. One is also feature by the fact that it is more complex than traditional computer forensics, requiring better-trained teams and specialized equipment to obtain legally acceptable results.

Unlike traditional Forensic Computing, where any intervention in the system must be avoided at all costs, in forensics on mobile devices it is necessary to perform access and make interventions directly on the hardware for possible attempts to bypass the access block mechanisms, and also install elevation of privilege applications. However, the correct use of techniques allows to preserve the evidence, and consequently, the legitimacy of the obtained evidence.

Figure 1 represents, in a summarized way, the steps of a methodology that can be applied to mobile forensics.

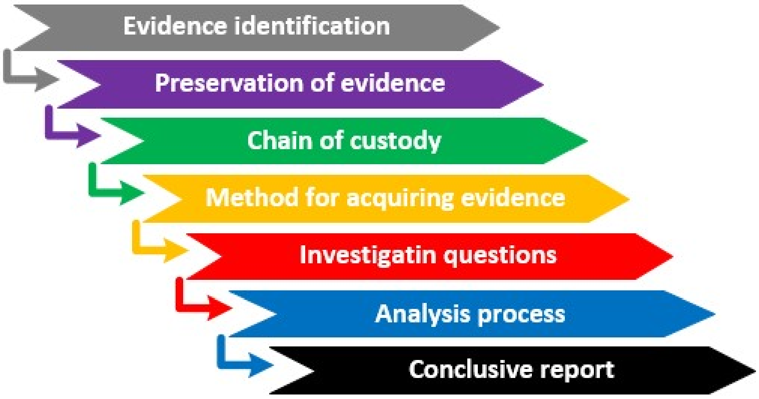


Figure 1. Steps of forensic methodology.

With regard to mobile device forensics, the scenario for obtaining evidence may also be different. We will briefly address each of the steps of the methodology, always aiming at obtaining the legality of the actions.

1. *Evidence identification*: As it is a mobile device, its identification may be more difficult, as the offender tries to hand over another device in place of the one who actually has the evidence. Therefore, it is essential to determine which fact needs to be clarified and which devices should be analyzed. It is important to record all details of the location and the seized items.
2. *Preservation of evidence*: The device to be analyzed must always be handled with gloves so that the fingerprints of the device user are preserved. Although it is not part of the acquisition and analysis of digital evidence, such evidence may be relevant to other forensic sciences. The device should be kept, whenever possible, in the state in which it was seized. If it is seized turned-off, it must remain off. Otherwise, it is also important to put the device in airplane mode, to prevent it from receiving new calls, SMS, creating new itinerary records by GPS, avoiding false positives during the analysis process, or even, that the data may be erased remotely. If this is not possible, a faraday bag must be used to transport the device to the laboratory where the data will be acquired. All features of the device must be documented.
3. *Chain of custody*: Chain of custody, in the legal context, refers to the chronological or historical register that records the sequence of custody, control, transfer, analysis, and disposition of evidence, whether physical or electronic, and is of fundamental importance in criminal cases.
4. *Method for acquiring evidence*: The forensic analyst should evaluate the best tool or methodology for data acquisition, preferably using the least invasive methods, and it may be used these according to the need.
5. *Investigating questions*: The authority responsible for elucidating the case must formulate the questions for which the forensic analyst must search the evidence for answers that satisfy the authority's questions.
6. *Analysis process*: These are the processes, tools or methodologies through which the acquired evidence will be analyzed to answer the investigative questions.
7. *Conclusive report*: It is the document that includes all the records made in the previous steps, and mainly, answers the investigative questions asked by the competent authority.

It has become common to find public reports showing that the use of smartphones in the practice of illegal acts is growing. Increasingly, these devices are used to practice criminal activities, becoming digital evidence, relevant for criminal investigations. Due to a more significant market share, the acquisition and forensic analysis of data from Android devices have gained significant importance in the field of digital forensic investigation [7].

One of the most recent examples of crucial evidence found on mobile devices can be seen in the case of the sniper Mohammed Saeed Al Shamrani, who on 6 December 2019, killed three American sailors at a military base in Florida. Authorities discovered contacts between Mohammed Saeed Alshamrani and Al Qaeda agents after being able to access the contents of the sniper's mobile devices [8].

In this work, a methodology was developed and named by the authors as Low-Level Data Acquisition ISPCF (LLDA-ISPCF). This methodology combines the use of In-System Programming (ISP) and Combination Firmware to bypass the lock screen, followed by specific procedures and through the use of proper tools for forensic analysis in both hardware and software.

As a way to validate the methodology, the device data was acquired by using the Universal Forensic Extraction Device (UFED) Touch 2, preserving the integrity of the data, and, finally, the acquired data were analyzed in the UFED Physical Analyzer. It is worth mentioning that the methodology developed is not limited to the tools used. That means that other tools, in the market, can perform tasks in the forensic analysis process for mobile devices forensics.

1.1. Main Contributions and Limitations of the Proposed Methodology

The main contributions of the proposed solution are the following:

- Enable the lock screen bypass, regardless of the OS version;
- Prove the effectiveness of the LLDA-ISPCF methodology on devices with File-Based Encryption;
- Enable data acquisition from a wide range of mobile devices, regardless the manufacturer;
- Enable device data acquisition using specialized, proprietary and open-source tools while preserving data integrity;
- Allow the root user to be enabled without the risk of losing user data, which allows physical extraction on models whose acquisition process is only possible with the rooted device;
- Allow the acquisition and analysis of data by any capable forensic tool, observing the compatibility of the analysis tool with the format generated by the acquisition tool.

Regarding the limitation of the methodology, it is considered that—to the best of the authors' knowledge and until the elaboration of this work—devices that use Universal Flash Storage (UFS) main memory or that have hardware-supported encryption are not compatible with the tools employed in the proposed methodology. That future updates to the firmware by the manufacturers may make the methodology unfeasible. Also considered is the fact that Special Firmware, such as the Combination Firmware used on Samsung devices may not be available for a specific model or manufacturer, and its use combined with ISP requires a specific study of the device from which the data will be acquired to bypass the lock screen mechanism.

1.2. Motivation

The main motivation of this work is to develop a novel technique that is able to help law enforcement agencies (LEA) to conduct forensics in mobile devices that have security protections enabled, since these protections prevent LEA to properly collect evidence and analyze them in case of crimes.

It is important to remember that criminals all over the world use communication devices such as mobile phones with security measures applied to them. Such mechanisms make it very hard to forensics specialists to bypass them and acquire evidence that helps in the solution of criminal cases.

1.3. Article Structure and Organization

The remainder of this paper is as follows. Section 2 presents some basic concepts for understanding the purpose of this work, as the state of the art reviews, and some related works. In Section 3 we present the problem description and the proposed solution, and describes the new methodology created and its phases. In Section 4 we present a proof of concept and results of the proposed methodology. We conclude this work in Section 5 with some considerations about future work.

2. Related Works and State of the Art Review

This section presents notions about standard acquisition techniques, forensic life cycle, some tools and software for smartphone repair, a brief description of In-System Programming (ISP), Combination Firmware, data encryption applied to the forensic context, bootloader in Android OS smartphones. In addition, this section reviews the main related works and the limitations encountered.

2.1. Acquisition Processes

The acquisition process is divided into physical, logical, and file system acquisition. These processes are summarized below.

1. *Physical Acquisition*: Physical acquisition on mobile devices consists of copying information from the device by direct access to the internal storage memory. The process creates a copy of the entire file system bit by bit. Such an approach is similar to that adopted in computer forensic investigations. A physical acquisition can acquire all data present on a storage device, including deleted data that has not yet been overwritten, in addition to copying unallocated space [9]. It is considered to be the most effective in forensic terms and is performed using specific tools. According to Mota Filho [10], the least amount of information that an OS can read in a filesystem is a block that, at the physical disk level, is equivalent to a cluster. Copies and readings are made block by block at the filesystem level. The File Systems widely used by Android devices were Flash-Friendly File System (F2FS) [11], and currently, Fourth Extended Filesystem (EXT4) [12]. Both file systems adopt 4KB-sized logical blocks. Still, according to Mota Filho [10], there are software such as *dd* and *dc3dd* that can read a physical sector, which has 512 bytes by default and is the smallest unit of information that can be read on an HD or pen drive by its controller.
2. *File System Acquisition*: The file system acquisition is technically seen as a type of logical acquisition [9]. However, it is more abundant in data, as the entire file system of the device is copied. It contains files and directories that the device uses to populate applications, system settings, and user settings, along with user storage areas. It also includes files not directly accessible to the user through the device interface, which requires specialized tools to access such artifacts. However, unlike physical acquisition, this type of acquisition does not copy the unallocated space of physical memory.
3. *Logical Acquisition*: Logical acquisition is a copy of logical storage objects, such as file systems, directories, and files. Data is copied from the allocated space on disk, still accessible to the user in the file system. Such data includes the phone book, calls, messages, some application data, and other data that can be expected from a software backup with iTunes or Android itself, that is, what we can see if we manually examine the device [13]. It is observed as the fastest and least invasive, but it is the most limited of acquisitions.

Figure 2 compares, in a summarized way, the different data that can be acquired by the types of acquisitions presented.

2.2. Digital Forensics Life Cycle

Ajijola et al. [14] presented a comparative assessment between forensic guidelines from NIST SP 800-101 Rev.1: (Guidelines on Mobile Device Forensics) [15] and International Organization for Standardization (ISO)/IEC 27037: 2012 (Guidelines for identification, collection, acquisition, and preservation of digital evidence) [16]. The study resulted in the proposal for an integrated implementation of the two forensic guidelines.

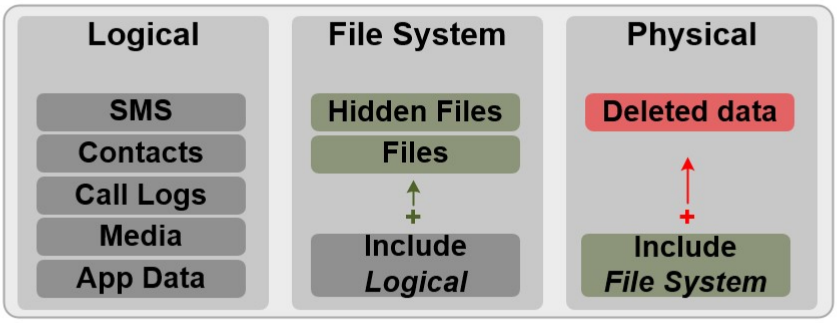


Figure 2. Summary of data types for each type of acquisition.

Considering the standards mentioned, none of them deals with all processes of digital forensic investigations. There are forensic guidelines common to the two standards, and others that only one of the standards contemplates, exposing their limitations that affect more current issues of forensic processes. According to Neumer e Weippl [17], another guideline that can be applied together is the RFC 3227 (Guidelines for Evidence Collection and Archiving) [18], which also provides guidelines for proper digital forensics practices. Figure 3 shows the typical life cycle or the main stages of digital forensics.

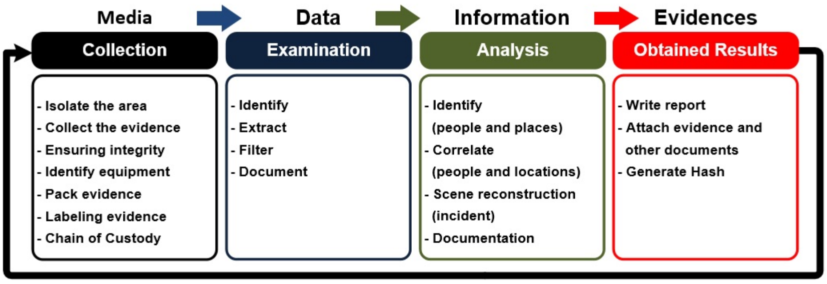


Figure 3. Digital forensics life cycle. (Source: Adapted from [19]).

2.3. *Tools and Software for Smartphone Repairing*

According to Yang et al. [20], companies that provide technical support or perform repairs on mobile devices use tools known as *Box*. They are interfaces endowed with functionalities that allow reading and writing in areas of the mobile devices internal storage that are not accessible to the user and can also be used as forensic tools.

Among the various models used, we can mention Octoplus Pro [21], UFI [22], Easy JTAG Pro [23], and their respective software. Although these boxes have resources to execute ISP, JTAG, and Chip-Off, they are rarely used for this purpose. They are widely used for mobile devices software repairs, considering that the companies focus that provide technical assistance is to return the user a functioning device without worrying about the data contained in it.

There are other tools and software used for repairs, both physical and logical, on mobile devices. For software repairs, we can mention the interfaces known as Dongles, which consist of a USB interface that serves to read a smart card and validate the software license. For physical repairs, the same tools are used to prepare the device for the execution of ISP, JTAG, and Chip-Off. The Chip-Off forensic technique uses a Soldering Station, Surface-Mount Device (SMD) Rework, antistatic forceps, antistatic mat, tin for soldering, soldering flux, stereoscopic microscope, electron microscope, PCB (Printed Circuit Board) holder, probes and spatulas, Multi-Purpose Rotary Tool, high precision digital multimeter, LCD Disassembly Machine, among others.

2.3.1. In-System Programming

In-System Programming (ISP) is a technique already used for forensic analysis of mobile devices. However, in this article, we will discuss its use not only for reading data from the device's main memory, but also for writing data, changing partitions of the device's main memory, it is considered that—to the best of the authors' knowledge and until the elaboration of this work—it is a possibility not yet explored. Its use will not be as a main tool, but as a component of a methodology that combines the use of other tools for the successful acquisition of data.

According to Afonin and Katalov [24], The forensic analysis using the ISP technique is a less invasive or destructive variation than the Chip-Off technique. The ISP technique involves an advanced acquisition process between the Joint Test Action Group (JTAG) and the Chip-Off.

According to Pappas [9], ISP is an acquisition technique similar to JTAG, but faster because its connection is made directly to flash memory, bypassing the processor, through Test Access Points (TAPs), and can be used when the device is not compatible with JTAG. ISP acquisition applies to an eMMC or eMCP flash memory, not limited to smartphones only, and can be used on any device that uses these types of memories, such as SD cards.

In ISP, it is necessary to know the TAPs that connect to the flash memory. It is necessary to use the Box to connect the computer, where the software will run, with the device board. There are several Box models on the market with the ability to run ISP, the most well known being Easy JTAG Plus and Z3X Pro.

For the ISP technique, during the acquisition process, the contents of the internal memory are copied without removing the chip. To the best of the authors' knowledge, to date, ISP acquisition is only available for devices that use flash memory as Embedded Multimedia Card (eMMC) or Embedded Multi-Chip Package (eMCP) with Ball Grid Array (BGA) encapsulation type.

In more detail, running the ISP is a process that requires several steps. First, connect the flash memory access taps to an adapter. Second, weld one end of a metallic conductor to the TAP and the other end to the adapter. Third, the TAPs are connected to the Box. The software of the Box provides the TAPs diagram for most smartphones. There are also specialized sites that make these images available. Figure 4 shows the solders on the adapter and the TAPs on the board.

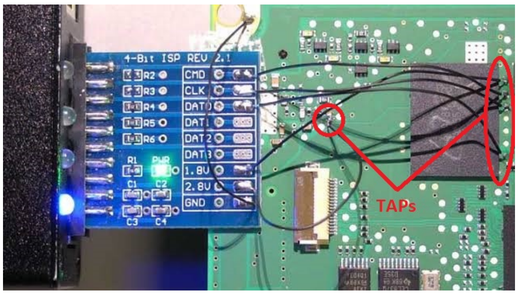


Figure 4. Board ready to run ISP.

The use of the ISP technique allows another analyst to reproduce or repeat all steps of the process for data acquisition since the device remains functional and with all the components on the board. However, even when the Chip-Off technique is applied is possible to return the component to its original location and made to device run again, doing the component soldering process, known as reballing.

It is also possible to make use of an interface called VR-TABLE, which has articulated arms, equipped with high-precision metal probes, which dispense the welding of the conductors on the mobile device board, eliminating the possibility of damage caused by overheating. Figure 5 demonstrates the use of a VR-Table.

The software of the Box is capable of performing reading/writing operations directly on the eMMC. After soldering all the necessary conductors, the Box is connected to the computer using a USB cable. After completing this step, it is necessary to select the correct software to be used and to configure the necessary parameters. When everything is connected and the software with the parameters correctly configured, the Box can then access or manipulate the content of the memory.

Running the ISP, it is possible to manipulate the data using 1 or 4-bit bandwidths. The most common is the use of the value "1 bit" because it is less susceptible to errors during copying and because it reduces the exposure of the board to heat since for execution with "4-bit" bandwidth, it is necessary to weld 3 more conductors.

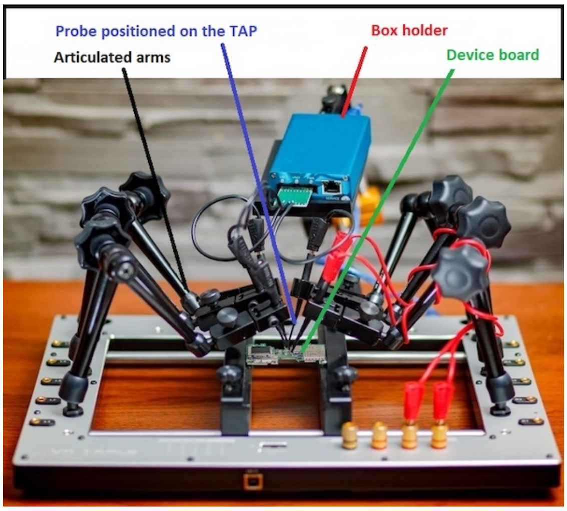


Figure 5. VR-Table. (Source: Adapted from [25]).

In Table 1, TAPs that must be connected for the execution of the ISP are described.

Table 1. Functions of TAPs at the ISP.

ISP Pinout	
TAP	Function
CMD	Command in/Response out
DAT0	Data input/output
CLK	Clock
VCC	Supply voltage for Core (2.8/3.3 V)
VCCQ	Supply voltage for I/O (1.8/3.3 V)
GND	Ground

In Figure 6, the TAPs on the card of a Samsung mobile device model GT-N8000 are identified. In this example, DAT1, DAT2, and DAT3, which make it possible to run an ISP with a 4-bit bandwidth, are also identified. The transfer rate can vary from 1.2 to 1.8 Kbps, according to the selected frequency. If the bandwidth is 4 bit, the transfer rate is multiplied by 4.

It is also possible to create a binary image of the device memory using the software of the Box. This image can be analyzed by forensic tools such as UFED [26], XRY [27] or AXIOM [28].

The process of preparing the mobile device for the execution of the ISP also enables physical acquisition in devices equipped with SoCs, produced by Qualcomm. Using its Emergency Download Mode (EDL), a method proposed by Wu et al. [7], it is necessary to weld a conductor in the TAP **CMD** of the device for a physical acquisition using EDL. On devices from some manufacturers, such as XIAOMI, there is also the possibility to log into EDL mode logically by typing a code on the dial pad. For XIAOMI devices the code is ***#*#717717#*#***.

2.3.2. Combination Firmware

An Engineering Firmware or Special Service Firmware is a special type of code developed to read the device's hardware, displaying complete information, in addition to allowing hardware testing and the execution of basic device applications and functionality. It is possible to obtain the Special Firmware for devices from various manufacturers, however, they are found with different nomenclatures. For Xiaomi devices, they are known as ENG Firmware or Engineering Rom and for Samsung devices, although they have the same functionality, this firmware is known Combination Firmware. It is important to clearness that the most common use of these Firmwares is the maintenance of mobile devices.

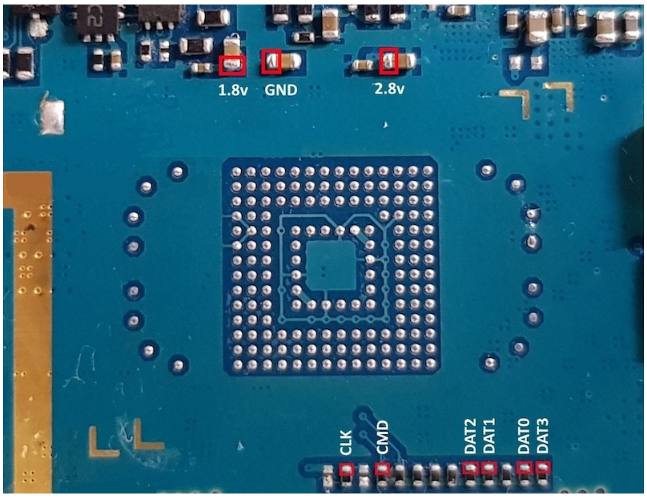


Figure 6. Samsung GT-N8000 Pinout.

According to Morgillo and Viola [29], Read-Only Memory (ROM) is a type of memory generally used in embedded systems to safely store all files that are part of the system core. The firmware that runs the Android operating system on mobile devices is stored in an area of memory whose permission is "Read Only"; that is why this firmware is popularly known as ROM. There is an effort by manufacturers to ensure the highest integrity of the system as possible, assuring that the primary system is fault-tolerant and remains intact during the device re-initialization. However, using specific tools, it is possible to manipulate this internal memory area and install its custom firmware.

It is considered that such customization is possible because Android is one of the most popular open-source projects and because, in general, the *custom ROMs*, which correct specific errors for specific scenarios, are optimized versions of the original system. In the latter case, they overwrite all areas of the internal memory when replacing the original firmware.

According to Parry and Carter [30], the mobile devices firmware update can be performed over a wireless connection known as Over-the-air (OTA). Open Mobile Alliance (OMA) has developed firmware update techniques that may require tens, hundreds, or even thousands of files, and some solutions bundle all the necessary update files into a single file to be downloaded directly to the mobile device. The replacement of the original firmware with another, developed by third parties, requires a specific tool and is done through a computer.

For installing a Combination Firmware, it is necessary to use specific software that rewrites the firmware. Once the original firmware is replaced with the Combination Firmware, it is possible to check all the hardware resources such as rear camera, front camera, sensors, RAM, ROM, and also run a test that shows the diagnosis results.

Engineers and technicians widely use Special Firmwares and, although there is no official documentation on them, there are many topics in specialized forums, such as *XDA Developers* [31], about their use. The files can be downloaded from websites specialized in tools for mobile devices maintenance.

From a forensic point of view, other interesting features can be explored through a Combination Firmware. It allows bypassing and removing factory reset protection or Google account verification. However, the main point of interest is to allow full access to the device content since replacing the original firmware with Combination Firmware does not change the data partition. This preserves data integrity and allows physical acquisitions, file system acquisitions, and logical acquisitions to be made using forensic tools. It is also possible to enable the root user if necessary. According to Almehmadi and Batarfi [32]—who investigated the impact on user data integrity by enabling the root user on an Android smartphone—, modifications to the device data do not affect the user data Such a process is, therefore, legally valid and evidence extracted from Android devices, as a result of the rooting process, is robust and reliable for sentencing in criminal cases.

It is noteworthy that a Combination Firmware is not intended for the end-user, but specialized tasks according to the need. Using such techniques without adequate knowledge can invalidate the use of a device in a criminal investigation process, eliminating substantial evidence for crime investigation. It is also necessary to note that the Combination Firmware always runs the same Android version as the original firmware.

When there is no Special Firmaware for the target device, it is possible to download the image file of the *boot*, *recovery*, and *system* partitions, also developed for repair mobile devices, and making adaptations so that they are similar to those found in the Special Firmware. However, this requires individual analysis for each device. This procedure will not be described in this article.

2.4. Using Encryption on Android

According to Loftus and Baumann [33], Google has made available since version 3.0 (Honeycomb) of Android, the Full-Disk encryption (FDE) encryption model, which allows encrypting the */data* partition, also called "user data". After turning on the device, all data in the */data* partition are inaccessible until the user provides authentication credentials. Even using such advances, this has not prevented attacks such as *Evil Maid Attack* and *Cold Boot* from succeeding against FDE. From a security and forensic perspective, there are still other drawbacks arising from the use of the FDE. For example, after rebooting, several critical functions of the device cannot be used without user interaction; as an example, we can mention the inability to receive calls after an unexpected restart.

Yet, according to Loftus and Baumann [33], File-Based Encryption (FBE) was introduced to overcome this problem as part of the launch of Android 7.0 (Nougat). FBE protects user data by encrypting each file with a unique key, which consists of a hardware key with the user's input (alphanumeric password, PIN, etc.) associated with the Secure Boot protection (Direct Boot). Nevertheless, it also has the convenience of not suffering limitations, correcting the mentioned flaw of the FDE, besides allowing a more refined control of what is encrypted.

Direct Boot can be set during the initial setup and while specifying the lock screen password. When the secure boot option is enabled, and the user informs their password, the data partition is mounted and decrypted during initialization, and some applications can start and access their data, even if the screen is not unlocked. The credential request occurs before most Android services and applications are allowed to start. The credential is required to generate the actual encryption key.

Some cases of apps that run in secure boot mode include apps with scheduled notifications (e.g., alarm clock, SMS) and that provide accessibility services, such as Talkback. According to Alendal [34], this security mechanism can be circumvented.

On devices manufactured by Samsung, FBE encryption is present only on Samsung Galaxy devices with Android 9.0 or higher, or with Knox 3.3 or higher. In the others, the encryption is still FDE [35].

According to Ribunov et al. [36], the coexistence of critical and non-critical applications is becoming usual on mobile devices. Critical applications must be run separately in the Trusted Execution Environment (TEE) so that the code and associated data can be protected against malicious applications.

So as to Leignac et al. [37], the Trusted Execution Environment (TEE) was created to improve security within the SoC based on the ARM architecture. It offers a compromise between the functionality of the Rich Operating System (Rich OS), such as Android, and the safety of a Secure Element (SE). ARM TrustZone separates SoC between two environments, one considered secure and the other not-as-secure.

The keys are derived from the information stored in the TEE, as well as the user credentials (PIN, password, etc.) used to unlock the phone. If file-based encryption is used, the phone can initialize and access data stored in the device's specific encrypted area, protected with hardware keys. Most of the information, however, is stored in the encrypted area with various credentials. This area is protected with keys based on the user's credentials.

Thus, an acquisition made through the software of a Box, either by ISP or by Chip-Off, will not bring satisfactory results, as the device is turned off and, consequently, encrypted.

2.5. Bootloader

To Hay [38], there is a chain of boot loaders that originate from the Original Equipment Manufacturer (OEM) or the SoC manufacturer. The primary boot loader (PBL), written by the chipset manufacturer, triggers the Bootloader (ABOOT). There is still the TrustZone, which provides security mechanisms such as Secure Boot.

Android devices come with the bootloader blocked to ensure the integrity of the OS. To unlock the bootloader—which allows the installation of an alternative bootloader and custom firmware—some devices require an authorization code from the manufacturer. On some devices, the factory reset occurs. That is, the user data will be lost, which is not interesting from a forensic point of view.

2.6. Forensic Suites Specialized in Mobile Devices

There are numerous forensic tools and suites on the market aimed at acquiring and analyzing data from mobile devices. Studies of Rao and Chakravarthy [13], and Khan and Mansuri [39] point out some of them and those used by the National Institute of Standards and Technology (NIST) [40]: SAFT, AFLogical, LiME Module, Nandroid Backups, Open Source Android Forensics–Tool Kit (OSAF-TK), Santoku Linux, Andriller, JTAG, Chip-Off, Cellebrite UFED, Mobile Oxygen Forensics, Paraben, Mobile Device Seizure, MOBILedit Forensic, MPE+, XRY, AXIOM, EnCase Forensic, X-Ways, XRY, UFED, Paraben, and Final Mobile Forensics.

We can also quote the IPED Digital Forensic Tool, which can be obtained free of charge on the website of the Federal Police Department of Brazil [41].

According to Pappas [9], any tool or software used for data acquisition and analysis must be tested and verified before its use in real cases to guarantee its performance and that the documents related to the software/hardware are periodically reviewed. The NIST provides validation reports for some tools on its website [42]. The reports point out the features and limitations of these tools.

2.7. Related Works

This section presents some related works, in which efforts were undertaken towards data acquisition from mobile devices using methodologies and techniques different from those presented because they already exist in the market. Such works were grouped according to the techniques and methodologies used.

2.7.1. Acquisitions Based on Device Firmware Attack

Yang et al. [20] proposed an acquisition method based on the analysis of firmware update protocols for Android smartphones from manufacturers Samsung, LG, and Pantech. They stated that it is possible to perform physical acquisition on Android smartphones using the flash memory read command, applying reverse engineering to the firmware update protocol on the boot loader. However, at the time of the work, the Android devices used were the models G3 (F400S, D851), Optimus G (F180S, E975), R3 (IM-A850S), Iron2 (IM-A910S), and Nexus 4/5 (E960, D821), which have not yet the main memory encrypted by default.

Li et al. [43] presented a forensic tool based on the replacement of the Recovery Mode with a focus on NAND and eMMC cards. The possibility of data recovery was explored after uninstalling an application on Android devices with EXT4 file system. Of the devices submitted to data extraction, the most recent version of Android was 4.4.X, with kernel 3.4.39.

Wu et al. [7] proposed a forensic acquisition for Android smartphones with Qualcomm processors in an approach using special modes. They addressed the special modes 9008 and 9006 of the Qualcomm processor to extract the data partition. This mode is also known as Emergency Download Mode (EDL). Acquiring the data image using Qualcomm 9008 Mode, there is no need to unlock the boot loader; start the device in fastboot mode and set the phone to Qualcomm 9008 mode. Mode 9006 is applied to cause intentional damage to the boot partition, after which a computer can read the device's data partition. In the experiments, data integrity and the possibility of executing the proposed methods were also proven. However, the experiment was done on devices with Android OS Version 5.1.1, and the proposed approach reaches only devices using Qualcomm SoCs.

Alendal et al. [34] exploited the Common Criteria (CC), which is a feature that increases the security level of Samsung devices and therefore makes it difficult for forensic acquisition for law enforcement. Due to the impossibility of gaining access to the project's specifications, documentation, and source code, the authors reverse-engineered the implementation of CC mode, as Samsung's secure boot manager protects it. They presented how this security mode is applied, its vulnerabilities, and how they can be used to bypass CC mode by increasing the attack surface for later forensic acquisition. However, the work is limited to devices manufactured by Samsung.

Dave et al. [44] recovered 3500 AT commands from more than 2000 Android smartphone firmware images from eleven suppliers. The commands were tested on eight Android devices from four different suppliers via the USB interface. The authors identified the possibility to rewrite the device's firmware, bypass Android's security mechanisms, obtain sensitive device information, perform screen unlocks, and inject touch events using AT commands only. There are still other features that can be exploited by the AT command, which features an attack surface on Android devices. AT commands were written in the early 1980s, whose purpose was to control modems, but they can still be used on most Android OS smartphones. The technique proved to be effective only on Samsung devices up to the Galaxy S7 Edge model with Android OS up to version 7.0. Tested on a Galaxy S8 Plus model, the technique proved ineffective, as well as on Google Nexus 6P and Google Pixel models.

2.7.2. Acquisitions Using ISP, JTAG, and Chip-Off

Pappas [9] conducted a study focused on JTAG and ISP physical acquisition techniques to prove that these forensic techniques are equivalent when compared to any other method. To this end, he proposed three tests. The first test showed the differences in the results of the forensic analysis when performing a physical acquisition, file system, and logical acquisition, using UFED Touch, and later comparing the results using the Physical Analyzer. The results revealed that the physical extraction is more potent and, in forensic terms, retrieves a larger quantity of evidence.

The second test [9] aimed to prove that all physical acquisitions are equivalent, comparing the data acquired from the same device using UFED Touch and the "dd" tool. The result showed that there was a difference in the hash of two files; however, the author explains that the differences in "user data" are due to unallocated space and changes in the "File System" files, such as "inode table" and "superblock."

The third test [9] consisted of examining the content of an encrypted device to show whether it is possible to find evidence, which was acquired using the "dd" tool, verifying whether physical acquisition of an encrypted device can provide any useful information. It was concluded that if the device is encrypted, the physical acquisition will produce a complete encrypted code, useless for forensic analysis because no information that makes sense can be found.

In the case of old or outdated Android devices, there is the possibility of breaking the encryption. The author [9] also describes a step by step guide to acquiring a mobile device using ISP and JTAG and its subsequent analysis in the UFED Physical Analyzer. The author [9] also sought to highlight the diverse degrees of difficulty when using different tools, such as UFED Touch, which is easy to use, comparing it to ISP and JTAG, with which there is a risk of damaging the device. The main focus of the study was to provide an overview of the acquisition of JTAG and ISP, and present some information about these techniques and to show that the UEFD Physical Analyzer can read the extraction product.

In the study [9], the device used for data acquisition in the experiment was a rooted Samsung GT-I9505 (Galaxy S4) with Android OS Version 5.0.1, with encrypted main memory, and a Nokia Lumia 635, with Windows Phone 8.1.

Sathe and Dongre [45] conducted a comparative study of requirements, capabilities, and limitations of forensic techniques for the logical and physical acquisition of data from mobile devices, using a Samsung Galaxy Grand Duos GT-I9082 device to perform the comparative tests. The following suites and forensic tools were analyzed: UFED, MOBILedit, Oxygen Forensics and XRY, ADB Pull, Backup Analysis, AFLogical, Wondershare Dr. Fone for Android, JTAG, and Chip-Off. Although they address the use of JTAG and Chip-Off, the analyzed device was launched with Android OS Version 4.1.2 and was updated until version 4.2.2, both did not provide, by default, the main memory encryption.

Chanajitt et al. [46] analyzed seven m-banking apps for Android in Thailand and described the forensic artifacts that can be recovered, as well as the results of the applications' assessment concerning security. They describe JTAG as a physically invasive data acquisition method, but that allows access to data without the need for USB debugging enabled, or without the enabled root user (without root) and still circumvent password lock, bypassing the OS security mechanism. In the experiment, a rooted Samsung GT-I9500 (Galaxy S4) and a non-rooted Samsung GT-I9190 (Galaxy S4 Mini) were used, both with Android OS in Version 4.4.2 with unencrypted main memory.

On 28 January 2020, NIST released the results of tests performed using JTAG and Chip-Off to acquire data from damaged mobile devices [47]. The objective was to test the validity of the methods; to see if they produce accurate results with reliability. After data acquisition, the following forensic suites were used for analysis: AXIOM, EnCase Forensic, X-Ways, XRY, UFED, Paraben, and Final Mobile Forensics, being identified locations, texts, photos, social media data, and others. The extracted data was compared to the one loaded initially on each phone, proving that JTAG and chip-off extracted the data without modifying it and that some of the tools used are more efficient for analyzing the data than others, particularly for social media application data. The devices analyzed were from different manufacturers, and the latest version of the Android OS was 5.1. The results are published in a series of free online reports [40].

2.7.3. RAM Data Acquisition

Soares and Sousa Júnior [48] presented a technique for analyzing Java objects data by the extraction of RAM from devices with ARM 32-bit architecture, but with the flexibility to be adapted to other architectures (including 64-bit). The authors disregarded the data of large objects or allocated by native code libraries present in the Android operating system, version 5.0, run in an emulator. For the analysis process, the Volatility framework was used. The authors also created a set of extensions for Volatility that enabled the retrieval of information about the execution environment and the recovery of allocated Java objects, as well as mappings for interpreting the data in the ART, OAT, and DEX files, which allowed the recovery of the execution environment data structures. For experimental validation, they emulated a Nexus 5 device with Android OS Version 5.01 using *goldfish*. For data analysis, they used version 0.4 of the Linux Santoku Distribution, specialized in mobile device forensics. As a result, the authors were able to extract and analyze Java objects with an understanding of storage structures, overcoming traditional techniques based on detecting artifacts intrinsic patterns.

Soares and Sousa Júnior [49] describe a transition case resulting from a new version of Android Runtime (ART). They improved the technique previously developed [48], validating the tools proposed in emulated and real devices with Android OS version 5.0.1, with the physical device rooted, illustrating the difficulties related to forensic analysis due to the different specific implementations by several manufacturers of mobile devices.

In Table 2 we present a summary of the different techniques and methodologies employed in related works.

Table 2. Techniques and methodologies employed in related works.

	Acquisitions Based on Device Firmware Attack	Acquisitions Using ISP, JTAG, and Chip-Off	RAM Data Acquisition
[20]	X	-	-
[7]	X	-	-
[34]	X	-	-
[44]	X	-	-
[43]	X	-	-
[9]	-	X	-
[45]	-	X	-
[46]	-	X	-
[47]	-	X	-
[48]	-	-	X
[49]	-	-	X

3. Problem Statement and Proposed Solution

Although the studies do not point out explicitly, it is observed that the most significant difficulty encountered by digital forensics professionals, with regard to the acquisition of data from mobile devices, are the lock screens configured by the user. Such a restriction prevents the acquisition of data by the extensive list of forensic tools usually employed, with different models and manufacturers.

It corroborates the statement that Cellebrite (Cellebrite DI is an Israeli company founded in 1999 and headquartered in Petah Tikva, which manufactures data extraction and data analysis devices from mobile devices, mass storage devices, and drones. The company is a subsidiary of Sun Corporation, Japan. Cellebrite DI has two subsidiaries, Cellebrite USA Corp. and Cellebrite GmbH, based in Parsippany, New Jersey, USA and Munich, Germany, respectively.) [50] already has 11 laboratories around the world, one of which is installed in Brazil, specialized in advanced unlocking and extraction services using UFED Ultimate [51]. The service offered by these laboratories is called Cellebrite Advanced Service (CAS), whose main objective is to unlock the device by breaking the password by brute force attack. Such a technique can be used in Apple devices with iOS, Samsung, Huawei, and LG devices [52] with an approximate cost of $ 2500.00 (two thousand and five hundred dollars) per unlock.

Another company offering a similar solution is Grayshift, which has developed a product called GrayKey [53], intended exclusively for unlocking iPhones. In addition to the removal or bypass of the lock screen, it is also necessary to make the physical acquisition of the device data, to have access to the content deleted by the user and that of instant messengers, such as WhatsApp, without contaminating the evidence and allowing the analysis by tools specialized, commercial or not.

In some cases, even with the device unlocked, it may not be possible to perform the physical acquisition, being necessary to enable the *root* user on the device to do so. In some models of Android devices, during the procedure performed to *root* the device, the internal memory is safely erased, which is not feasible from a forensic point of view.

In considering these aspects, this section describes the problem addressed in this article and its proposed solution. This solution will result in bypassing the lock screen, rooting the device (if necessary) without the risk of losing or contaminating the user's data, subsequent data acquisition through a forensic tool for mobile devices, and analyzing the acquired data.

3.1. Problem Statement

Mobile device manufacturers have been improving security mechanisms to prevent or hinder unauthorized access to device data. Currently, the user has encryption mechanisms, lock screen mechanisms that automatically prevent other changes that could allow access to data, in addition to other security mechanisms.

The implementation and improvement of such mechanisms prevents or hinders the work of state law enforcement agencies. The judicial and scientific police find it challenging to acquire data from mobile devices that may constitute evidence, given that current forensic suites, both proprietary and open-source, fail to bypass the lock screen mechanisms of some models of mobile devices.

3.2. Proposed Solution and Justification

This work proposes a solution that combines the use of ISP and Combination Firmware aiming to bypass the device lock screen, allowing the physical acquisition (with the possibility of *rooting* the device if necessary), the file system acquisition, and data logical acquisition. After the respective acquisitions, the expert can perform his analysis using specialized forensic tools.

During the process established in the LLDA-ISPCF methodology, it is also possible to backup the partitions that will be changed to use the Combination Firmware. This backup is restored after acquiring the data, leaving the device exactly as it was before, with the current user lock screen.

This methodology applies to a wide range of devices, regardless of the OS version, model, or SoC manufacturer.

The forensic possibilities offered by the LLDA-ISPCF methodology is of great importance for the present work since it can retrieve data contained in mobile devices for the elucidation of the most various crimes. The ability to acquire and analyze data from mobile devices with the latest versions of the Android OS, since a small number of devices still runs android Version 10, represents a significant advance in mobile forensics in the current scenario.

After verifying that the devices collection phase has complied with the digital forensic life cycle stage presented in Figure 3, for expert examination, the methodology proposed in this work is ready to be applied for physical, system, and data acquisition.

3.3. Description of the LLDA-ISPCF Methodology

The methodology is presented in Figure 7, and its phases are detailed below.

3.3.1. Phase 1: Preliminary Check of the Device's Security Mechanisms

At this phase, it should be checked whether the device has any lock screen mechanism. The most common ones found on Android devices are the standard PIN, alphanumeric password, biometrics, facial recognition, and iris scanning. There is also an unlocking screen mechanism by Near Field Communication (NFC) Token and USB, although they are less common. Although, if the lock screen mechanism protects the device, it will be necessary to start the device in Recovery Mode . By doing so, we will be able to get information about the status of its bootloader (locked or unlocked), USB debugging mode (enabled or disabled), FRP status (On or Off), and Firmware version, with special attention to the binary version of the Firmware.

With the device locked, it is necessary to use a combination of keys, which varies according to the manufacturer and the device model. The most common way is, starting from the situation in which the device is turned off, simultaneously pressing the Power and Volume Up keys, and keep them pressed for about 3 s. It is recommended to take note of all the information, including the version of the OS that runs on the device, as this data will be used in the next phases. If the device has USB debugging mode disabled when it is connected to the computer, although it is recognized, it will not be possible to access any file system partition; the smartphone will only recharge the battery via the USB port. If USB debugging mode is enabled, the technique of using the Android Debug Bridge (ADB) [54] can be explored. This technique is a command-line tool that provides access to a Unix shell, is used to execute various commands on an Android OS device, but which will not be covered in this work.

If the device does not have a lock screen mechanism, other tools or even steps of the proposed methodology. However, this hypothesis will not be detailed in this work since it does not address the situation of the enabled security mechanism.

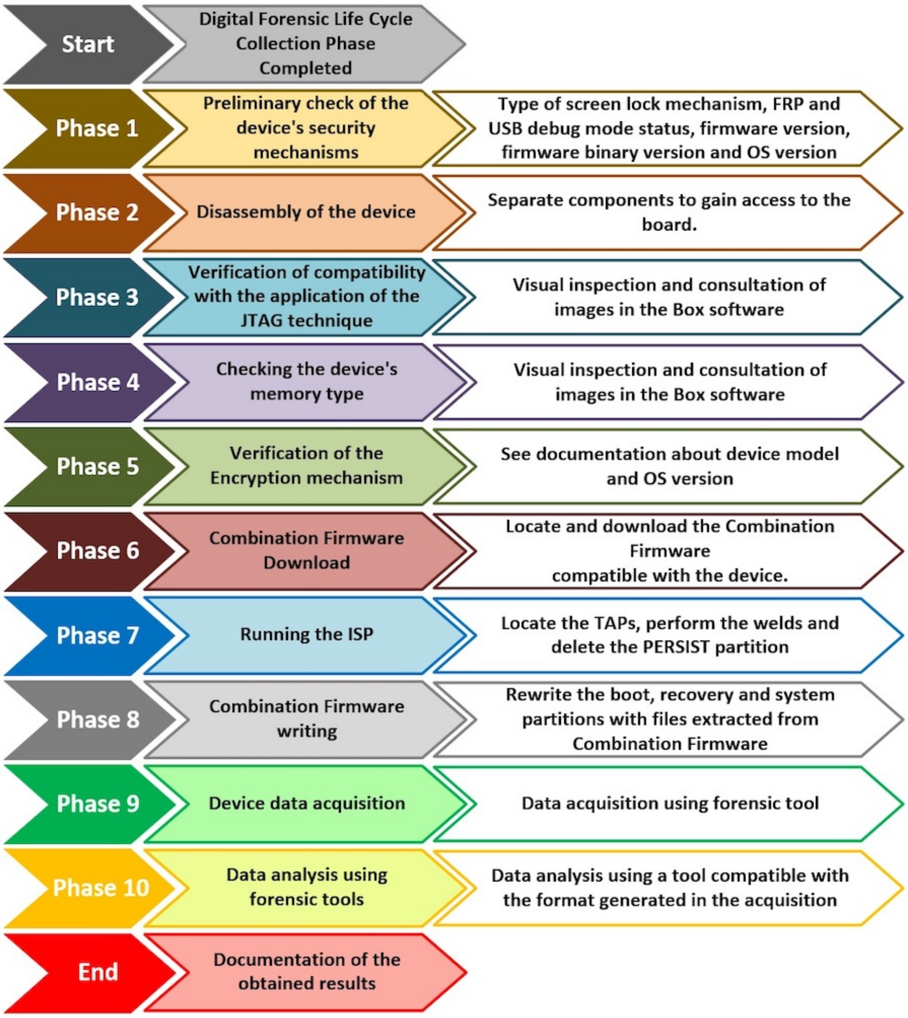

Start
Digital Forensic Life Cycle Collection Phase Completed
Phase 1
Preliminary check of the device's security mechanisms
Type of screen lock mechanism, FRP and USB debug mode status, firmware version, firmware binary version and OS version
Phase 2
Disassembly of the device
Separate components to gain access to the board.
Phase 3
Verification of compatibility with the application of the JTAG technique
Visual inspection and consultation of images in the Box software
Phase 4
Checking the device's memory type
Visual inspection and consultation of images in the Box software
Phase 5
Verification of the Encryption mechanism
See documentation about device model and OS version
Phase 6
Combination Firmware Download
Locate and download the Combination Firmware compatible with the device.
Phase 7
Running the ISP
Locate the TAPs, perform the welds and delete the PERSIST partition
Phase 8
Combination Firmware writing
Rewrite the boot, recovery and system partitions with files extracted from Combination Firmware
Phase 9
Device data acquisition
Data acquisition using forensic tool
Phase 10
Data analysis using forensic tools
Data analysis using a tool compatible with the format generated in the acquisition
End
Documentation of the obtained results

Figure 7. The proposed methodology execution flow.

3.3.2. Phase 2: Disassembly of the Device

This phase consists of removing the display from the device and separating the components to access the board. Besides the display, it is necessary to remove the battery, cables, heat sinks, and anything necessary to perform the solder with adequate precision, without risk of damaging the equipment and making acquisition impossible.

For a display removal, it is ideal to use an LCD/Touch Screen Disassembly tool, equipment that evenly heats the screen at the correct temperature so that the glue used on the LCD is more easily released, reducing the breakage risk. In some devices, it may be necessary to remove segments of the metallic protection of some components to access the TAPs.

3.3.3. Phase 3: Verification of Compatibility with the Application of the JTAG Technique

At this stage, it is necessary to check if the device has the TAPs for the execution of the JTAG. This verification can be carried out visually. If the forensic analyst does not know how to locate the standard JTAG TAPs, he can consult the Box software to check the diagram of the TAPs of the device in question. It should be noted that the most recent devices do not include JTAG TAPs. If the device to be analyzed has a TAP, the JATG technique can be used, according to the works developed by Pappas [9], Sathe, and Dongre [45], in addition to the tests carried out by NIST [47], whose devices were equipped with TAPs for JTAG.

3.3.4. Phase 4: Checking the Device's Memory Type

The ISP, until the elaboration of this work, is only compatible with eMMC and eMCP memories. If the device is equipped with another type of memory, such as UFS, the Chip-Off technique can be used, according to the work of Sathe and Dongre [45] and the tests carried out by NIST [47].

3.3.5. Phase 5: Verification of the Encryption Mechanism

To check whether the main memory is protected with encryption and which model has been applied, it is necessary to consult the version of the OS running on the device, data already obtained in Phase 1 (Section 3.3.1), and to seek technical information about it.

It is necessary to confirm that the use of the Chip-Off technique will carry out the acquisition of encrypted data, and that forensic suites may not be able to decrypt such data. If any encryption does not protect the main memory, it is possible to use the Box software to perform the physical data acquisition, without the need to use the Combination Firmware.

3.3.6. Phase 6: Combination Firmware Download

In this phase, in possession of the firmware version and the binary version of the device, obtained in Phase 1 (Section 3.3.1), it is necessary to locate and download the Combination Firmware compatible with such a device. Some websites make these files available for free. However, it is advisable to download it from specialized websites, reducing the probability of getting files corrupted or infected by malware. One of the most popular websites that provide access to mobile devices maintenance tools is *Halabtech Support* [55].

3.3.7. Phase 7: Running the ISP

This phase begins with a search in the Box software to locate the diagram containing the TAPs location of the exact device model. After identifying the TAPs, it is necessary to weld the conductors. One end of the conductors is soldered on the TAPs, and the other end is weld on the ISP-dedicated connector.

The next step is to plug the ISP connector into the Box. At this point, it is possible to use the eMMC Tool Suite to make the physical acquisition of the device's internal memory. However, the encrypted data acquired can only be analyzed if the forensic suite used for analysis can decrypt them. If it is not possible to decrypt the data, the JTAG Classic Suite software will be used to manipulate the data on the *Persist* partition, without changing the partition where the user data is located.

Android devices have multiple partitions that have different functions [56]. However, because it is an *Open Source* project, manufacturers can make changes or modifications at will. This way, we will have a varying number of partitions in the different device models running with Android OS. Consequently, partitions may vary depending on SoC, brand, OS version, and more.

To install the Combination Firmware, that is, overwrite the *boot*, *recovery*, and *system* partitions, it is necessary to previously overwrite with 0 (zero) the area where the FRP is recorded, and on newer devices, the FRP is usually on the *Persist*. While the FRP is active, it is not possible to overwrite these partitions.

All values for configuring the software required to run the ISP can be obtained in *Seeking the Truth from Mobile Evidence* [57].

3.3.8. Phase 8: Combination Firmware Writing

Following the phases proposed in this methodology, the Combination Firmware writing is the last phase that precedes the acquisition itself. It is necessary to rewrite the *boot*, *recovery*, and *system* partitions. Such partitions will have their original content replaced by the files contained in the Combination Firmware previously obtained. It is advisable to make a backup of the partitions that will be overwritten before performing the operation. It is also possible to extract the contents of the file with *MD5* extension and generate a new compressed file with the same name as the complete file, overwriting it. Doing so will be easier to use the software for writing the firmware.

For the writing operation, both the *Box Octoplus Pro* and the *Odin* tools can be used. The necessary procedures for the Combination Firmware writing using the Box can be obtained in *Seeking the Truth from Mobile Evidence* [57].

After replacing the partitions mentioned above, the device will boot normally, without asking for the password or the mechanism imposed by the user.

The device on which the LLDA-ISPCF methodology was applied uses file-based encryption. When the device is initialized, the system accesses the data stored in its special encrypted area, protected with hardware keys. All data on the *data* partition is inaccessible until the user provides authentication credentials, as this area is protected with keys based on the user's credentials. The user password is one of the cryptographic keys, and when it ceases to exist, only the system password, defined in the source code by the developers, remains. As there is only the default system key, no key exchange takes place, and the partition is automatically decrypted. The process is transparent, and as soon as the initialization process is completed, the system displays the home screen.

After of initialization, the device displays a screen with essential functions, being possible to view photos, explore files, record images, among other basic operations.

From that point on, the device is ready for data acquisition, being possible to activate the USB Debugging Mode, activate the option *Stay Active*, and also, if necessary for the acquisition, root the device. It is important to note that the firmware now installed runs the same Android version as the original firmware. Therefore, the acquisition methods must be compatible with the OS version if such a device is not *rooted*. The choice of the forensic acquisition tool is at the discretion of the specialist who will perform the procedure.

3.3.9. Phase 9: Device Data Acquisition

At this stage, the forensic analyst may select the most appropriate tool for the devices data acquisition. The need to root or not the device by the analyst has to be carefully analyzed. This action is due to his ability to manipulate the forensic tools at his disposal.

3.3.10. Phase 10: Data Analysis Using Forensic Tools

After data acquisition, this step marks the end of the previous phases proposed by the methodology. The forensic analyst will carry out the analysis of the acquired data, devices being able to use any available tools, as long as they are compatible with the data format generated by the tool that performed the acquisition. It is worth mentioning that, in a physical acquisition, it is possible to obtain files containing the user's credentials for accessing online services and social networks, such as Google services (Keep, Photos, Google Drive, etc.), Facebook, Twitter, Instagram, among others. These credentials, when extracted and inserted in software for searching and analyzing data in the *cloud*, function as "tokens", providing authorized access to such services, considerably increasing the volume of data obtained.

4. Case Study: Analysis of a Device

In this section, the actions in each phase of the proposed methodology will be detailed, from the preliminary verification of the security mechanisms to the acquired data analysis.

4.1. Choice of the Device to Validate the Proposed Methodology

For study purposes, we performed the physical data acquisition of a Samsung SM-A105M/DS smartphone (Galaxy A10). The choice for this device took into account four main criteria.

The first was the scope of the device and the global sales volume. According to (the market analysis and business consulting company Counterpoint)[58], Samsung Galaxy A10 (A105M) was the best-selling Android smartphone in the world in the third quarter of 2019, and placed second in overall sales volume, just behind the iPhone XR.

The second criterion was the Android version on the device because the more recent the smartphone, the more layers of security are implemented, and the more secure the system tends to be. In this case, the SM-A105 model runs Android version 9.0 (Pie), which runs on most Android devices (41.9%) [59].

The third criterion was the difficulty in carrying out the welds on the board, considering that the device has tiny components. Therefore, without using a VR-Table for the realization of an ISP, it is still possible to weld the conductors on the TAPs.

The fourth criterion is the fact that the SM-A105 data has device encryption enabled automatically (Direct Boot).

4.2. Application of the Methodology

Before using the proposed methodology, we performed all kinds of extraction methods available on UFED Touch 2 (software version 7.32.0.68). These methods are compatible with the characteristics of the SM-A105M/DS device (Galaxy A10), which had an activated lock screen (alphanumeric characters), bootloader blocked, USB debugging mode disabled, and FRP active. Unfortunately, we realized that none of these methods worked in the attempt of unlocking those smartphones. All steps illustrated in Figure 8 will be applied to validate the proposed methodology.

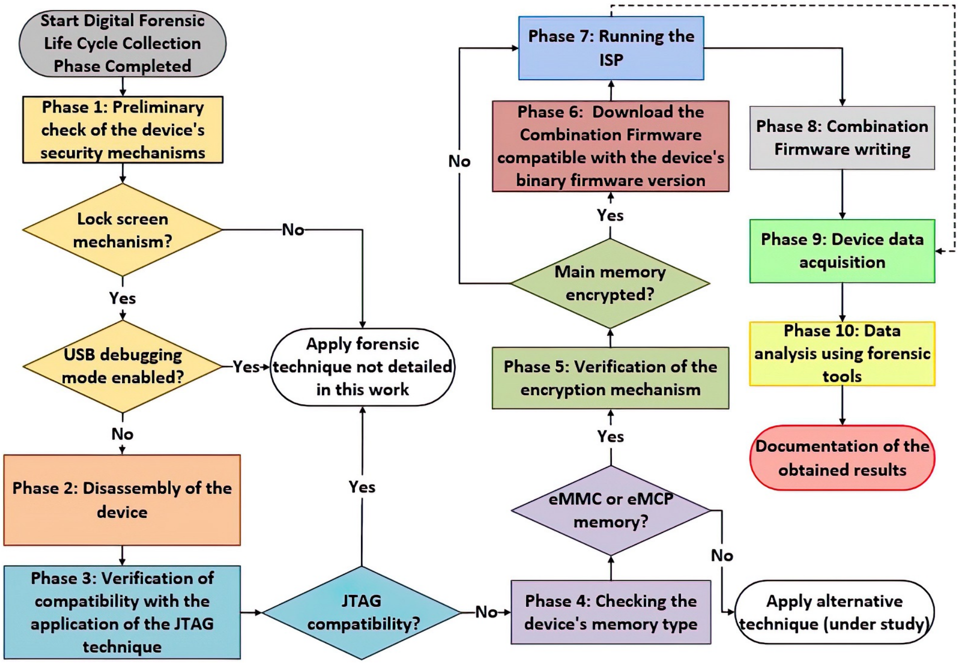
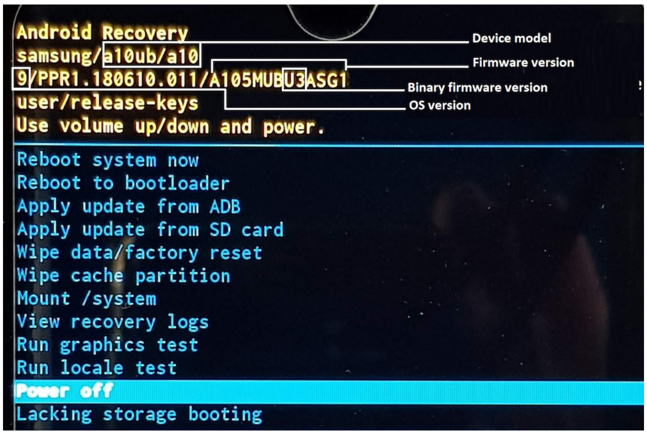

Start Digital Forensic Life Cycle Collection Phase Completed
Phase 1: Preliminary check of the device's security mechanisms
Lock screen mechanism?
No
Yes
USB debugging mode enabled?
Yes
Apply forensic technique not detailed in this work
No
Phase 2: Disassembly of the device
Yes
Phase 3: Verification of compatibility with the application of the JTAG technique
JTAG compatibility?
No
Phase 4: Checking the device's memory type
eMMC or eMCP memory?
No
Apply alternative technique (under study)
Yes
Phase 5: Verification of the encryption mechanism
Main memory encrypted?
Yes
No
Phase 6: Download the Combination Firmware compatible with the device's binary firmware version
Phase 7: Running the ISP
Phase 8: Combination Firmware writing
Phase 9: Device data acquisition
Phase 10: Data analysis using forensic tools
Documentation of the obtained results

Figure 8. Flowchart of the LLDA-ISPCF methodology applied in the case study.

Each phase of the methodology proposed in this work to analyze the device in question will be detailed below.

4.2.1. Phase 1: Preliminary Check of the Device's Security Mechanisms

In the case study, the device had an activated lock screen (alphanumeric characters), bootloader blocked, USB debugging mode disabled, and FRP active. Figure 9 displays information about the devices firmware version, OS version, device model, and binary firmware version.



```
Android Recovery
samsung/a10ub/a10
9/PPR1.180610.011/A105MUBU3ASG1
user/release-keys
Use volume up/down and power.
Reboot system now
Reboot to bootloader
Apply update from ADB
Apply update from SD card
Wipe data/factory reset
Wipe cache partition
Mount /system
View recovery logs
Run graphics test
Run locale test
Power off
Lacking storage booting
```

Figure 9. Highlighted, the device binary version.

4.2.2. Phase 2: Disassembly of the Device

At this stage, the devices display and battery were removed to allow access to the board. Flat cables and other connectors were disconnected so that no components were damaged during the welding process. Due to the difficulty in accessing the TAPs, it was necessary to remove metal parts that serve as protection for the components, as can be seen in Figure 10.

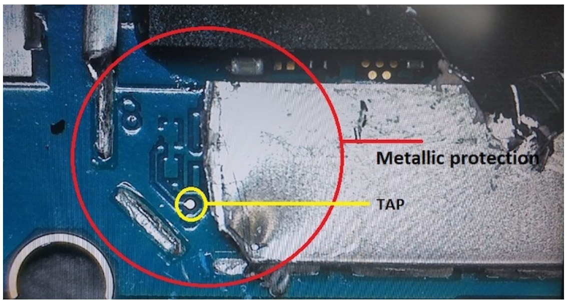


Figure 10. Metal part removal to allow access to the TAP.

4.2.3. Phase 3: Verification of Compatibility with the Application of the JTAG Technique

At this stage, we checked if standard TAPs were present on the device board before applying the JTAG. The execution of this phase was a mere protocol, as the device was launched in 2019 and did not have TAPs for a JTAG execution (present only in older devices).

4.2.4. Phase 4: Checking the Device's Memory Type

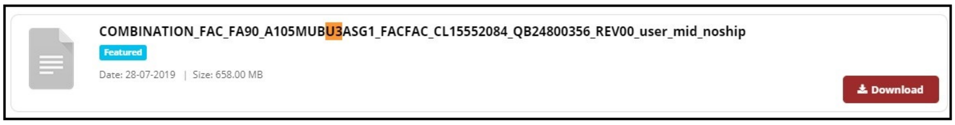
The device used in the case study had eMMC type memory and was therefore compatible with ISP. If the device was equipped with another type of memory, other techniques could be employed, such as Chip-Off.

4.2.5. Phase 5: Verification of the Encryption Mechanism

The device used in the case study (SM-A105M/DS) uses FBE type encryption, and Android OS Version 9 (Pie).

4.2.6. Phase 6: Combination Firmware Download

After identification, a search was performed to obtain the Combination Firmware. After locating the firmware-related files, those were downloaded to the local computer. Figure 11 shows the Combination Firmware serial number for download.



COMBINATION_FAC_FA90_A105MUBU3ASG1_FACFAC_CL15552084_QB24800356_REV00_user_mid_noship

Date: 28-07-2019 | Size: 658.00 MB

Figure 11. Combination Firmware download.

4.2.7. Phase 7: Running the ISP

In this phase, the research was managed in the Box software to locate the diagram containing the TAPs location of the exact device model. After identifying the TAPs, the conductors were soldered on the TAPs. Figure 12 shows, with the help of a microscope, one of the conductors already welded.

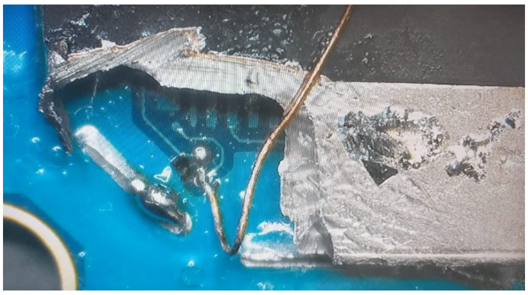
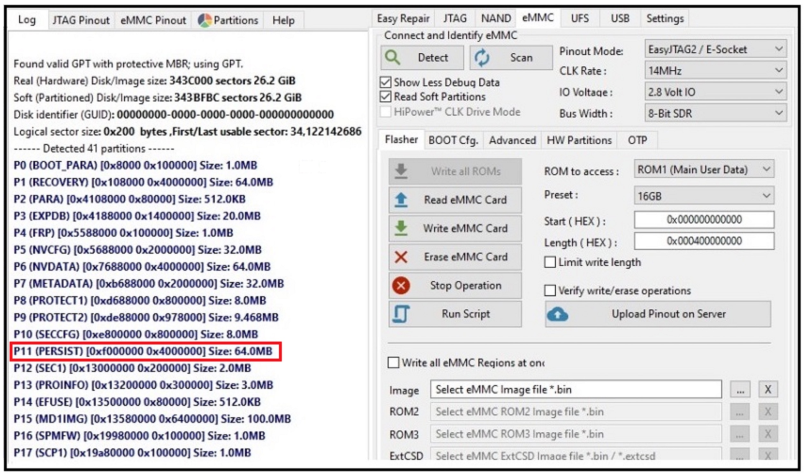


Figure 12. Conductor welded to the TAP.

Once the necessary conductors were soldered, the TAPs were connected to the corresponding points on the ISP connector, and then, connected to the Box. The *JTAG Classic Suite* utility was started. Figure 13 shows part of the *JTAG Classic Suite* utility screen, showing the primary memory partition of the device and the values of each parameter to manipulate the *Persist* partition. Such values may vary depending on the device and the eMMC manufacturer. The values referring to the parameters were obtained in *Seeking the Truth from Mobile Evidence* [57].



```
Found valid GPT with protective MBR; using GPT.
Real (Hardware) Disk/Image size: 343C000 sectors 26.2 GiB
Soft (Partitioned) Disk/Image size: 343BFBC sectors 26.2 GiB
Disk identifier (GUID): 00000000-0000-0000-0000-000000000000
Logical sector size: 0x200  bytes ,First/Last usable sector: 34,122142686
------ Detected 41 partitions ------
P0 (BOOT_PARA) [0x8000 0x100000] Size: 1.0MB
P1 (RECOVERY) [0x108000 0x4000000] Size: 64.0MB
P2 (PARA) [0x4108000 0x80000] Size: 512.0KB
P3 (EXPDB) [0x4188000 0x1400000] Size: 20.0MB
P4 (FRP) [0x5588000 0x100000] Size: 1.0MB
P5 (NVCFG) [0x5688000 0x2000000] Size: 32.0MB
P6 (NVDATA) [0x7688000 0x4000000] Size: 64.0MB
P7 (METADATA) [0xb688000 0x2000000] Size: 32.0MB
P8 (PROTECT1) [0xd688000 0x800000] Size: 8.0MB
P9 (PROTECT2) [0xde88000 0x978000] Size: 9.468MB
P10 (SECCFG) [0xe800000 0x800000] Size: 8.0MB
P11 (PERSIST) [0xf000000 0x4000000] Size: 64.0MB
P12 (SEC1) [0x13000000 0x200000] Size: 2.0MB
P13 (PROINFO) [0x13200000 0x300000] Size: 3.0MB
P14 (EFUSE) [0x13500000 0x80000] Size: 512.0KB
P15 (MD1IMG) [0x13580000 0x6400000] Size: 100.0MB
P16 (SPMFW) [0x19980000 0x100000] Size: 1.0MB
P17 (SCP1) [0x19a80000 0x100000] Size: 1.0MB
```

☑ Show Less Debug Data
☑ Read Soft Partitions
□ HiPower™ CLK Drive Mode

Pinout Mode: EasyJTAG2 / E-Socket
CLK Rate : 14MHz
IO Voltage : 2.8 Volt IO
Bus Width : 8-Bit SDR

ROM to access : ROM1 (Main User Data)
Preset : 16GB
Start (HEX) : 0x000000000000
Length (HEX) : 0x000400000000
□ Limit write length
□ Verify write/erase operations

□ Write all eMMC Regions at onc

Image Select eMMC Image file *.bin
ROM2 Select eMMC ROM2 Image file *.bin
ROM3 Select eMMC ROM3 Image file *.bin
ExtCSD Select eMMC ExtCSD Image file *.bin / *.extcsd

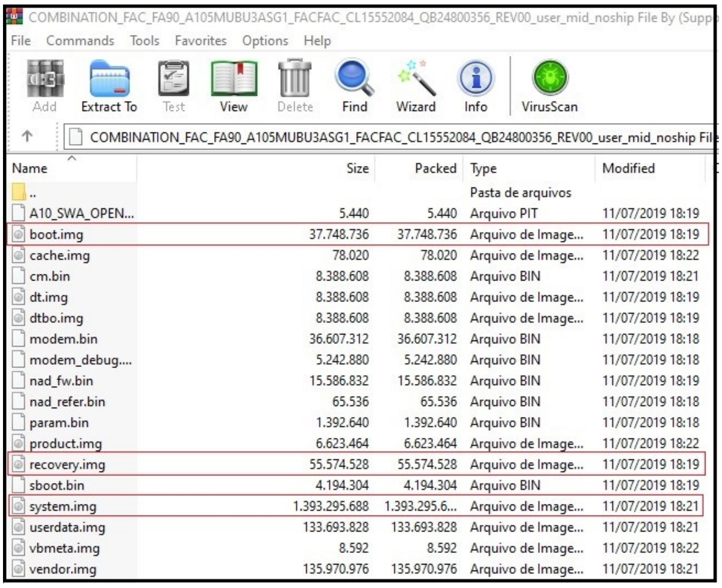
Figure 13. *JTAG Classic Suite* home screen.

We then recorded an FRP in the smartphone as to manipulate its *Persist* partition. After setting the correct parameters, the *Persist* partition was overwritten. Nevertheless, it is safe to mention that the partition with the user's data was not invaded, and therefore the data remained intact.

4.2.8. Phase 8: Combination Firmware Writing

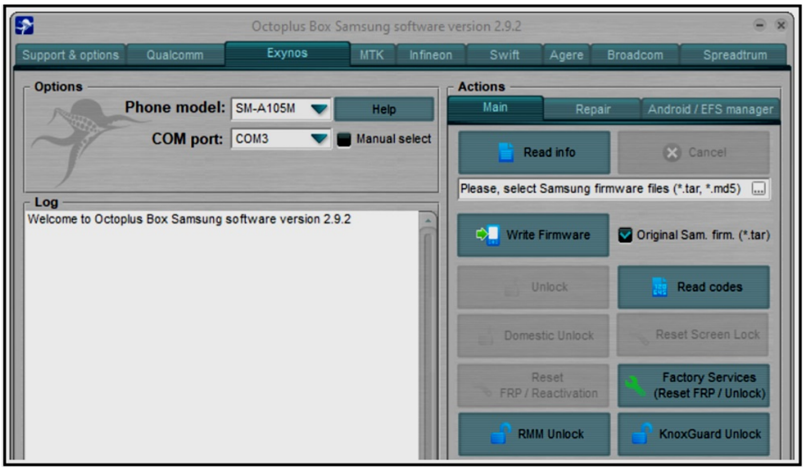
For a Combination Firmware writing, only relevant files that correspond to the *boot*, *recovery*, and *system* partitions should be preserved. The content of the file with extension *MD5* can be seen in Figure 14.

Next, the *Box Octoplus Pro* and *Octoplus Box Samsung Software* were used for replacing the partitions specified in Figure 14. Figure 15 shows the *Octoplus Box Samsung Software* tool home screen. All settings were obtained in *Seeking the Truth from Mobile Evidence* [57].


COMBINATION_FAC_FA90_A105MUBU3ASG1_FACFAC_CL15552084_QB24800356_REV00_user_mid_noship File By (Supp
File
Commands
Tools
Favorites
Options
Help
Add
Extract To
Test
View
Delete
Find
Wizard
Info
VirusScan
COMBINATION_FAC_FA90_A105MUBU3ASG1_FACFAC_CL15552084_QB24800356_REV00_user_mid_noship Fil

Name	Size	Packed	Type	Modified
..			Pasta de arquivos	
A10_SWA_OPEN...	5.440	5.440	Arquivo PIT	11/07/2019 18:19
boot.img	37.748.736	37.748.736	Arquivo de Image...	11/07/2019 18:19
cache.img	78.020	78.020	Arquivo de Image...	11/07/2019 18:22
cm.bin	8.388.608	8.388.608	Arquivo BIN	11/07/2019 18:21
dt.img	8.388.608	8.388.608	Arquivo de Image...	11/07/2019 18:19
dtbo.img	8.388.608	8.388.608	Arquivo de Image...	11/07/2019 18:19
modem.bin	36.607.312	36.607.312	Arquivo BIN	11/07/2019 18:18
modem_debug....	5.242.880	5.242.880	Arquivo BIN	11/07/2019 18:18
nad_fw.bin	15.586.832	15.586.832	Arquivo BIN	11/07/2019 18:19
nad_refer.bin	65.536	65.536	Arquivo BIN	11/07/2019 18:18
param.bin	1.392.640	1.392.640	Arquivo BIN	11/07/2019 18:18
product.img	6.623.464	6.623.464	Arquivo de Image...	11/07/2019 18:22
recovery.img	55.574.528	55.574.528	Arquivo de Image...	11/07/2019 18:19
sboot.bin	4.194.304	4.194.304	Arquivo BIN	11/07/2019 18:19
system.img	1.393.295.688	1.393.295.6...	Arquivo de Image...	11/07/2019 18:21
userdata.img	133.693.828	133.693.828	Arquivo de Image...	11/07/2019 18:21
vbmeta.img	8.592	8.592	Arquivo de Image...	11/07/2019 18:22
vendor.img	135.970.976	135.970.976	Arquivo de Image...	11/07/2019 18:21

Figure 14. SM-A105M most relevant Combination Firmware files.



Phone model: SM-A105M

COM port: COM3

■ Manual select

Please, select Samsung firmware files (*.tar, *.md5)

☑ Original Sam. firm. (*.tar)

Log

Welcome to Octoplus Box Samsung software version 2.9.2

Figure 15. *Octoplus Box Samsung Software* home screen.

After replacing the *boot*, *recovery*, and *system* partitions, the device restarted without asking for the lock screen password, allowing the viewing of media files, installed applications, and all content (files) present on the device. It should be noted that the firmware newly installed, runs the same Android version that runs on the original firmware, in this case, version 9.

4.2.9. Phase 9: Physical Acquisition Using UFED Touch 2

In this phase, we selected the forensic tool suitable for physical data acquisition. As we understand that physical data acquisition can be carried out by other forensic tools specific to other mobile devices, the option for UFED Touch 2 was motivated by a Cellebrite announcement. They stated that starting with version 7.23, the UFED Touch 2, 4PC, and InField would be able to perform the physical data acquisition of "A" and "J" Samsung's line-up equipped with SoC *Exynos*, using a generic profile [60].

With the device already unlocked, physical extraction attempts were made using UFED Touch 2, with software before version 7.32.0.68. We realized that it was not possible to perform the physical acquisition without the root user being enabled. Therefore, in this case, it was possible to perform the file system acquisition and the logical acquisition without rooting the device. Figure 16 shows part of the UFED Touch 2 screen. Highlighted, is the method that allows the physical acquisition of the SM-A105M/DS, enabling the root user.

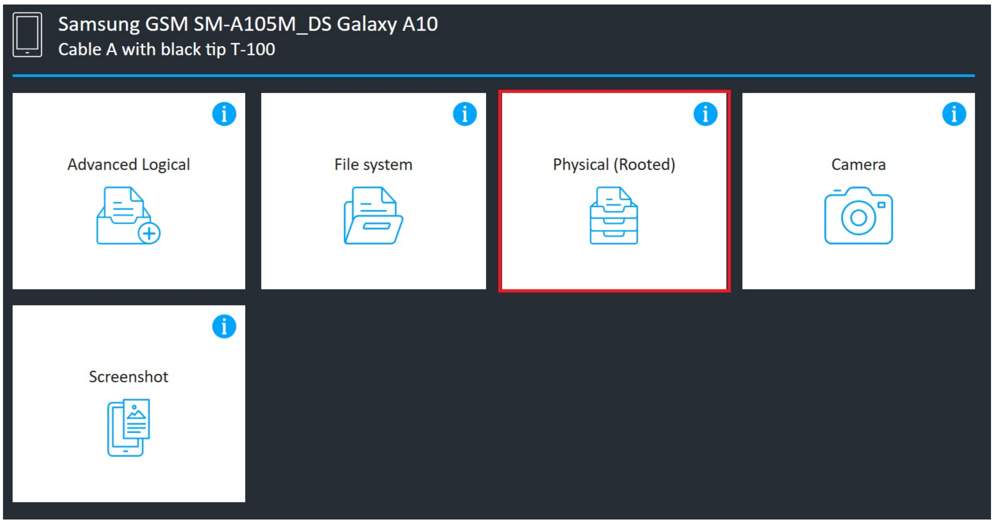

Samsung GSM SM-A105M_DS Galaxy A10
Cable A with black tip T-100
Advanced Logical
File system
Physical (Rooted)
Camera
Screenshot

Figure 16. Possibility of physical acquisition of the SM-A105M/DS.

Using version 7.32.0.68 of the UFED Touch 2 software and selecting a generic profile, was possible to perform the physical acquisition without the need to enable the root user, in addition to the File System and Logic acquisitions. It is worth remembering that UFED Touch 2 was unable to bypass the lock screen in the experiments carried out before the methodology proposed in this work, although the tool promises to do so. Figure 17 shows a part of the UFED Touch 2 screen and, highlighted, the aforementioned functionality, allowing the physical acquisition of the SM-A105M/DS.

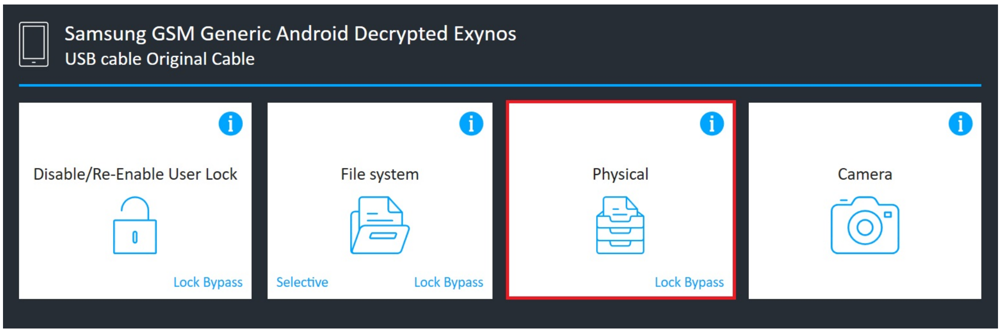
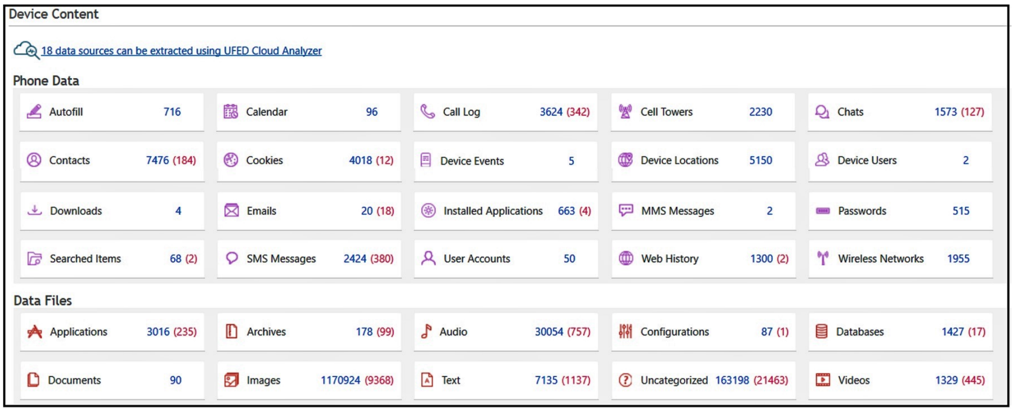


Figure 17. Physical acquisition of the SM-A105M/DS.

4.2.10. Phase 10: Data Analysis Using UFED Physical Analyzer

In this phase, the data acquired with UFED Touch 2 were analyzed using the software UFED Physical Analyzer Version 7.32.0.16. The extraction was indexed without errors, and the results were consistent, including the credentials for services in the *cloud*, which can be used in applications with acquisition capacity specifically aimed explicitly for such services, such as the UFED Cloud Analyzer.

Although the objective of this work is not to compare the capabilities of forensic suites, the acquired data were also analyzed with the AXIOM Examiner and XRY XAMN tools, all presenting consistent results from the perspective of forensic analysis. It should also be noted that data acquisitions can be carried out with one tool, and the analysis with another, being limited only to the format generated by the acquisition tool. Figure 18 shows the list of recovered files that were analyzed, including files deleted by the user, and that had not yet been overwritten. The files from instant messaging applications, such as WhatsApp, were also recovered.



Device Content

18 data sources can be extracted using UFED Cloud Analyzer

Phone Data

Autofill	716	Calendar	96	Call Log	3624 (342)	Cell Towers	2230	Chats	1573 (127)
Contacts	7476 (184)	Cookies	4018 (12)	Device Events	5	Device Locations	5150	Device Users	2
Downloads	4	Emails	20 (18)	Installed Applications	663 (4)	MMS Messages	2	Passwords	515
Searched Items	68 (2)	SMS Messages	2424 (380)	User Accounts	50	Web History	1300 (2)	Wireless Networks	1955

Data Files

Applications	3016 (235)	Archives	178 (99)	Audio	30054 (757)	Configurations	87 (1)	Databases	1427 (17)
Documents	90	Images	1170924 (9368)	Text	7135 (1137)	Uncategorized	163198 (21463)	Videos	1329 (445)

Figure 18. Recovered files.

4.3. Comparative Analysis and Discussions

After completing the execution of the proposed methodology in the case study, the results will be compared to related works. A discussion of the proposed methodology concerning its efficiency and limitations will also be presented.

The methodology proposed in this work expands the possibilities of extracting data from mobile devices and can be used in devices with different characteristics. The acquisition capabilities in relation to the characteristics presented by the devices are as follows:

- **Capability 1 (C1)**: Data acquisition from devices that have an enabled lock screen mechanism;
- **Capability 2 (C2)**: Data acquisition from devices with FDE disk encryption;
- **Capability 3 (C3)**: Data acquisition from devices with FBE disk encryption;
- **Capability 4 (C4)**: Data acquisition from devices with OS up to Version 9;
- **Capability 5 (C5)**: Data acquisition from devices with eMMC and eMCP memory types;
- **Capability 6 (C6)**: Data acquisition regardless of SoC manufacturer;
- **Capability 7 (C7)**: Data acquisition from devices with F2FS and EXT4 file systems;
- **Capability 8 (C8)**:Data acquisition from physical devices.

Table 3 shows the comparison between the results achieved using the proposed methodology and the results obtained by the tools and methodologies presented in the related works. Concerning the data acquisition from mobile devices, the capabilities were synthesized to ease the visualization of what is common to each referenced work.

By analyzing the related works, it appears that there is a great effort to improve tools, techniques, and methodologies in favor of data acquisition for forensic analysis. This effort is due to the highest difficulty to overcome or circumvent the security mechanisms that protect the devices data against unauthorized access. It is observed that such mechanisms also hamper law enforcement, as specialists encounter the barriers mentioned above, even when some legal devices and mechanisms support allow the analyst to access such data.

Table 3. Capabilities of both the proposed methodology and the related works.

	Proposed Methodology	[20]	[7]	[34]	[44]	[43]	[9]	[45]	[46]	[47]	[48]	[49]
C1	X	X	X	X	X	-	-	X	-	X	-	-
C2	X	-	-	X	X	-	X	-	-	-	-	-
C3	X	-	-	X	X	-	-	-	-	-	-	-
C4	X	-	-	-	-	-	-	-	-	-	-	-
C5	X	-	X	X	X	X	X	X	X	X	X	X
C6	X	X	-	-	-	X	X	X	X	X	X	X
C7	X	X	X	X	X	-	X	X	X	X	X	X
C8	X	X	X	X	X	X	X	X	X	X	-	X

Among the mechanisms available on all smartphones running the Android OS, the lock screen, when activated, prevents access to the devices data and functionalities. It also prevents communication via the USB port, as the USB debugging mode is disabled by default. Although the forensic tools on the market promise to remove or bypass the lock screen on the overwhelming majority of smartphone models running the Android OS, this does not work in practice. In a wide range of models, the use of tools from various manufacturers, such as UFED, XRY, AXIOM, among others, is ineffective in extracting data when the lock screen is activated.

An example is the model SM-A105 and its variants, whose commercial name is Galaxy A10. The methodology proposed in the case study of this work used that device, which is sold globally, runs Android OS version 9 (Pie), and was the second best-selling smartphone in the world in the third quarter of 2019 [58]. In this case, conventional forensic solutions can only extract data if the device is unlocked.

The application of the proposed methodology, which combines the use of ISP and Combination Firmware, made it possible to bypass the lock screen and the subsequent acquisition and analysis of the data. Also, it is emphasized that the data integrity was preserved, which is fundamental in forensic analysis. Taking into account that Samsung is the company with the largest market share of mobile devices [3], we believe that its application is a worthy contribution to forensic analysts of law enforcement divisions, due to the abundance of devices and the number of security barriers that can be overcome.

5. Conclusions and Future Work

Given the results obtained in the case study, it is possible to ratify the efficiency of the proposed methodology, especially concerning the possibility of circumventing the lock screen mechanism, which constitutes one of the main obstacles to the execution of mobile devices forensics. The case study carried out to validate the proposed methodology proved that the combination of the listed techniques is feasible and presents satisfactory results. It was possible to bypass the lock screen, which is one of the main obstacles to mobile devices forensics. This characteristic allowed the device data acquisition.

Concerning encryption, with the FRP removal previously stored in the *Persist* partition, it was possible to change the partitions, *boot*, *recovery*, and *system*, and thus bypass the process of exchanging cryptographic keys, because the private key was removed, and the system understands that no blocking mechanism has been configured or was removed, so the data partition (*/data*) will be decrypted without any brute force attack or reverse engineering of the cryptographic algorithm, just bypassing the mechanism transparently.

It is important to note that the methodology has already been applied to other devices manufactured by Samsung, with Qualcomm, Exynos and MediaTek SoCs, and Xiaomi devices with Qualcomm SoCs. According to the studies that have been carried out, it is possible to apply the methodology on any devices that have TAPs for ISP and whose Special Firmware is available. However, we have not yet carried out tests with all devices, considering that the objective of the study is to address devices recently launched and that have a high volume of sales, which represents a greater attack surface.

It is crucial to confirm that the tool used for data extraction and analysis (AXIOM, XRY, among others) is at the analyst discretion or according to availability. Also, it is still possible to combine other tools to make additional acquisitions.

In devices with a security *patch* or latest binary firmware version, modifications in the architecture or system organization may occur. These modifications may also occur in the implementation of a new security mechanism that prevents the methodology from working precisely as detailed, requiring the analysis of specific cases that are beyond the control of the proposed actions.

Future Work

As future work, we intend to add to the set of techniques already applied, the *Chip-off* technique, and the technique presented by Alenadl et al. [34], aiming to bypass Samsung's secure boot mechanism. A methodology for bypassing the lock screen password without using the Combination Firmware is also being studied, as well as a technique that deals with component swapping, including damaged smartphones. Finally, we intend to study techniques that allow data acquisition from smartphones that use the Universal Flash Storage (UFS) memory type.

Author Contributions: C.M.d.S. developed the necessary combinations for the technique; G.D.A.N. conducted the concept and methodology reviews; R.T.d.S.J. proposed experiments and methodology validation; R.d.O.A., G.A.d.O.J. and A.L.S.O. assisted in the development of the methodology; R.d.O.A., G.A.d.O.J. and L.J.G.V. reviewed the techniques applied. All authors have revised the results and the final version of the manuscript.

Funding: Authors of this research received the following funding: R.T.D.S.J. was funded by CAPES (grants 23038.007604/2014-69 FORTE and 88887.144009/2017-00 PROBRAL), CNPq (grants 312180/2019-5 PQ-2, BRICS2017-591 LargEWiN, and 465741/2014-2 INCT in Cybersecurity), FAP-DF (grant 0193.001366/2016 UIoT) and the Administrative Council for Economic Defense (grant CADE 08700.000047/2019-14). R.d.O.A. was funded by FAP-DF (grant 0193.001365/2016 SSDDC). G.D.A.N was funded by the Ministry of the Economy (grant ENAP 083/2016) and the General Attorney of the Union (grant AGU 697.935/2019). The experiments were made in a laboratory that received funding from cooperation projects with the Ministry of the Economy (grants DIPLA 005/2016 and ENAP 083/2016), the Institutional Security Office of the Presidency of the Republic (grant ABIN 002/2017), the Administrative Council for Economic Defense (grant CADE 08700.000047/2019-14) and the General Attorney of the Union (grant AGU 697.935/2019).

Acknowledgments: The authors would like to thank the support of the Brazilian research, development and innovation agencies CAPES (grants 23038.007604/2014-69 FORTE and 88887.144009/2017-00 PROBRAL), CNPq (grants 312180/2019-5 PQ-2, BRICS2017-591 LargEWiN, and 465741/2014-2 INCT in Cybersecurity) and FAP-DF (grants 0193.001366/2016 UIoT and 0193.001365/2016 SSDDC), as well as the cooperation projects with the Ministry of the Economy (grants DIPLA 005/2016 and ENAP 083/2016), the Institutional Security Office of the Presidency of the Republic (grant ABIN 002/2017), the Administrative Council for Economic Defense (grant CADE 08700.000047/2019-14) and the General Attorney of the Union (grant AGU 697.935/2019).

Conflicts of Interest: The authors declare that there are no conflicts of interest.

Abbreviations

Abbreviations used throughout the manuscript:

ART	Android Runtime
BGA	Ball Grid Array
CC	Common Criteria
EDL	Emergency Download Mode
eMCP	Embedded Multi-Chip Package
eMMC	Embedded Multimedia Card
EXT4	Fourth Extended Filesystem
FBE	File-Based Encryption
FDE	Full-Disk encryption
FRP	Factory Reset Protection
F2FS	Flash-Friendly File System
IPED	Digital Evidence Processor and Indexer
ISO	International Organization for Standardization
ISP	In-System Programming
JATG	Joint Test Action Group
LLDA-ISPCF	Low-Level Data Acquisition ISPCF
NFC	Near Field Communication
NIST	National Institute of Standards and Tecnology
OEM	Original Equipment Manufacturer
OMA	Open Mobile Alliance
OTA	Over-the-air
OS	Operating System
SE	Secure Element
SMD	Surface-Mount Device
SoC	System on a chip
ROM	Read-Only Memory
TAP	Test Acess Point

TEE	Trusted Execution Environment
TSK	The Sleuth Kit
UFED	Universal Forensic Extraction Device
UFS	Universal Flash Storage
USB	Universal Serial Bus

References

1. Vargas, F.G. 30ª Pesquisa Anual do FGVcia da FGV/EAESP. 2019. Available online: https://eaesp.fgv.br/sites/eaesp.fgv.br/files/noticias2019fgvcia_2019.pdf (accessed on 19 January 2020).
2. StatCounter. Desktop vs. Mobile vs. Tablet Market Share Worldwide. Available online: https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet (accessed on 28 February 2020).
3. StatCounter. Mobile Vendor Market Share Worldwide. Available online: https://gs.statcounter.com/vendor-market-share/mobile (accessed on 28 February 2020).
4. StatCounter. Operating System Market Share Worldwide. Available online: https://gs.statcounter.com/os-market-share (accessed on 28 February 2020).
5. Simão, A.M.d.L.; Sícoli, F.C.; de Melo, L.P.; de Deus, F.E.; de Sousa Júnior, R.T. Acquisition of digital evidence in android smartphones. *Int. J. Forensic Comput. Sci.* **2011**, 28–43. doi:10.5769/J201101002. [CrossRef]
6. Kingston, C. The Future of Mobile Forensics. Master's Thesis, Utica College, Utica, NY, USA, 2018.
7. Wu, S.; Xiong, X.; Zhang, Y.; Tang, Y.; Jin, B. A general forensics acquisition for Android smartphones with qualcomm processor. In Proceedings of the 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 27–30 October 2017; pp. 1984–1988.
8. The New York Times. F.B.I. Finds Links Between Pensacola Gunman and Al Qaeda. Available online: https://www.nytimes.com/2020/05/18/us/politics/justice-department-al-qaeda-florida-naval-base-shooting.html (accessed on 2 June 2020).
9. Pappas, S. Investigation of JTAG and ISP Techniques for Forensic Procedures. Master's Thesis, University of Tartu, Tartu, Estonia, 2017.
10. Mota Filho, J.E. *Discovering Linux—3rd Edition: Understand the GNU/Linux Operating System*; Novatec Editora: São Paulo, Brazil, 2012.
11. Linux Kernel Organization. WHAT IS Flash-Friendly File System (F2FS)? Available online: https://www.iso.org/standard/44381.html (accessed on 20 October 2019).
12. Linux Kernel Organization. ext4. Available online: https://www.kernel.org/doc/Documentation/filesystems/ext4/ondisk/about.rst (accessed on 20 October 2019).
13. Venkateswara Rao, V.; Chakravarthy, A. Survey on android forensic tools and methodologies. *Int. J. Comput. Appl.* **2016**, *154*, 17–21.
14. Ajijola, A.; Zavarsky, P.; Ruhl, R. A review and comparative evaluation of forensics guidelines of NIST SP 800-101 Rev.1:2014 and ISO/IEC 27037:2012. In Proceedings of the World Congress on Internet Security (WorldCIS-2014), London, UK, 8–10 December 2014; pp. 66–73.
15. Ayers, R.P.; Brothers, S.; Jansen, W. *Guidelines on Mobile Device Forensics*; Technical Report; NIST: Gaithersburg, MD, USA, 2014.
16. International Organization for Standardization. ISO/IEC 27037:2012 Information Technology—Security Techniques—Guidelines for Identification, Collection, Acquisition and Preservation of Digital Evidence. Available online: https://www.iso.org/standard/44381.html (accessed on 20 October 2019).
17. Neuner, S.; Schmiedecker, M.; Weippl, E. Effectiveness of File-Based Deduplication in Digital Forensics. *Secur. Commun. Netw.* **2016**, *9*, 2876–2885. [CrossRef]
18. Brezinski, D.; Killalea, T. RFC 3227—Guidelines for Evidence Collection and Archiving. Available online: https://tools.ietf.org/html/rfc3227 (accessed on 20 October 2019).
19. Lopes, P.A. Digital Forensic—Computational Forensic Expertise. Available online: https://periciacomputacional.com/pericia-forense-computacional-2/ (accessed on 20 October 2019)
20. Yang, S.J.; Choi, J.H.; Kim, K.B.; Chang, T. New acquisition method based on firmware update protocols for Android smartphones. *Digital Investig.* **2015**, *14*, S68–S76. [CrossRef]

21. Octoplus PRO Box. Available online: https://octoplusbox.com/pt/products/products/ (accessed on 20 October 2019).
22. UFI-Box. UFI BOX Overview. Available online: https://www.ufi-box.com/pages/ufi-box-features (accessed on 20 October 2019).
23. EasyJTAG. EasyJTAG Plus Box Hardware. Available online: http://easy-jtag.com/easyjtag-2nd-generation-hw/ (accessed on 20 October 2019).
24. Afonin, O.; Katalov, V. *Mobile Forensics—Advanced Investigative Strategies*; Packt Publishing: Birmingham, UK, 2016.
25. Multi-COM. VR-TABLE EMMC, JTAG, FBUS—User Manual—rev 1.0b. Available online: https://teeltechcanada.com/2015/wp-content/uploads/2016/09/VR-Table_user_manual_EN.pdf (accessed on 20 April 2020).
26. Cellebrite. UFED Ultimate. Available online: https://www.cellebrite.com/pt/ufed-ultimate-4/ (accessed on 22 January 2020).
27. MSAB Systemation. XRY. Available online: https://www.msab.com/ (accessed on 22 January 2020).
28. Magnet Forensics Inc. Magnet AXIOM. Available online: https://www.magnetforensics.com/products/magnet-axiom/ (accessed on 22 January 2020).
29. Morgillo, I.; Viola, S. *Learning Embedded Android N Programming*; Packt Publishing: Birmingham, UK, 2016.
30. Parry, T.O.; Carter, J.L. Updating Firmware on Mobile Devices. U.S. Patent 9,069,641, 30 June 2015.
31. XDA-Developers. Xdadevelopers. Available online: https://www.xda-developers.com/ (accessed on 22 January 2020).
32. Almehmadi, T.; Batarfi, O. Impact of Android Phone Rooting on User Data Integrity in Mobile Forensics. In Proceedings of the 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 1–3 May 2019; pp. 1–6.
33. Loftus, R.; Baumann, M.; van Galen, R.; de Vries, R. Android 7 File Based Encryption and the Attacks Against It, 2017. Available online: https://delaat.net/rp/2016-2017/p45/report.pdf (accessed on 18 December 2019).
34. Alendal, G.; Dyrkolbotn, G.O.; Axelsson, S. Forensics acquisition—Analysis and circumvention of samsung secure boot enforced common criteria mode. *Digit. Investig.* **2018**, *24*, S60–S67. [CrossRef]
35. Samsung Co. File-Based Encryption (FBE) and Full-Disk Encryption (FDE). Available online: https://support.samsungknox.com/hc/en-us/articles/360039577713-File-based-encryption-FBE-and-full-disk-encryption-FDE- (accessed on 28 February 2020).
36. Rubinov, K.; Rosculete, L.; Mitra, T.; Roychoudhury, A. Automated partitioning of android applications for trusted execution environments. In Proceedings of the 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), Austin, TX, USA, 14–22 May 2016; pp. 923–934.
37. Leignac, P.; Potin, O.; Rigaud, J.B.; Dutertre, J.M.; Pontié, S. Comparison of side-channel leakage on Rich and Trusted Execution Environments. In Proceedings of the Sixth Workshop on Cryptography and Security in Computing Systems, Valencia, Spain, 21 January 2019; pp. 19–22.
38. Hay, R. fastboot oem vuln: Android Bootloader Vulnerabilities in Vendor Customizations. In Proceedings of the 11th USENIX Workshop on Offensive Technologies (WOOT 17), Vancouver, BC, USA, 16–18 August 2017.
39. Khan, A.; Mansuri, Z.H. Comparative study of various digital forensics logical acquisition tools for Android smartphone's internal memory: A case study of Samsung Galaxy S5 and S6. *Int. J. Adv. Res. Comput. Sci.* **2018**, *9*, 357–369. [CrossRef]
40. National Institute of Standards and Technology Test Results for Binary Image JTAG, Chip-Off Decoding and Analysis Tool Paraben's Electronic Evidence Examiner. Available online: https://www.dhs.gov/publication/st-binary-image-jtag-chip-decoding-and-analysis-tool-paraben-s-electronic-evidence (accessed on 28 February 2020).
41. MJ—Department of Justice Federal Police. CGTI—General Coordination of Information Technology—IPED. Available online: https://servicos.dpf.gov.br/ferramentas/IPED/ (accessed on 28 February 2020).
42. The Department of Homeland Security. National Institute of Standards and Technology (NIST) Computer Forensic Tool Testing (CFTT) Reports. Available online: https://www.dhs.gov/science-and-technology/nist-cftt-reports (accessed on 3 March 2020)

43. Li, Z.; Xi, B.; Wu, S. Digital forensics and analysis for Android devices. In Proceedings of the 2016 11th International Conference on Computer Science & Education (ICCSE), Nagoya, Japan, 23–25 August 2016; pp. 496–500.
44. Tian, D.; Hernandez, G.; Choi, J.I.; Frost, V.; Ruales, C.; Traynor, P.; Vijayakumar, H.; Harrison, L.; Rahmati, A.; Grace, M.; et al. Attention spanned: Comprehensive vulnerability analysis of AT commands within the android ecosystem. In Proceedings of the 27th USENIX Conference on Security Symposium, Baltimore, MD, USA, 15–17 August 2018; pp. 273–290.
45. Sathe, S.C.; Dongre, D.N. Data Acquisition Techniques in Mobile Forensics. In Proceedings of the Second International Conference on Inventive Systems and Control, Coimbatore, India, 19–20 January 2018; pp. 280–286.
46. Chanajitt, R.; Viriyasitavat, W.; Choo, K.K.R. Forensic analysis and security assessment of Android m-banking apps. *Aust. J. Forensic Sci.* **2018**, *50*, 3–19. [CrossRef]
47. NIST. NIST Tests Forensic Methods for Getting Data From Damaged Mobile Phones. Available online: https://www.nist.gov/news-events/news/2020/01/nist-tests-forensic-methods-getting-data-damaged-mobile-phones (accessed on 28 February 2020).
48. Soares, A.M.M.; de Sousa, R.T., Jr. A Technique for Extraction and Analysis of Application Heap Objects within Android Runtime (ART). In Proceedings of the International Conference on Information Systems Security and Privacy, Porto, Portugal, 19–21 February 2017; pp. 147–156.
49. Soares, A.M.M.; de Sousa, R.T., Jr. Forensic Analysis of Android Runtime (ART) Application Heap Objects in Emulated and Real Devices. In *Information Systems Security and Privacy, ICISSP 2017, Communications in Computer and Information Science*; Mori, P., Furnell, S., Camp, O., Eds.; Springer: Cham, Switzerland, 2017; Volume 867, pp. 130–147.
50. Celebrite—Home. Available online: https://www.cellebrite.com/en/home/ (accessed on 22 January 2020).
51. Cellebrite. Cellebrite Advanced Services Brazil. Available online: https://www.cellebrite.com/en/advanced-services/cas-latam/ (accessed on 22 January 2020).
52. Cellebrite. Serviços Avançados da Cellebrite. Available online: https://www.cellebrite.com/pt/cas-sales-inquiry-pt/ (accessed on 22 January 2020).
53. Grayshift. Introducing GrayKey. Available online: https://graykey.grayshift.com/ (accessed on 22 January 2020).
54. Skulkin, O.; Tindall, D.; Tamma, R. *Learning Android Forensics: Analyze Android Devices with the Latest Forensic Tools and Techniques*, 2nd ed.; Packt Publishing: Birmingham, UK, 2018.
55. Halabtech Support—Home. Available online: https://support.halabtech.com/ (accessed on 22 January 2020).
56. Android Open Source Project. Partições e Imagens. Available online: https://source.android.com/devices/bootloader/partitions-images (accessed on 20 January 2020).
57. Bair, J. *Seeking the Truth from Mobile Evidence: Basic Fundamentals, Intermediate and Advanced Overview of Current Mobile Forensic Investigations*; Academic Press: Cambridge, MA, USA, 2017.
58. Counterpoint. iPhone XR Was the Top-Selling Model Globally in Q3 2019. Available online: https://www.counterpointresearch.com/iphone-xr-top-selling-model-globally-q3-2019/ (accessed on 2 January 2020).
59. StatCounter. Mobile & Tablet Android Version Market Share Worldwide. Available online: https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide (accessed on 19 January 2020).
60. Cellebrite. Exclusive Access to Untouched Evidence in Samsung Exynos Devices. Available online: https://www.cellebrite.com/en/productupdates/exclusive-access-to-untouched-evidence-in-samsung-exynos-devices/ (accessed on 22 September 2019).

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).