Skype Forensics in Android Devices

Mohammed I. Al-Saleh
Jordan University of Science and Technology
Computer Science Department
P.O. Box 3030
Irbid, Jordan, 22110

Yahya A. Forihat
Jordan University of Science and Technology
Computer Science Department
P.O. Box 3030
Irbid, Jordan, 2211

ABSTRACT
The discipline of smartphone forensics has recently got more at- tention because of the tremendous growth in the smartphones market. Smartphones, to some extent, have similar capabilities to that of PCs. They can store large amount of data and diver- gent categories of information. Among other mobile platforms, Android-based devices are getting more popularity. Variety of mo- bile Applications (Apps) are increasingly developed to mainly extend the functionally of the phones. The usage of Voice over IP (VoIP) Apps has explosively increased for their wide avail- ability and cheap prices. As Skype is one of the most popular VoIP Apps, in this paper we investigate the artifacts of Skype calls and chats in the Android devices. We inspect both the RAM and NAND flash memories in different scenarios and time dura- tions. Even though Skype provides secure communications over the Internet, this paper shows that Skype call and chat evidences can be truly found in the devices. To the best of our knowl- edge, we are the first to investigate Skype in the Android devices.

General Terms
Digital Forensics, Cyber Security

Keywords
Skype, Android, RAM Artifacts, NAND Artifacts, Digital Evi- dence.

1. INTRODUCTION
Digital Forensics (DF) techniques are utilized by analysts and law enforcement agencies to detect the activities of the cyber criminals and prove them guilty under the law. Cyber crimes include break- ing into others machines, executing unauthorized code, destroy- ing others information, deceiving innocent people, and distribut- ing child pornography. This paper assumes that Skype, in a way or another, might help in committing some of the above mentioned crimes. DF, besides the other security countermeasures, plays a ma- jor role against cyber crimes. DF may be conducted on computers, mobile devices, and networks. The main DF procedure involves seizing the targeted digital device, preserving and acquiring the device contents, analyzing the contents, and extracting evidences. However, the diversity of the modern hardware architectures and software components makes the forensics analysis process compli- cated, challenging, and ever-evolving.

Smartphones are remarkably increasing throughout the world for their useful features and affordable prices. Furthermore, the com- petition between the smartphones manufacturers and mobile car- riers creates variety of options for customers to choose the most suitable smartphones in terms of features and prices. According to a study [17], by the end of 2012, 31% of the US mobile users will be using smartphones.

Smartphones capabilities are notably increasing. They are cur- rently equipped with multi-core processors, high-capacity memo- ries, high-resolution cameras, WiFi connection, and GPS facilities. As a result, smartphones forensics becomes necessary for the capa- bilities they empower and the possibility that cyber criminals abuse them to commit crimes.

Android (firstly produced in October 2008) is one of the mostly used platforms in smartphones. Because Android is an open-source system and it is based on the Linux kernel, it is given more attention from the mobile phones carriers, manufacturers, as well as Apps de- velopers. Consequently, Android dramatically influences the smart- phones market. It captures 26% of the US smartphones subscribers and 25.5% worldwide [7]. According to the official Android web- site (www.android.com) and as of this writing, over 850,000 An- droid devices are activated every day with 55 manufacturers and over 300 carriers.

One of the most astonishing features of the smartphones is that their ability to install Applications (Apps) for the purpose of extending their functionalities. These Apps utilize the phone features and ca- pabilities to provide something new and useful to the users. Smart- phones developers provide variety of Apps that use the VoIP tech- nology to make free (or cheap) calls. Skype is a well-known VoIP App that is mainly used to make calls and create chat sessions. Given the importance of both the Android platform and the Skype App and the possibility of abusing them by criminals to commit cy- ber crimes, this paper studies the Skype calls and chats artifacts on Android systems for forensics purposes. Even though Skype pro- vides secure communication over the Internet [13, 9],we show that multiple copies of call patterns and chat messages can be found in both of the RAM and flash memories of the Android devices for a long time, even after deleting histories and signing out of Skype. We think that the results shown in this paper will have a major impact on Android and VoIP forensics.

This paper is organized as follows. First, we give a background and our investigation model. This is followed by the Experiments section that explains our experiments, and then our results are shown. A discussion and future work follow. Finally, related work and the conclusion are shown..

2. BACKGROUND AND INVESTIGATION MODEL

2.1 Android Architecture
Android is an open source Operating System (OS) developed by Google for mobile devices. It is based on Linux 2.6 kernel

[7]. The Android architecture is shown in Figure 1. The Linux Kernel part contains a set of drivers such as Audio Driver, Power Man- agement, and Wi-Fi Driver. Libraries offer the core functionalities required by developers and users. Google introduced its own Vir- tual Machine (VM) that is called Dalvik Virtual Machine (DVM). DVM allows applications to run concurrently, each having a sepa- rate VM. Therefore, files created by one Android application can- not be viewed by others unless they are explicitly shared. The Ap- plication Framework provides the API interface through which de- velopers can interact with the system. The last layer contains the device applications.

2.2 Random Access Memory (RAM) and NAND Flash

The memory in Android devices is divided into two primary types: the Random Access Memory (RAM) and the NAND flash memory. Operations and data need to go through the RAM to get manipu- lated. The RAM is volatile, which means that it does not preserve its state without power. However, it can hold important information such as encryption keys, usernames, passwords, and applications data [7]. On the other hand, the NAND flash memory is nonvolatile. This means that if the device is powered off or rebooted, the data in the NAND flash are preserved. The NAND flash memory is used to store both system and user data. The compact size, high speed, and low-power consumption of the NAND flash make it a preferable storage medium.

2.3 Investigation Model

Figure 2 summarizes our investigation model. Both the criminal and the victim have to be signed into the Skype. The criminal has an Android device while there are no assumptions about the victim's machine. The criminal starts call and chat sessions with the victim. Investigators need to extract evidences about the call and chat ses- sions from the criminal's device by inspecting both the RAM and NAND flash memories.

3. EXPERIMENTS

Figure 3 shows the basic setup for our experiments. Android SDK is installed and integrated with Eclipse 3.7.2 Indigo tool. The Em- ulator is launched with the following specifications:

—Android version: 4.0.3, API level 15

—Processor: ARM (armeabi-v7a), Model cortex-a8

—RAM size: 512 MB

—SD card size: 100 MB

First, we install the Skype App on the emulator. Then, we run and sign into the Skype and start our experiments with the following procedure:

—First, we make 6 consecutive calls from the Skype App in the emulator to another remote machine that has another Skype ac- count. The calls are made with different time durations. The du rations of the 6 calls in seconds are 180, 150, 120, 90, 60, and

30, respectively.

—Second, right after making the above calls, we launch the Skype Instant Messaging (IM). An interactive chat session is initiated. A total of 6 chat messages are sent forth and back between the two Skype accounts (3 messages from each).

—Third, after conducting the second step above, we test against each of the following scenarios separately:

—Scenario 1: stay signed-in all the experiment time (up to 12 hours, see the fourth step).

—Scenario 2: sign out of the Skype right after the second step.

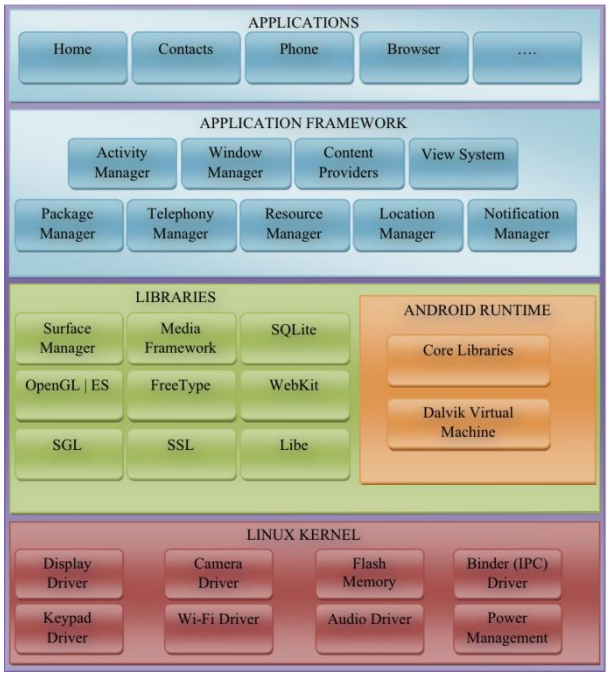

Fig. 1. Android architecture, adapted from [7]

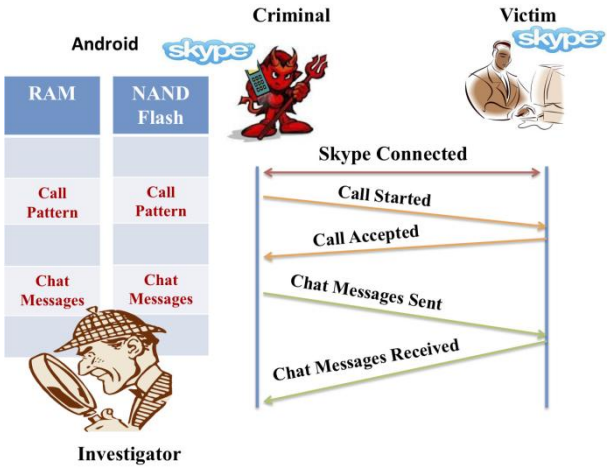

Fig. 2. Investigation model

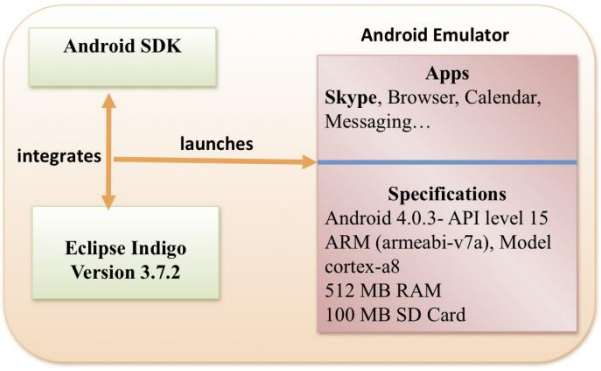

Fig. 3. Basic experimental setup

—Scenario 3: after the second step, we delete the Skype calls and chats histories and then sign out of the Skype.

—Fourth, with one chosen scenario from the third step, the RAM memory and the NAND flash of the emulator are dumped at var- ious times: 0 minutes, one hour, six hours, and 12 hours.

—Fifth, to ensure repeatability, all the above steps are repeated 3 times with a fresh Skype install.

A shell prompt to an Android device is provided by the Android Debug Bridge (ADB) tool. An Android process's memory can be dumped by sending it a special signal that is called SIGUSR1 through an active shell. Here are the steps to extract a process's memory [7]:

—Listing Processes IDs (PIDs):

—adb shell ps

—Running an interactive shell and changing the permission mode of the /data/misc directory:

—adb shell

—chmod 777/data/misc

—Sending SIGUSR1 to a specific PID:

— kill -10 PID

—The previous step causes the process's memory to be dumped into /data/misc/heap-dump-num-pidPID.hprof file. This file needs to be pulled by this command:

—adb pull /data/misc/heap-dump-num-pidPID.hprof .

A process's memory can also be easily dumped using the DDMS (Dalvik Debug Monitor Server) tool.

Each App stores its data in the NAND flash in /data/data/App-Dir directory. The Skype's data can be easily dumped by executing:

—adb pull /data/data/com.skype.raider

—Sixth: to ensure that our results apply to real Android devices, we conduct the experiment described in the First and Second steps on a real Android device. We only apply Scenario 3 from the Third step because it checks against the hardest test (i.e., after deleting histories and signing out). The device we use is Samsung Galaxy mini 2 S6500. The phone has 4GB storage and

512 MB RAM. In order to have access to the device's data, the phone must be rooted (a process to escalate to root privileges and get full access to all phone resources). An ADB connection is needed to communicate with the device. A shell is obtained by executing adb shell. Root privileges are required (by execut- ing su command) to change the permission rights of some files (Skype data files in our case) using the chmod command. The files are then pulled using the adb pull command or with the help of DDMS tool.

```
< part identity "Skype Account Name for Caller">
        <name> "Caller Name" </name>
        <duration> "The Duration of Call in Seconds" </duration>
</part>

<part identity = "Skype Account Name for Recipient">
        <name> "Recipient Name" </name>
        <duration> "The Duration of Call in Seconds" </ duration>
</part>
```

Fig. 4. Call pattern

4. RESULTS

In this section we present the results of the experiments explained in the previous section. We aim at showing that Skype artifacts in the Android systems can be utilized in digital forensics. We search the dumps of both of the RAM and NAND flash memories for the artifacts of Skype calls and chats. To search the RAM and NAND flash for Skype artifacts, we use manual searching, grep tool, and Eclipse Memory Analyzer tool. Interestingly, we find that calls in- formation is found as patterns in both memories. Figure 4 shows the call pattern structure. However, the chat messages are found as plain texts in both memories.

Figures 5 and 6 show the persistency of the calls patterns in both of the RAM and NAND flash memories while applying Scenario

1. The figures show that the calls patterns are truly found in both memories with averaged occurrences ranging between 2 and 3. The figures also show that the persistency of the calls patterns do not change over time; the same numbers of occurrences appear for the different time durations (Zero Min, One Hour, Six Hours, and Twelve Hours). Figures 7 and 8 show the persistency of the calls patterns in both of the RAM and NAND flash memories while ap- plying Scenario 2, while Figures 9 and 10 show the persistency of the calls patterns in both of the RAM and NAND flash memories while applying Scenario 3. The four figures show no difference from the previous figures. Therefore, we can conclude that calls patterns can be truly found in both of the RAM and NAND flash memories regardless of being singed in or signed out the Skype or deleting Skype calls history. Furthermore, the persistency of calls patterns is static over time, which means that calls patterns will stick in both memories for a long time. Finally, repeating the ex- periments three times for each scenario shows consistent results. Figure 11 shows the persistency of the calls patterns in the NAND flash memory of a real Android device while applying Scenario 3. The figure shows that the results for a real Android device have the same conclusions as for the emulator.

Figures 12 and 13 show the persistency of the chat messages in both of the RAM and NAND flash memories while applying Sce- nario 1. The figures show that the chat messages are truly found in both memories. Figure 12 shows that the average number of oc- currences for the chat messages in the RAM memory is above 5 for all messages for the different time durations (Zero Min, One Hour, Six Hours, and Twelve Hours). Even though the average decreases for all messages over time, the remaining number of messages still significant and can be used as a chat evidence. Figure 13 shows that the average number of occurrences for the chat messages in the NAND flash memory is 1 for all messages for all periods. Thismeans that chat messages stick in the flash memory for a long time but without redundancy.

Figures 14 and 15 show the persistency of the chat messages in both of the RAM and NAND flash memories while applying Sce- nario 2, while Figures 16 and 17 show the persistency of the chat messages in both of the RAM and NAND flash memories while ap- plying Scenario 3. Figure 14 shows that signing out the Skype has a little impact on the average number of chat messages occurrences in the RAM memory (compared to Figure 12). Figures 13, 15, and 17 show that applying Scenario 1, Scenario 2, or Scenario 3 has no effect at all on the average number of chat messages occurrences in the NAND flash memory; it is 1 for all scenarios and all time periods. Figure 16 shows that applying

Scenario 3 (deleting chat history and signing out Skype) has also a little impact on the average number of chat messages occurrences in the RAM memory (compared to Figure 12).

Figure 18 shows the persistency of the chat messages in the NAND flash memory of a real Android device while applying Scenario 3. The figure shows that the results for a real Android device have the same conclusions as for the emulator.

Besides Skype calls and chats artifacts, other Skype-related infor- mation can be found in both of the RAM and NAND flash mem- ories. Table 1 shows that the account contacts and E-mails can be found in both memories for most of the scenarios.

In summary, Skype calls patterns and chat messages can be found in both of the RAM and NAND flash memories for a long time and regardless of deleting calls and chat histories and sign- ing out of the Skype.

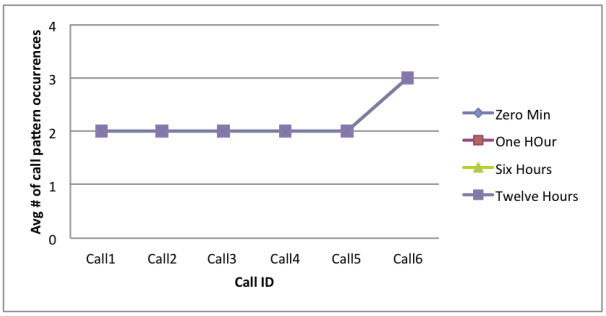

Fig. 5. The average number of occurrences for the call patterns in theRAM while signed in (Scenario 1)

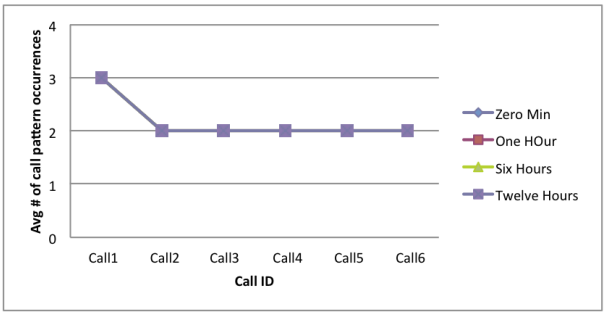

Fig. 6. The average number of occurrences for the call patterns in theNAND flash while signed in (Scenario 1).

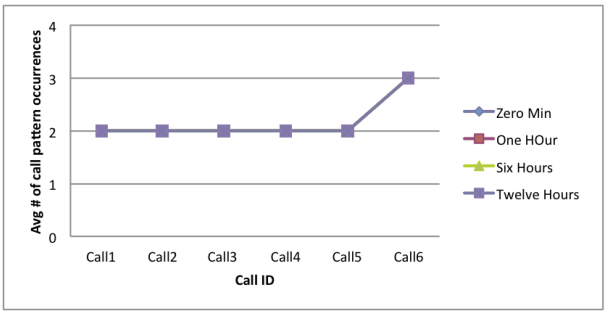

Fig. 7. The average number of occurrences for the call patterns in the RAM after signing out (Scenario 2)

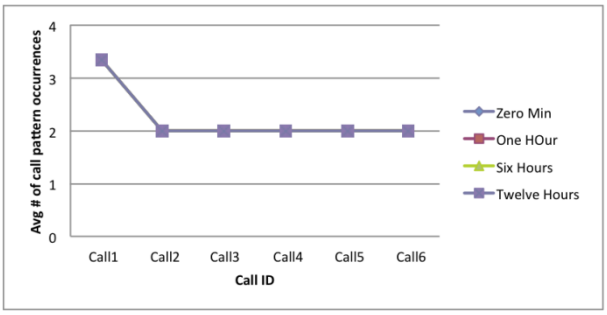

Fig. 8. The average number of occurrences for the call patterns in the NAND flash after signing out (Scenario 2).

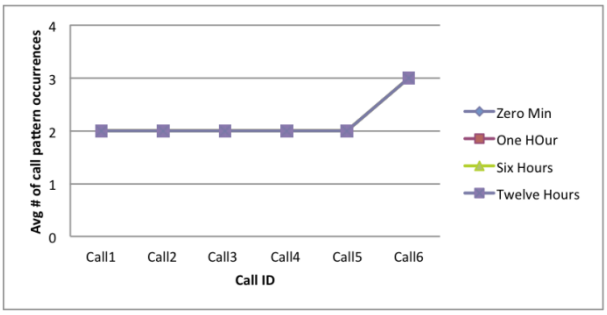

Fig. 9. The average number of occurrences for the call patterns in theRAM after deleting history and signing out (Scenario 3)

Table 1. Contacts and E-mails artifacts

Scenarios	RAM		NAND Flash	
	E-mail	Contacts	E-mail	Contacts
Scenario1	Yes	Yes	Yes	Yes
Scenario2	Yes	Yes	Yes	Yes

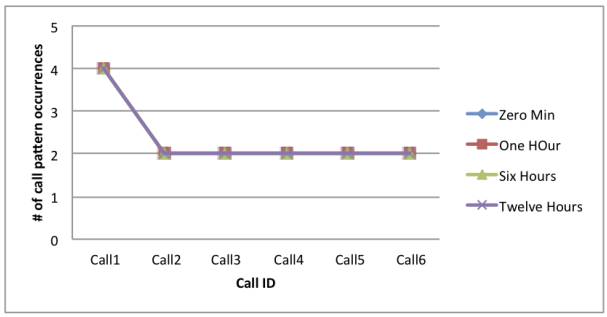

Fig. 11. The number of occurrences for the call patterns in the NAND flash after deleting history and signing out (Scenario 3) in a real Android device

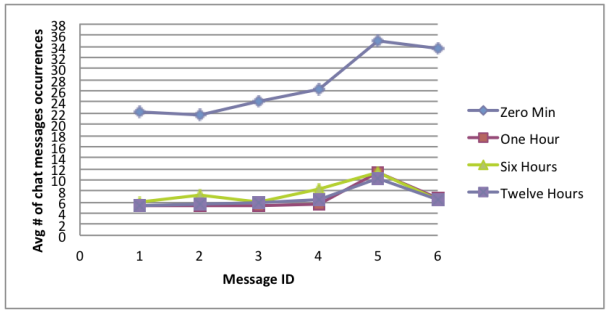

Fig. 12. The average number of occurrences for the chat messages in the RAM while signed in (Scenario 1)

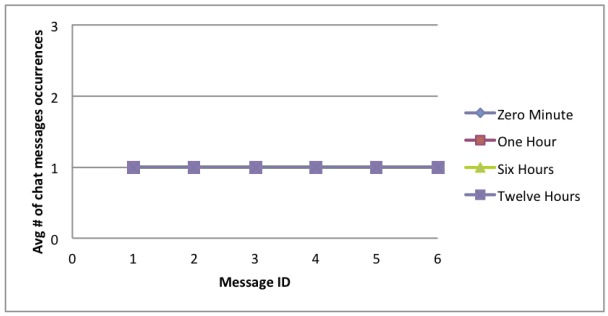

Fig. 13. The average number of occurrences for the chat messages in theNAND flash while signed in (Scenario 1)

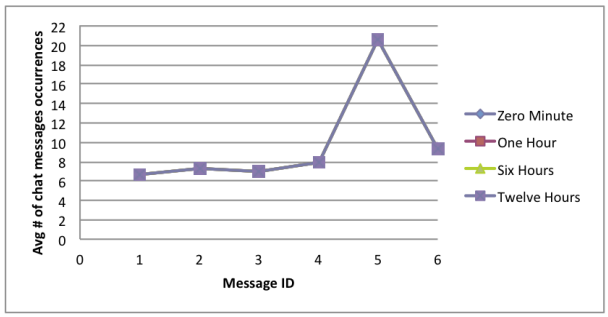

Fig. 14. The average number of occurrences for the chat messages in the RAM after signing out (Scenario 2)

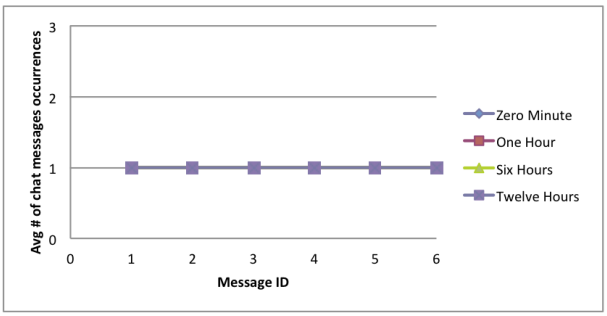

Fig. 15. The average number of occurrences for the chat messages in the NAND flash after signing out (Scenario 2)

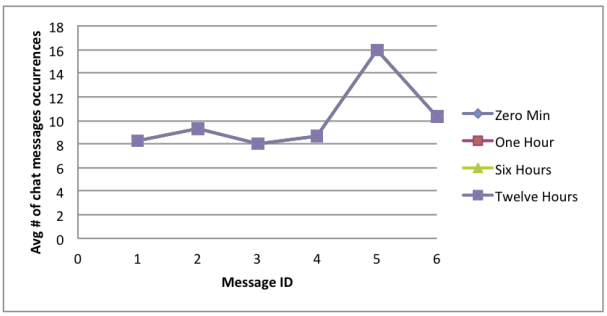

Fig. 16. The average number of occurrences for the chat messages in theRAM after deleting history and signing out (Scenario 3)

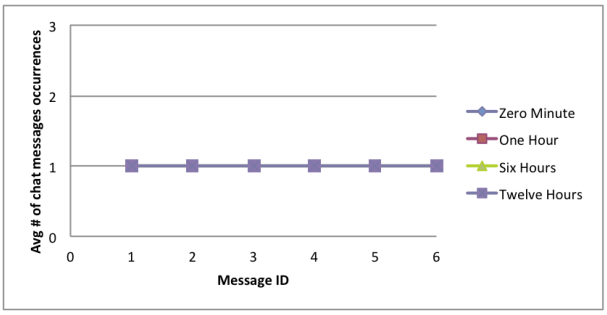

Fig. 17. The average number of occurrences for the chat messages in theNAND flash after deleting history and signing out (Scenario 3)

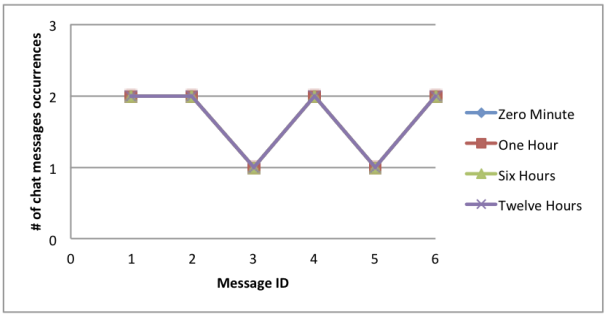

Fig. 18. The number of occurrences for the chat messages in the NAND flash after deleting history and signing out (Scenario 3) in a real Android device

5. DISCUSSION AND FUTURE WORK

Digital Forensics (DF) is challenging because of the wide variety of digital devices, software complexity, and criminals' smartness. There is no standard technique for all DF cases; every case might be different. Forensics analysts need to look for every possible evidence to be able to accuse criminals. This thesis helps foren- sics analysts in that direction. We chose the Android platform in our study for its source-openness and feature-richness. Also, many smartphones manufacturers and carriers utilize or customize An- droid for their own needs. We chose testing the Skype App because it is one of the most popular VoIP applications. In this study, we sought for digital evidences in the volatile (RAM) and non-volatile (NAND flash) memories in order to produce the required evidence. The RAM memory has very valuable information, which makes in- specting its contents very important. However, the RAM's contents will vanish if the smartphone is powered off. In addition, the RAM pages can be reused by the OS if needed. Consequently, some of the contents will be replaced over time or extensive use. The NAND flash, however, keeps the stored information permanently unless it is explicitly deleted.

As a future work, we aim at testing the artifacts of other VoIP Apps (such as Google Talk and Viper) and incorporating other platforms (such as iPhone devices with iOS Operating System). Also, recon- structing/replaying the audio parts of VoIP calls and checking their persistency over time is a future direction.

6. RELATED WORK

Android forensics is an emerging field that attracts researchers and forensics analysts. [19] presents two methods for extracting a physical image from the internal NAND flash memory to uncover deleted files. Challenges associated with mobile forensics are pre- sented by [21]. [20] provides a comprehensive study for popular digital forensics tools.

A full capture for the volatile memory (RAM) for Android devices using a kernel module is proposed [15]. Nine messaging and VoIP applications for smartphones are evaluated by [10], focusing on authentication mechanisms in real-world scenarios. The evaluation shows that the authentication mechanisms are broken for many ap- plications and are vulnerable to the hijacking attacks. [16] presents an automated system for analyzing volatile memory properties and acquiring real-time evidences. The authors focus on analyzing the incoming and outgoing chat messages in the volatile memory. They investigate different scenarios, and show that the outgoing mes- sages have higher persistency in the memory than the incoming messages. The lifetime, sensitivity, and security of the RAM mem- ory contents of the computer systems have been studied by different researchers [1, 8, 14, 11, 18, 12, 3, 6, 5, 4, 2].

Recovering digital evidences from VoIP applications in computer systems has already been studied [13, 12]. [13] show that the phys- ical memory of a machine can be inspected to look for VoIP calls. First, they search the physical memory for the packets that are in- volved in a specific call. Then, they extract the audio payloads from the packets and reconstruct the call. Consequently, they are able to replay the audio. However, the algorithm they use to find the in- volved packets in the memory and reconstruct the audio streams is not open for the public. [12] show that some Skype information can be recovered from the physical memory. For example, they show that Skype passwords can be found in the memory if the Skype is still running. Furthermore, the Skype contacts can be found in the memory. However, the encryption keys cannot be found. The study, however, does not test Skype artifacts against calls or chats. This work is the first to investigate Skype in Android systems by making real calls and chats.

7. CONCLUSION

Smartphones Forensics (SF) is getting more popularity for the tremendous growth in the smartphones market and extraordinary capabilities which they are equipped with. Furthermore, smart- phones usually express their users' personality, which helps in ac- cusing cyber criminals. The smartphones have different hardware and software components that are used to generate different kinds of information, such as audios, videos, pictures, and GPS loca- tions. This information, in turn, enriches the SF. Android is one of the most popular smartphones platforms. Its source-openness and feature-richness make it an ideal forensics platform. Smart- phones developers create applications (Apps) that mainly extend the functionality of the phone by making use of the different com- ponents of the phone. Among Apps, Voice over Internet Protocol (VoIP) Apps are extensively used by the smartphones users for their wide availability and cheap prices. The Skype App is one of the mostly used VoIP Apps. This paper investigates the Skype App in the Android system from forensics perspectives. We design differ- ent experiments that make use of the Skype App by making calls and exchanging chat messages. We show that Skype calls patterns and chat messages not only can be found in both of the RAM and NAND flash memories (even after deleting calls and chats histo- ries and signing out of the Skype), but also can stick there for a long time. The results of this paper can be utilized by the forensics analysts and Law Enforcement Agencies (LEA) in accusing cyber criminals who use the Skype App.

8. REFERENCES

[1] Mohammed I. Al-Saleh and Ziad A. Al-Sharif. Utilizing data lifetime of tcp buffers in digital forensics: Empirical study. Digital Investigation, 9(2):119 – 124, 2012.

[2] Pete Broadwell, Matt Harren, and Naveen Sastry. Scrash: a system for generating secure crash information. In Proceed- ings of the 12th conference on USENIX Security Symposium- Volume 12, SSYM'03, pages 19–19, Berkeley, CA, USA, 2003. USENIX Association.

[3] Jim Chow, Ben Pfaff, Tal Garfinkel, Kevin Christopher, and Mendel Rosenblum. Understanding data lifetime via whole system simulation. In Proc. 13th USENIX Security Sympo- sium, August 2004.

[4] Jim Chow, Ben Pfaff, Tal Garfinkel, and Mendel Rosenblum. Shredding your garbage: reducing data lifetime through se- cure deallocation. In Proceedings of the 14th conference on USENIX Security Symposium - Volume 14, SSYM'05, pages 22–22, Berkeley, CA, USA, 2005. USENIX Association.

[5] Dawson Engler, David Yu Chen, Seth Hallem, Andy Chou, and Benjamin Chelf. Bugs as deviant behavior: a general ap- proach to inferring errors in systems code. In Proceedings of the eighteenth ACM symposium on Operating systems prin- ciples, SOSP '01, pages 57–72, New York, NY, USA, 2001. ACM.

[6] Tal Garfinkel, Ben Pfaff, Jim Chow, and Mendel Rosenblum. Data lifetime is a systems problem. In Proceedings of the 11th workshop on ACM SIGOPS

European workshop, EW 11, New York, NY, USA, 2004. ACM.

[7] Andrew Hoog. Android Forensics: Investigation, Analysis and Mobile Security for Google Android. Syngress Publish- ing, 1st edition, 2011.

[8] Hajime Inoue, Frank Adelstein, and Robert A Joyce. Visu- alization in testing a volatile memory forensic tool. Digital Investigation, 8(Supplement):S42–S51, 2011.

[9] M. Jahanirad, A. L. N. Yahya, and R. M. Noor. Security mea- sures for VoIP application: A state of the art review. Scientific Research and Essays, 6(23):4950–4959, 2011.

[10] Sebastian Schrittwieser, Peter Fru¨ hwirt, Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Markus Huber, and Edgar Weippl. Guess Who's Texting You? Evaluating the Se- curity of Smartphone Messaging Applications. In Proceed- ings of the 19th Annual Network & Distributed System Secu- rity Symposium, February 2012.

[11] Andreas Schuster. The impact of microsoft windows pool al- location strategies on memory forensics. Digital Investiga- tion, 5, Supplement(0):S58 – S64, 2008. The Proceedings of the Eighth Annual DFRWS Conference.

[12] Matthew Simon and Jill Slay. Recovery of skype application activity data from physical memory. In ARES, pages 283–288, 2010.

[13] Jill Slay and Matthew Simon. Voice over ip forensics. In Proceedings of the 1st international conference on Foren- sic applications and techniques in telecommunications, in- formation, and multimedia and workshop, e-Forensics '08, pages 10:1–10:6, ICST, Brussels, Belgium, Belgium, 2008.

ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[14] Jason Solomon, Ewa Huebner, Derek Bem, and Magdalena Sze?ynska. User data persistence in physical memory. Digital Investigation, 4(2):68 – 72, 2007.

[15] Joe Sylve, Andrew Case, Lodovico Marziale, and Golden G. Richard III. Acquisition and analysis of volatile memory from android devices. Digital Investigation, 8(3-4):175–184, 2012.

[16] Vrizlynn L.L. Thing, Kian-Yong Ng, and Ee-Chien Chang. Live memory forensics of mobile phones. Digital Investiga- tion, 7, Supplement(0):S74 – S82, 2010. The Proceedings of the Tenth Annual DFRWS Conference.

[17] Timothy Vidas, Chengye Zhang, and Nicolas Christin. To- ward a general collection methodology for android devices. Digit. Investig., 8:S14–S24, August 2011.

[18] Aaron Walters and Nick L Petroni. Volatools : Integrating volatile memory forensics into the digital investigation pro- cess. Digital Investigation, pages 1–18, 2007.

[19] Svein Yngvar Willassen. Forensic analysis of mobile phone internal memory. In Mark Pollitt and Sujeet Shenoi, editors, IFIP Int. Conf. Digital Forensics, pages 191–204. Springer, 2005.

[20] Maynard Yates, II. Practical investigations of digital foren- sics tools for mobile devices. In 2010 Information Security Curriculum Development Conference, InfoSecCD '10, pages 156–162, New York, NY, USA, 2010. ACM.

[21] Amjad Zareen and Shamim Baig. Mobile phone forensics: Challenges, analysis and tools classification. In Proceedings of the 2010 Fifth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE '10, pages 47–55, Washington, DC, USA, 2010. IEEE Computer Society.