Impact of Android Phone Rooting on User Data Integrity in Mobile Forensics

Tahani Almehmadi[1]
Technical College for Girls in Jeddah
Technical and Vocational Training Corporation
Jeddah, KSA

Omar Batarfi[2]
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, KSA

***Abstract*—Modern cellular phones are potent computing devices, and their capabilities are constantly progressing. The Android operating system (OS) is widely used, and the number of accessible apps for Android OS phones is unprecedented. The increasing capabilities of these phones imply that they have distinctive software, memory designs, and storage mechanisms. Furthermore, they are increasingly being used to commit crimes at an alarming rate. This aspect has heightened the need for digital mobile forensics. Because of the rich user data they store, they may be relevant in forensic investigations, and the data must be extracted. However, as this study will show, most of the available tools for mobile forensics rely greatly on rooted (Android) devices to extract data. Rooting, as some of the selected papers in this research will show, poses a key challenge for forensic analysts: user data integrity. Rooting per se, as will be seen, is disadvantageous. It is possible for forensic analysts to extract useful data from Android phones via rooting, but the user data integrity during data acquisition from Android devices is a prime concern. In suggesting an alternative rooting technique for data acquisition from an Android handset, this paper determines whether rooting is forensically sound. This is particularly due to the device's modification, which a root often requires, that may violate the data integrity.**

***Keywords*—Android; rooting; data integrity; mobile forensics**

I. Introduction

It is scarcely fitting to refer to the device that many people use while receiving the occasional call as a telephone currently. This device's capabilities are growing by no less than the number of mobile subscribers using them. For instance, as of October 2012, about one-third of the US populace (121 million subscribers) had a smartphone [1]. These modern mobile handsets not only match low-priced computers in regards to computing capacity but can also store and generate sizeable quantities of data. Due to the devices' computing capacities (and hardware attributes), the gamut of download-accessible usages and the array of tasks that they can accomplish is astounding. These usages/apps are capable of storing data locally in the modern handset [2]. Among these modern (mobile) handsets, the Android OS has recently become the preferred OS [3]. The mobile devices' capacities promote a rapid uptake in consumer and business settings, and Android's open-source nature thus enables scientific research and "reproducibility" [[3], p. 1937].

The increasing prevalence of smartphones has, however, not been without negative consequences. Smartphones have been (and are being) increasingly used in crimes. These devices have been located at crime scenes in the course of investigations. Criminals have used smartphones to commit email fraud, harass others via texts (SMS), for child trafficking and child pornography, and in narcotics-related communications [4]. They have also become shrewd enough to wipe all traces of their activity. This trend has heightened the necessity of digital smartphone forensics, with Android OS-based devices being no exception. To justify this, the data deposited in smartphones can be very valuable to experts during investigations. Smartphones have already proven themselves to carry a sizeable quantity of probative data that is linked to their users based solely on phonebook contacts, SMS and call histories, instant messaging logs, email threads, and browser history. It is probable that these phones have more probative data that can be traced to a user per byte than the majority of PCs, and the acquisition of these data is harder via forensically appropriate methods. This problem is partly due to the overabundance of cell handsets that are currently available. It is worth noting a large number of Android-based phones, the numerous features they possess, the numerous apps specific to them, and, similarly, the valuable data that can be acquired from local storage. There were approximately 1.4 billion in-use Android phones globally as of September 2015 [2]. Coupled with this overabundance are the general scarcity of hardware and software, and the (deficient) standardization of interfaces in the industry. The multiplicity of Android smartphones implies a variation in the models' features, ranging from the media for data storage, the file system, the OS version and the efficacy of some tools. Even separate Android smartphone models produced by the same maker may require separate data cables and software to access the phone data.

Furthermore, the fact that criminals can wipe their activity off of their smartphone's memory, thereby making it difficult for law-enforcement experts to retrieve data from the devices, has become an investigative challenge [5]. It could be that the existing criminal investigation techniques are still immature. It has already been noted that digital smartphone forensics tools are necessary for investigations since the quality collection and analysis of mobile device data depends on them. However, forensic data extraction methods do not usually validate alterations to subscriber data. The forensic acquisition of data is, to a considerable extent, an "invasive" activity because, typically, investigators "crack" the phone to obtain the needed data. This is often done minus the device owner's consent. As such, cracking the device without exposing the integrity of the needed data is a complicated endeavor. This study focuses on the aspect of user data integrity by exploring

whether "rooting" an Android device which is the gaining of administrative privileges before data extraction from Android devices, threatens the user data integrity. The focus on Android devices is due to that Operating System become dominant.

II. Background and Related Work

Although the problem of forensic data acquisition is not new, the majority of expert-designed forensics tools were created out of necessity, and their focus was singularly on the Microsoft Windows OS (a platform that dominated the market for the past 20 years) [3]. Conversely, the cell phones' (factually) comparatively small market share and the differences in (their) hardware and software specifications have hampered the creation of similar tools for cell phones. Smartphones' enhanced capabilities, in comparison to conventional "feature phones," are more intricate. Mobile devices today have features similar to those of computer systems. Android and iOS, the current dominant platforms for smartphones, are built on modern, hardy OSs (Linux for Android and OSX/FreeBSD for iOS) [3]. Even so, these devices' hardware and software are different from those of Windows PCs, for which the present forensics tools and processes are intended. Smartphones, for instance, have no modular hardware (hard drives and detachable RAM cards) that typify modern PCs. Cellular phones may incorporate removable SD memory expansion modules, which can easily be examined via methods similar to those executed on conventional PC systems, but they only serve as auxiliary storage modules. Plus, "many manufacturers are moving away from their use" [[3], p. 1937]. Likewise, cellular phones often run "exotic" file systems and deploy different low-level protocols for accessing data storage modules "that make better use of the embedded non-volatile memory" [[3], p. 1937]. These inbuilt distinctions weaken proper criminal investigations involving cellular phones by using existing tools; thus, novel tools are needed to effectively deal with the new challenges being posed by modern cell phones.

Scrivens and Lin [2] identified the critical elements in forensic investigations on mobile devices, viz. the location(s) for data storage, data mining, and data analytics. The investigator must specifically know where the data are deposited, how the data are deposited, and any attendant file permissions before attempting an extraction. Once these particulars are identified, data extraction must be done since it is an essential part of forensic investigations. Extraction is so critical that using a wrong technique may mess up an investigation. According to Vidas, Zhang, and Christin [6], the prevalence of Android OS devices facilitates the usage of shared attributes to reduce the variety (which digital forensics tools should have) while simultaneously exploiting the capacity for sound data extraction. Makers and network providers tend to maintain competitive advantages by including bonus features in and offering support services to mobile handsets. However, Android handsets have a common framework that is used during acquisition. Specific to Android phones, rooting, in which the investigator or user gains root/administrative privileges where s/he is supposed only to gain unprivileged access, usually involves taking advantage of a security flaw (which is typically dependent on the device and the firmware version) with the intent being installation of unsupported software in the phone. The reasons for rooting Android devices are varied and include the ideological want by users to have control, bypassing controls that are specific to carriers that inhibit the use of particular software, and firmware upgrades (installing an Android version that is higher than that currently supported by the carrier). Rooting, as Grover [1] contends, essentially enables the user to implement elevated-privilege functions on the handset that are usually unavailable in regular user modes. It may be used legitimately or illegitimately. The user may desire to circumvent security controls or to interfere with the data collected via security apps. Overall, rooting can consequently undermine the phone's operating system's security, alter parts of the phone that may collect users' data, diminish interoperability and endanger the device provider's warranty.

Nevertheless, despite the apparent compromises to user data integrity, root access may be inevitable when forensic investigators legitimately deploy it for data extraction. This is contingent upon the situation and the needed data. Whenever possible, root access ought to be avoided.

A. Related Work

Android phones are usually made up of some partitions that are usually mapped to Memory Technology Device (MTD)-type devices. The exact partitioning scheme is dependent on the vendor configuration, but generally, Android phones typically have six partitions. The most common partitions are the /system, /user data, /cache, /boot, and /recovery [5], [6]. The /user data partition is the most forensically pertinent because all the data generated from apps installed by the user is deposited in this partition. As such, wiping it out is like performing a factory reset. It is from the /user data partition where evidence files are often acquired. Alternatively, the /recovery partition, which is "the alternative "system" partition" [[5], p. 288], can be exploited when the system booting fails or when the custom ROM has to be flashed. Forensic investigators use this partition when acquiring a system partition image. Notably, in normal mode, no application data is deposited in the /recovery partition; therefore, data corruption or overwriting there has no likelihood of altering data on the phone that may subsequently be used in a criminal case.

Acquiring data from Android phones is generally categorized into physical and logical acquisition techniques [7]. Logical acquisition methods (in which the focus of this study lies) include file/folder copying, Content Providers, and Recovery Mode [7], whereas physical acquisition techniques involve data partition imaging. Son et al. [7] focused on the Recovery Mode. In determining whether the Android Recovery Mode maintains the integrity of the user data during its acquisition, the authors justified that the Mode can grant administrator access while the phone is in a state where the corruption of the user data can be reduced. Conversely, for (the) imaging of the data partition containing the user data and/or copying files/folders, the phone must be rooted first. In this case, the phone must be booted normally. Normal booting, as Son et al. argue, may not ensure the integrity of the user data or that of unallocated data. Therefore, the authors detailed a process intended to lessen the time and extra work required for the forensic investigation of a suspect Android phone. From the procedure, they developed a tool (Android Extractor) to automatically execute the process via a series of experiments

using several Android device models. Their tests confirmed the preservation of the integrity of the user data. Comparatively, the JTAG (Joint Test Action Group), which the authors used for physical data acquisition, was effective in fully acquiring the device data. When the JTAG is used first before the Android Extractor, they concluded (based on the JTAG-compatible devices that were used) that JTAG also maintained user data integrity. However, Hazra and Mateti [5] noted that the JTAG forensics technique of acquiring memory data is executed only when data acquisition via physical or logical extraction is unsuccessful and that it is risky. Although it is useful in extracting locked data, the risk of losing evidence is always there.

In [4], the authors noted that imaging the device's memory is critical in mobile forensics because the memory may contain useful data. Its access can be possible by rooting the device. They detailed a procedure for acquiring all the information from Android Negated AND (NAND) flash(ing). One method suggested the facilitated collection of a byte-by-byte duplicate of the NAND flash per se to recover deleted data. The process required rooting the device to extract a dd image of the appropriate partition(s) and store it in a detachable SD card mounted in the phone, after which the (memory) dumps were examined for prospective evidence. Its disadvantage is that a microSD card slot must be present, which is a deficiency present in many popular Android phone models.

Moreover, extracting a dd image file is likely "when permissions are altered to gain access to the root directory" [[4], p. 3]. As such, rooting is not forensically reliable. Furthermore, root access to obtain the dd image requires the installation of a 3rd-party program in the phone. This would make the acquired data is used as evidence, inadmissible in court. It must be noted that there are other ways to gain administrative privileges on other Android phones that require no 3rd-party software installations. Rooting via 3rd-party installation(s) could be customized to be forensically sound if alternative ways of gaining root privileges are found.

In [6], the authors outline a process for acquiring the logical and physical images of phone storage via the custom recovery image (CRI) technique, and its focus is on Android phones' /recovery partition and the Android Recovery Mode. It requires altering the /recovery partition. Nevertheless, as discussed earlier, the /user data partition is the partition of interest since much of the data that forensic analysts are interested in is found there. As such, the alteration of the /recovery partition will not affect the data. Its operational outline is as follows: (i) acquire a CRI that incorporates the special utilities that facilitate the recovery of the data, ADB, and superuser; (ii) flash the CRI to the Android phone; (iii) reboot the phone in /recovery mode; and (iv) use the command "ADB shell" from the forensic computer terminal "to execute data recovery binaries from the recovery image" [[2], p. 5]. Some data dumping utilities may be utilized, which are contingent upon the flash storage technology in use. Many Android phones use MTD [5]. The Media Technology Device system is an extraction layer for raw (NAND) flash phones that grants software permission to use one interface in accessing multiple flash technologies or a device driver used for directly accessing NAND flash storage. The nanddump for MTD phones may be executed to acquire "NAND data independent of the higher-level filesystem deployed on the memory" [[6], S17]. For phones with no MTD mechanism, other acquisition methods must be used. The dd utility, for instance, may be utilized for copying data. Both of these utilities may be deployed in the recovery of a physical image. Additionally, it is worth noting that not all files are necessarily warehoused in the onboard memory since many Android phones support one microSD module. While the user can install particular apps and store specific data on their phones, some makers may opt to install the /user data partition in its entirety on the module.

The work of Son et al. [7] continues that of [6], although their focus is on the issue of data integrity. After the creation of the custom recovery mode image, the phone must be booted in the flash mode for the image to be flashed to /recovery (or /boot). Here, Son et al. emphasize a crucial aspect associated with the data integrity. If the image is flashed to the /recovery space, the phone ought to shift to Recovery Mode after being flashed. However, the phone "must be manually entered into Recovery Mode" [[7], S7]. In the case that booting into Recovery Mode does not work, the phone will go on to boot normally, thus using the /user data partition and possibly compromising the integrity of user data.

Conversely, in the case that the image is flashed to the /boot partition, the phone may subsequently, instantly and automatically go into recovery mode. With root permission in this mode, forensic investigators can obtain device access via the use of the Android Debug Bridge (ADB) command. From here, investigators can acquire all the needed data. This mode was the basis of the Extractor that was developed and deployed [[7], S8]. Based on the two primary data acquisition methods outlined earlier (data partition imaging and file copying for emphasis), mounting the partition to acquire the (targeted) partitioned unit is unnecessary. Nevertheless, for the file unit to be acquired, the /user data partition ought to be mounted in read-only mode. In this way, data acquisition can be made via the ADB pull command and, more importantly, data integrity is guaranteed.

III. Research Motivation

Data extraction from smartphones during a forensic investigation poses a number of challenges for forensic experts. By using the proper techniques and tools, it is possible to mine useful data from call logs, contact lists, SMS and email threads and browser history. However, the integrity of users' data during acquisition is a major issue for forensic analysts. This need is what has prompted the design of this study, its specific focus on rooting, and the data integrity concerns that have been posted. Therefore, we seek to compare user data integrity when an Android phone is rooted with data extracted from the phone via a custom recovery image, which is believed to affect only the recovery partition without the user data partition. In addition, we compare them with the basic data extracted from the phone before rooting.

1) Hypothesis: If versatile, high-reliability rooting software is used on an Android phone and user data is extracted using forensic software, all the data can be acquired without changing its integrity. These data can thus be used as reliable evidence during forensic investigations.

IV. Experiments

The provision of a proper environment for performing the (intended) experiment is crucial to ascertain that the findings drawn from it are correct. The data acquisition tools are detailed below. Table.I for hardware tools and Table.II for software tools. It should be noted that all the programs used in the experiment's implementation are licensed.

TABLE I. Hardware Tools

Hardware	Specification
Dell Inspiron 15 7000	Intel Core i7, 2.80 GHz, 16 GB
Samsung Galaxy S4	GT-I19505
USB Cable	Micro USB Data Charger Cable
MicroSD card	64 GB

TABLE II. Software Tools

Software	Specification
Microsoft Windows10	64-bit
SAMSUNG USB Driver for Mobile Phones	Driver definitions to connect to the computer
Android Debug Bridge (ADB)	Access the mobile data on the computer
KingoRoot [8]	PC Version
Odin v3.09	A utility developed by Samsung to flash a custom recovery image to a Samsung Android device
TWRP recovery image [9]	Custom Recovery Image (CRI)
Belkasoft Evidence Center v9.2 - Trial version [10]	It analyzes digital evidence stored in computers and mobile devices
FileAlyzer v2.0	Tool to analyze files

A. Data Acquisition

The experiment will use ADB commands, the custom recovery image, and rooting techniques for data acquisition. A comparison will then be made to determine the effect of Android device rooting on user data integrity. The first step is shown in Fig.1. In detail, a backup was taken from an Android phone using ADB before any rooting operations were performed on the device. ADB is one of the command line tools that constitute the Android SDK package. It allows communication with Android devices and performs actions such as app installation and debugging and aids the safe backup of device and app data on PCs, regardless of the OS. Thus, after enabling developer options and connecting the Android phone to a PC, we ran the command-line interface to make a backup using ADB commands.

For the 2nd stage, a custom recovery image (CRI) was used in data acquisition. The last acquisition method focused on modifying the recovery partition. However, the important content is in the /user data partition, and so modifying the /recovery partition will not affect these data. The data can be acquired from the partition via the ADB pull command or by using the copy process to the MicroSD card from the TWRP homepage. We used the process of copying to the MicroSD card to interface with the smartphone while in recovery mode and extract all files and folders. In the 3rd and last stage, the researchers rooted the device using KingoRoot. KingoRoot Android works on Windows. It supports almost any Android device and version, is risk-free and can unroot at any time. After successfully rooting the Android phone, we used the Belkasoft Evidence Center backup that based on a

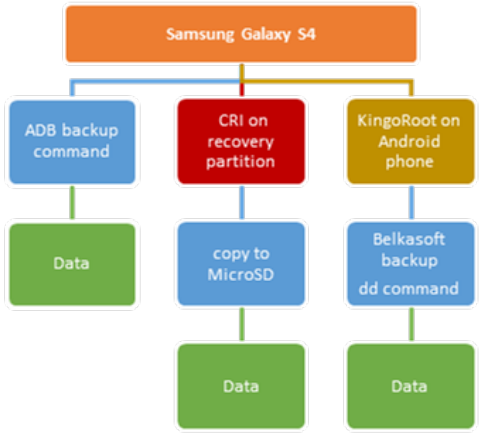

Fig. 1. Step 1 in data acquisition

dd command to gather data. As shown in Table.III the backup file characteristics and their corresponding hash value.

TABLE III. The Backup File Characteristics and their Corresponding Hash Value

ADB Backup GalaxyOriginalAndroid.ab - 4.45 GB 7BB2BA975D0E69E1CEFE5CCE2965CC1726597525
CRI GalaxyCRMI- 12.6 GB 5609BB28440CB5B20F5C1A25AA750F972BEFAB8A
KingoRoot - Belkasoft Backup GalaxyRootedAndroid.dd - 14.6 GB 0CF0458CB55CADDF495DA8E45A6A9DB8710C3453

B. Data Analysis

The Belkasoft Evidence Center program was used to extract and analyze the digital evidence from the three Android backups. The Images and Memos files were analyzed by selecting a random file from the extracted folder in the first phase and comparing the hash value of the file with the corresponding file extracted in the second and third phases of the experiment.

Images and Memo Files: The sample file (1470160927734.jpg) was extracted from the Images folder and analyzed using the Belkasoft Evidence Center as shown in Fig.2.

From the sample file extracted from the Images folder in the three acquisition states that were executed, it can be seen that the image's name, shape, identity, and actual path are retained. It can also be noted from the FileAlyzer report that the examined Memo also has the same hash values as shown in Fig.3.

C. Main Points of the Analysis

The results of the illustrated analyses indicate that no data changes occurred during the rooting process or during data extraction. This result is consistent with the results achieved recently, despite the different experiences and programs used [11]. Nevertheless, the results of the folder analysis show an apparent discrepancy in the amount of data that was retrieved using the Belkasoft Evidence Center. The reason for the

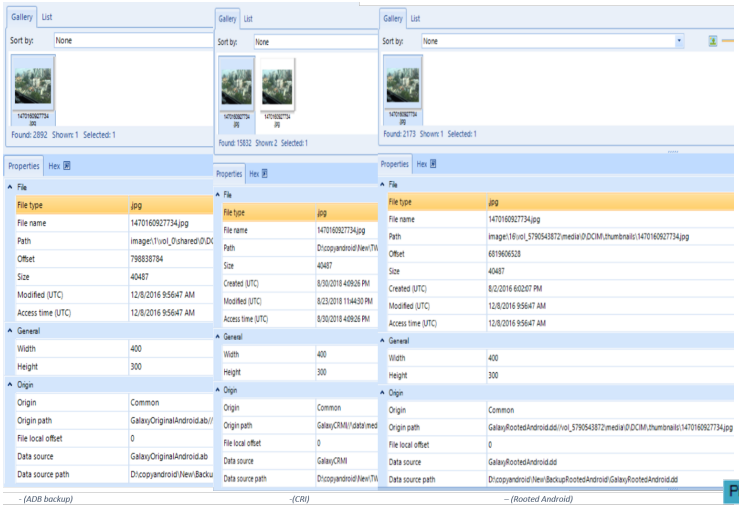

Fig. 2. Analyze sample file (1470160927734.jpg)

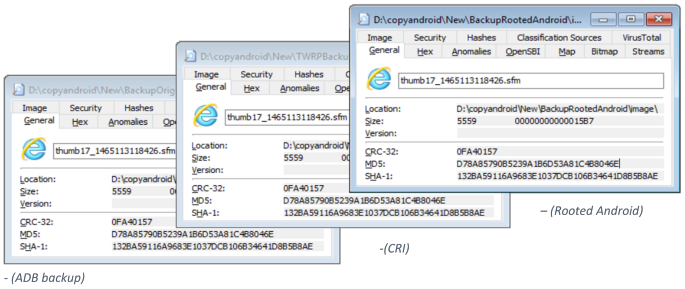

Fig. 3. Analyze sample file (thumb17_1465113118426.sfm)

different amounts of data goes back to the repeated files, where we notice the height of the images in the backup using CRI Fig.5, while for the backup using rooting Fig.6, we see a high number of documents. While this amount of data is not shown in the backup using ADB as shown in Fig.4. It is also worth noting that the tools that are used to install the root is 3rd-party utilities on the Android device. Nevertheless, the utilities did not affect the final data that was recovered.

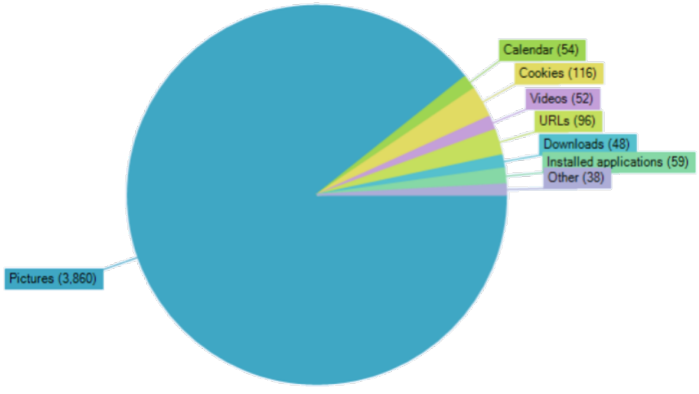

Fig. 4. Samsung Galaxy S4 – ADB Backup

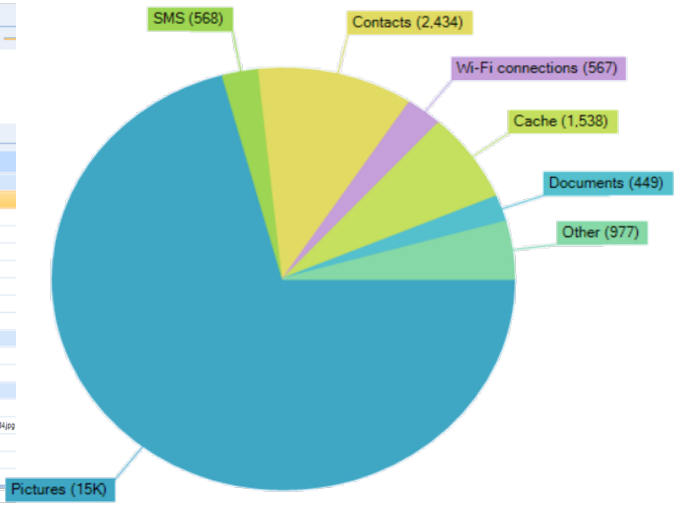

Fig. 5. Samsung Galaxy S4 – CRI

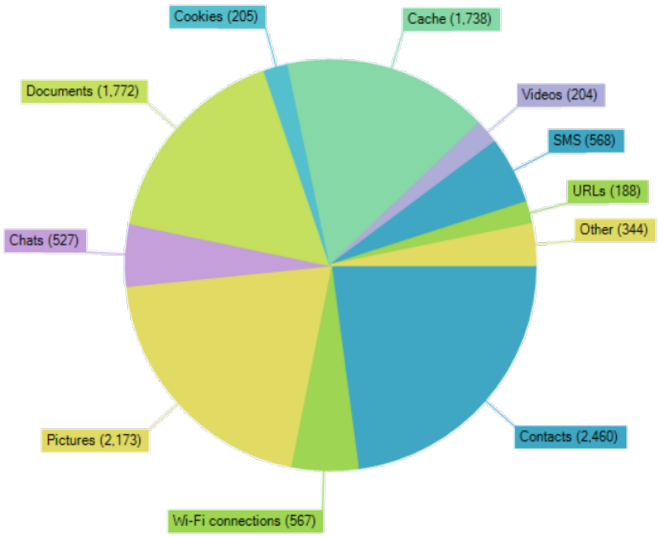

Fig. 6. Samsung Galaxy S4 – KingoRoot - Belkasoft Backup

V. Conclusion

The use of Android devices around the world is growing exponentially. Unfortunately, this rapid growth has led to the misuse of these devices. Similarly, smartphones are now important in criminal investigations. The data stored in different applications in smartphones can be used by forensic experts during the investigation of a crime. There are different tools and methods used to get and extract data from Android smartphones.

This paper sought to investigate the impact of rooting Android phones on the integrity of user data and the search for any damage resulting from the rooting of the device since Android device rooting to acquire physical data necessitates modifications to the device data. Herein, we did not notice any effect on the user data during the process of rooting. Believe it is preferable to document the processes and events during the extraction process and to avoid unnecessary changes to the user data.

The rooting process is therefore legally valid. In addition, the evidence extracted from android devices as a result of the rooting process is sound, reliable evidence of sentencing in criminal cases.

ACKNOWLEDGMENT

We thank Belkasoft LLC who provided support that greatly assisted the research.

REFERENCES

[1] J. Grover, "Android forensics: Automated data collection and reporting from a mobile device," Digital Investigation, vol. 10, pp. S12–S20, 2013.

[2] N. Scrivens and X. Lin, "Android digital forensics: data, extraction and analysis," in Proceedings of the ACM Turing 50th Celebration Conference-China, 2017, p. 26.

[3] D. Votipka, T. Vidas, and N. Christin, "Passe-partout: A general collection methodology for Android devices," IEEE Transactions on Information Forensics and Security, vol. 8, no. 12, pp. 1937–1946, 2013.

[4] J. Lessard and G. Kessler, "Android Forensics: Simplifying Cell Phone Examinations.," 2010.

[5] S. Hazra and P. Mateti, "Challenges in Android Forensics," in International Symposium on Security in Computing and Communication, 2017, pp. 286–299.

[6] T. Vidas, C. Zhang, and N. Christin, "Toward a general collection methodology for Android devices," digital investigation, vol. 8, pp. S14–S24, 2011.

[7] N. Son, Y. Lee, D. Kim, J. I. James, S. Lee, and K. Lee, "A study of user data integrity during acquisition of Android devices," Digital Investigation, vol. 10, pp. S3–S11, 2013.

[8] "KingoRoot for Android, the best One Click Root Tool/APK for free." [Online]. Available: https://www.kingoapp.com/. [Accessed: 22-Dec-2018].

[9] "Download TWRP for jfltexx." [Online]. Available: https://dl.twrp.me/jfltexx/. [Accessed: 22-Dec-2018].

[10] "Belkasoft: Evidence Search and Analysis Software for Digital Forensic Investigations." [Online]. Available: https://belkasoft.com/. [Accessed: 22-Dec-2018].

[11] M. Hassan and L. Pantaleon, "An investigation into the impact of rooting android device on user data integrity," in Emerging Security Technologies (EST), 2017 Seventh International Conference on, 2017, pp. 32–37.