Forensic Analysis of Encrypted Instant Messaging Applications on Android

Khushboo Rathi, Umit Karabiyik
Department of Computer Science
Sam Houston State University
Huntsville, Texas, 77341
Email:[khushboo.rathi, umit]@shsu.edu

Temilola Aderibigbe, Hongmei Chi
Department of Computer and Information Sciences
Florida A&M University
Tallahassee, Florida, 32307
Email:temilola1.aderibigbe@famu.edu, hongmei.chi@famu.edu

***Abstract*—Smartphone market is growing day by day and according to Statista, as of 2017, 68.4% of the U.S. population uses smartphones. Similarly, the amount of information stored on these mobile devices is tremendous and ranging from personal details, contacts, applications data, to exchange of texts and media. This information can become a significant evidence during a digital forensics investigation and thereafter in courts. As Android is one of the leading smartphone operating systems worldwide, it is important to have the knowledge of Android forensics. Moreover, chat messaging between the users becoming the most prominent communication medium particularly among the youth. The exponential increase in the interception of chat messages on mobile devices led to implementation of end to end encryption. This is mainly due to the concerns raised on privacy and security of user data on smartphones. In this paper we analyze widely used encrypted Instant Messaging (IM) applications namely WeChat, Telegram, Viber and Whatsapp. We also show how these applications store data in the Android file system. In addition we also discuss forensic implications of the IM applications that are utilizing encryption. Analysis of artifacts collected from these applications is performed using the Android Debugging Bridge (ADB) tool and some other open source tools. Moreover, we also present the challenges faced during the collection of the forensically important artifacts.**

***Index Terms*—mobile forensics, Android, instant messaging, encrypted communication**

I. Introduction

Smartphones with Android operating system (OS) occupy more than 87% [1] of mobile market share worldwide and with this rapidly gaining market share, there are high chances that the powerful features available in these devices are used for the wrong doing such as harassment through text messaging, committing fraud over e-mail, trafficking of child pornography images, etc. Indeed, mobile devices are already showing themselves to have probative information that is linked to an individual with information such as call logs, contacts, text messages, images, videos, navigating search information stored and geo-location information. If the extracted information of this data storage is known then acquisition and analysis become faster during the course of an investigation.

Through this paper we have analyzed the storage details of these evidence data stored in mobile devices. The methodology we discuss in this paper will be enlightening in acquisition of data from Android devices. In this paper we particularly studied four third party Instant Messaging (IM) applications: WeChat, WhatsApp, Viber and Telegram. As Fig. 1 [2] shows, this paper covers forensic analysis of the IM applications used by significant number of users. The main aim to study aforementioned applications is that the database files in all of those applications are kept encrypted. Availability of data can be of prime interest to the law enforcement agencies to be acquired and analyzed. In certain circumstances, the data stored in these encrypted databases can be made available to the investigators while in other cases the encryption key can even be found hard coded in the application.

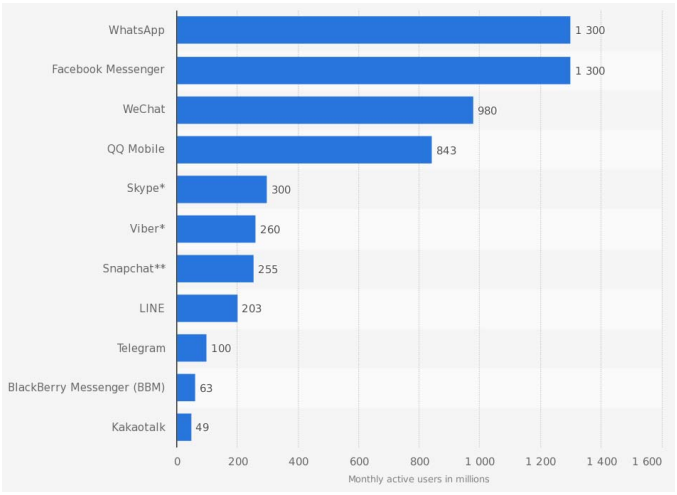

Fig. 1. Most popular global mobile messenger apps as of January 2018, based on number of monthly active users (in millions)

These are widely used instant messaging applications which are also famous for their secure and encrypted chat database files. WeChat has a major market particularly in China with more than 100 million users and Telegram has its major market significantly in Russia. Telegram is famous for its whole application security where even the media files are transmitted in an encryption fashion. Viber introduced the end to end encryption of messages, call and group chats after the version 6.0 released in 2016 and very less research work has been done to understand its new model hence doing the study on

978-1-5386-3449-3/18/$31.00

this application is essential.

Due to the frequent leakage of private conversations to public, messaging apps such as WhatsApp have recently implemented end to end encryption. Thereby making access to private and encrypted data on the users' phones is even more challenging. In this paper, we also investigated the method proposed by Karpisek et al. [3] to decrypt WhatsApp's encrypted database on Android. In addition, this paper helps in understanding the Android file system storage as it differs from the regular computer forensics. It will help the forensic examiners to understand the analyses on the applications' databases in an efficient manner.

This paper is organized as follows. The related work for the instant messaging applications given in Section II. Next, we discuss our methodology in Section III. Section IV presents our experiments and depicts our results. Finally, we conclude our paper in Section V.

II. Related Work

There are some studies previously published for carrying out forensic activity on Android devices related to file system analysis, analysis of various application, analysis of different instant messaging applications. However, with the new versions of Android OS released frequently and also with constant updates in the applications, it is important that continuous study on the changes be tested. Following is a brief discussion on the related work performed previously. We reviewed different IM applications to better understand the data structures used to store the information in these applications.

The method in [4] deals with discussing the rooting scripts and methods for gaining the super privileges on Android devices. Using 'dd' command through the Android Debug Bridge (ADB) the image of the device is acquired. Acquired images are analyzed using traditional forensic tools such as Cellebrite, Paraben, etc.

There have been studies particularly focused on developing methods for analysis of WhatsApp IM application. The authors in [5], [6] deal with the older version (less than 6.0) of WhatsApp where a backup of the database was possible. In the current version however, this technique cannot be used for data retrieval. In [7], authors discussed the ChatSecure application and the analysis of its encrypted database. According to the results presented, the decryption key that is used during encryption was found on the database folder, however it is only available in rooted devices.

In [8], different mobile GPS mapping application such as Google maps, Apple maps, Waze, MapQuest, Bing map and Scout map are analyzed on both Android and iOS devices. The data acquisition was done using XRY and Cellebrite forensic tools and smart navigation parser. Command line tool was developed to compare the result with the already existing tools.

In [9], the forensic analysis of messaging application such as Messenger, WhatsApp, Viber, WeChat is done using FTK imager, FTK, SQLite browser, oxygen, UFED physical analyzer tools and experiment results are compared. However, these application had significant updates therefore, the availability of data now varies from the results provided.

In [10], [11], [12], authors describe the methodology for forensic analysis of geo-location information stored on different location service application including Google maps and Pokémon GO on the Android devices. Data is acquired by rooting the devices and collected data is investigated.

In [13], the database file location of the WeChat application is discussed. Reverse engineering techniques are used to retrieve the database files. However, the techniques used and presented in this paper are now obsolete on the new version of the WeChat application.

In [14], the Telegram artifacts are discussed however the authors were not able to locate the chat database storage location. They only discuss the different types of chats available in Telegram and how secure Telegram is. In [15], authors present detailed discussion about the forensically relevant artifacts stored by Telegram messenger. The paper also discusses the methods to reconstruct the contact lists, text and non-textural messages in addition to the log files for voice calls. The message details which this paper was able to retrieve was the metadata information. The actual text messages were not available to be retrieved because of the encrypted nature.

Gudipaty and Jhala [16] have presented extraction of WhatsApp database using third party applications without rooting the device if it is not rooted. The paper proposed a step by step process to acquire WhatsApp encrypted backup files, decrypt them and parse them in human readable format. It was noticed that WhatsApp automatically backs up its database every day at 4:00AM and saves it in a folder called WhatsApp on an sdcard or internal storage of the Android device. The backups are valid for a period of 7 days before the oldest backup is over-written by a new backup. The backup conversations are important because it may contain valuable chat messages that may have been deleted from the existing messages. In order to decrypt the database, they used an application call WhatsApp key/DB Extractor to extract the key files present in the device and the software successfully extracted the WhatsApp's encryption key, unencrypted msgstore.db (which contains all the undeleted messages), unencrypted wa.db (which contains a complete listing of the user's contacts including phone number, display name and any other information that was provided during the WhatsApp registration). They were able to use the extracted encryption key to decrypt the encrypted backup database files using WhatsApp Viewer and view the extracted file using a SQLite Database Browser or it can be export to an HTML file using WhatsApp Viewer.

Several works have been published with respect to various instant messaging applications. In all of those methods customized rooting applications have been used for obtaining a forensic image and performing analysis with commercial or open source tools.

III. The Methodology

The objective of this study is to analyze the data storage locations for different commonly used IM applications in the

Android phone. The study is performed using Android phones with different Android OS versions running on them. Testing is performed on both rooted and un-rooted phones. This study not only determines the data storage on the file system but also categorize the applications for which we can retrieve the information without needing to have the super user privileges. The following are the steps involved in this study:

1) Telegram, WhatsApp, Viber and WeChat application were installed on both rooted an un-rooted Android phones. Table 1 give the specifications used for this study.

TABLE I
SPECIFICATIONS OF EXPERIMENTAL ANDROID DEVICES

Android Devices	Type	OS - Version
Samsung Galaxy S4	Rooted	Android 4.3.1
Motorola Moto G3	Un-rooted	Android 6.0
Samsung Galaxy S7	Un-rooted	Android 6.0

2) These devices were populated with test data for all four applications so that data can be acquired using ADB tool.
3) USB debugging option on the Android device was enabled, so that the device can be recognized by ADB tool and it can access the Android file system location.
4) All the applications' data storage locations are be analyzed. *'ADB pull'* command on the rooted device and *'ADB backup'* command on unrooted device is used to collect the evidences on the phone.
5) The experiments were performed on all the devices separately and results are documented.

This is the general methodology followed in this study to find the artifacts in all of the four applications under study.

IV. EXPERIMENTS, RESULTS AND DISCUSSIONS

As discussed earlier in Section II, currently available literature is outdated as new updates becomes available quite frequently. In this section, we will discuss the experiments performed in our study and present our results with appropriate discussion.

A. WeChat Forensics

More than 100 million active user accounts are shown on the Google play store while recently downloading WeChat. After installing the application with the latest version 6.5.16 on the Android device, application is placed in *"/data/data/com.tencent.mm/"* and *"/sdcard/Tencent/MicroMsg"*. Table II gives the overview of the application's specification used to conduct our experiments on WeChat. As we can see that we need a root privilege to access the *"com.tencent.mm"* directory, hence process for acquisition is different for rooted and un-rooted devices. First, we examined the rooted phone for which we used *'ADB pull'* command with the help of it we were able to export the entire *"/data/data/com.tencent.mm"* directory. For the un-rooted phone we had to degrade from the latest version to version 6.0, because *'adb backup'* option is only available for the versions 6.0 and below. For this reason we used *'adb install'* command to install the application and then used *'adb backup'* to backup the user data.

TABLE II
DETAILS OF WECHAT APPLICATION

Application Name	WeChat
Package Name	com.tencent.mm
Version	6.5.16

WeChat creates a unique identification number *"uin"* for each user and places a corresponding personal data folder under the main directory. The "uin" for one of our devices looks like *"d041d0bbbba7e971435119a7030bf330"*. We used the symbol <uin>to denote this personal folder name.

Table III shows the location of the artifacts we found during examining the application. We believe these artifacts are invaluable information for the law enforcement agencies.

TABLE III
LOCATION OF ARTIFACTS IN WECHAT APPLICATION

Artifacts	Location
Encrypted SQLite database of chat message (EnMicroMsg.db)	/data/data/../ <udir>/EnMicroMsg.db
SQLite database for moments	/data/data/../ <udir>/SnsMicroMsg.db
Send and Received Images	/sdcard/tencent/MicroMsg/ <udir>/image2
Send and Received Videos	/sdcard/tencent/MicroMsg/ <udir>/video
Downloaded Documents	/sdcard/tencent/MicroMsg/ Download
Send and received audio file	/sdcard/tencent/MicroMsg/ <udir>/video2
All the images captured via 'take photo'	/sdcard/tencent/MicroMsg/ WeChat
Downloaded images and video	/sdcard/tencent/MicroMsg/ WeChat
Profile picture	/sdcard/tencent/MicroMsg/ <udir>/avatar
Send and Received Stickers	/sdcard/tencent/MicroMsg/ <udir>/emoji

1) Decrypting the Messages Database: The SQLite database of a user's chat is named *EnMicroMsg.db* and this database is encrypted for the user privacy with the help of SQLCipher. It was found that the decryption key can be calculated from the device's unique International Mobile Equipment Identity (IMEI) number and the <uin>of the current WeChat user. One of the way to find the IMEI number was to enter ∗#06#, we did that and found the IMEI number of the device then used the formula below to get the decryption key.

$$Key = left7(MD5(IMEI + uin) \quad (1)$$

The method *left7* used in Eq. 1 extracts the first seven characters of a string. Using this key we are able to open the database in the SQLite browser and decode the message.

B. Telegram Forensics

Telegram is a very popular IM service with more than 100 million active users listed on Google play store. Table IV

depicts the overview of the application's specification used to conduct experiments on Telegram. It provides a secure one-to-one, one-to-many and many-to-many communication. Most of the Telegram users are active in the region of Russia however, with the strong encryption property of this application it is becoming popular worldwide as well. It has been reported that many criminal activities are taking place using this application hence it is important to examine and find crucial details which may act as evidence in many cases. Telegram has been pointed out as a threat to the national security by some leaders [17].

As discussed, we have utilized two types of Android devices, rooted and un-rooted. While analyzing the Telegram application it was found that none of the information was accessible by the un-rooted device as the application stores chat database and media files in the user partition for which the super user privilege is required. Table IV shows the package name and version for the Telegram application on which our experiments were conducted.

TABLE IV
DETAILS OF TELEGRAM APPLICATION

Application Name	Telegram
Package Name	org.telegram.messenger
Version	4.5.1

Telegram Messenger stores various data in the internal memory of the device particularly in the user partition, the folders are located in the *"/data"* directory. Table V depicts the artifacts and their corresponding locations in the Android file system.

TABLE V
LOCATION OF ARTIFACTS IN TELEGRAM APPLICATION

Artifacts	**Location**
Encrypted SQLite database of chat message (Cache4.db)	/data/data/org.telegram.messenger/ files/Cache4.db
Detail about account used	/data/data/org.telegram.messenger/ shared_prefs/userconfing.xml
Profile photo	/data/media/0/Android/data/ org.telegram.messenger/cache
Copies of files send and received	/data/media/telegram

In order to retrieve the database files, we used the *'adb pull'* command on the rooted phone. It was found that Telegram stored data in the complex data structure called Telegram data structure and it is stored in a binary serialized form. It was found that there are different type of chats which can take place using this application. All the chat data was encrypted and it was not possible to decrypt it without the encryption key. The following items show different types of chats that are possible in Telegram application [15].

a) One to one regular chats:: Server-client encryption is used to encrypt messages. This type of chat is also called cloud chat, where a copy of the message is kept and can be synchronized on all the devices. For each pair of users, Telegram allows only a single regular chat.

b) One to one secret chats:: In this type of chat, messages are exchanged directly between users and never get stored on the Telegram server. In order to keep the communication safe, end-to-end encryption is used. It is also possible that multiple pairs may create and use as many secret chats as needed.

c) One to many channels:: A channel may be either public or private, if it is public it can be joined by any one as it will be published on the Telegram server. Moreover, the username created the chat can be seen by everyone however, if the channels is private then only the invited members can see it.

d) Many to many group:: In this type, the message will be broadcasted to every member in the group. There are two type of groups – standard group which can have up to 200 members and super groups which can have 30,000 members and will have unified history meaning the message is deleted then it will be removed from all users.

According to the official website of Telegram [18] the encryption algorithms used in the application are 256-bit symmetric AES encryption, 2048-bit RSA encryption, and Diffie–Hellman secure key exchange. It is also discussed that cloud chat uses server-client encryption, and an additional layer of encryption is used for secure chat. Based on our experiments, we found that the keys are held only by the chat participants. In the secure chat, Telegram gives the option for self-destruct times which can be set by the participants. It can be used to destroy messages, photos, videos and files in set amount of time. The message will then disappear from the chat history of the both the participants.

Applications such as WhatsApp and Viber allow decryptable backup however Telegram offers two distinct types of chats. Telegram disables default system backup and provides an integrated security-focused backup solution to all users in the form of cloud chats (keeping the data in the cloud). It also has different entity in the form of secret chats, where the full control is given to the users.

C. WhatsApp Forensics

WhatsApp is a cross-platform IM service. As of 2016, it reached over 1 billion users and has continued to grow. Table VI shows the overview of WhatsApp application's specification used to conduct our experiments. WhatsApp can be used across various operating system including Android, iOS, Blackberry, Symbian. Due to the frequent leaking of private conversations, messaging apps such as WhatsApp have recently implemented end to end encryption. Thereby making access to private and encrypted data on a user's phone is challenging. In this section, we focus on discussing the procedures and tools to reveal this encrypted social messaging application's database.

WhatsApp can backup its database on the user directory daily, weekly or monthly depending on which the user selects. However, the contents of the backed up database on the user directory cannot be read in plaintext. In order to access the content, the decryption key would be needed and this is kept

TABLE VI
DETAILS OF WHATSAPP APPLICATION

Application Name	WhatsApp
Package Name	com.whatsapp
Version	2.17.417

TABLE VII
DATABASE FILES IN WHATSAPP

Tables	**Description**
Key.db	The encryption key
Msgstore.db	Decrypted WhatsApp backup contains all the chats
Wa.db	Contacts information

in the root directory of the device. Which implies that the device would have to be rooted in order to access the key. Our focus was to discover a way to extract the information without rooting the device. Hence, we started to set up the following environment and tools to prepare our desired data acquisition:

- Workstation (Os: Windows 10) with Java installed
- Galaxy Samsung S6 with WhatsApp installed and USB Debugging enabled
- Universal ADB Driver
- WhatsApp KeyDB Extractor
- WhatsApp Viewer
- SQLiteSpy

The tools mentioned above were installed on the workstation. Once the WhatsApp Key/DB Extractor was launched and the mobile device was connected to the workstation, we were then required to perform a full backup on the mobile device which backups and extracts all the WhatsApp folder content from the root directory. After the backup was completed, the files in Table VII are stored in a folder called *Extracted* inside the WhatsApp Key Extract folder.

The *Msgstore.db* file can be viewed using the SQLiteSpy and it contains all the test communications that took place as shown in the Fig. 2. Related to our experiments, Fig. 2 shows a screen shot of the experiment conversations between two people with timestamps associated with messages. As also shown, the investigator would be able to go through the chat messages extracted and detect forensically valuable messages as evidence. Note that, the chat messages shown in this experiment were all extracted without rooting the Android device.

D. Viber Forensics

The messaging application Viber has over 100 million active users on the Google Play has introduced an end-to-end encryption for all messages, calls on its platform, including the group chats after the version 6.0. Table VIII presents the overview of the application's specification used to conduct experiments on Viber. After installing the application with the latest version on the Android device, the application is placed in *"/data/data/com.viber.voip"* and media files are located in *"/sdcard/viber/media"*.

Fig. 2. Screenshot of WhatsApp messages recovered from unrooted Android device

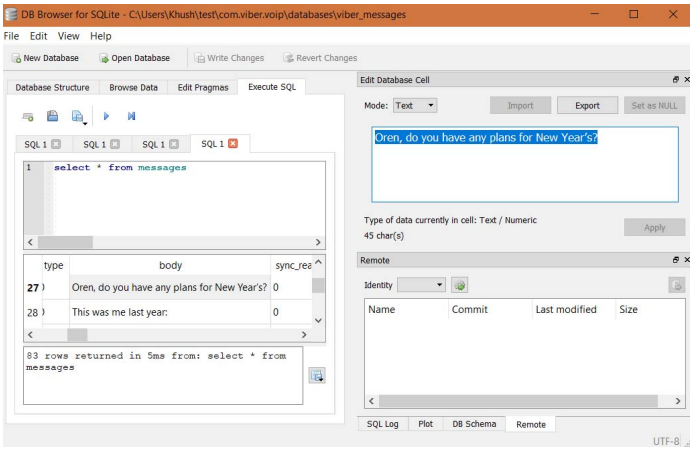

Fig. 3. Screenshot showing Viber chat Artifacts found on Android Phone

The experiment was conducted on both rooted and unrooted Android devices. The media files are located on the external memory on the sdcard. Table IX contains the artifact locations for all the media files retrieved during analysis.

It was found that the backup flag was set *'on'* for the Viber application hence *"adb backup"* command was used for taking the backup on the unrooted devices. *"adb backup -f [backup_path]"* command was used to collect the *'.ab'* file for Viber. The *.ab* file was converted to the *'.tar'* file format by Android backup extractor. The .tar file can be opened using any windows archive utilities such as WinRAR or 7zip. It was found that only the manifest file was available after the backup. No access to the databases files were granted.

TABLE VIII
DETAILS OF VIBER APPLICATION

Application Name	Viber
Package Name	com.Viber.voip
Version	6.3.1

TABLE IX
LOCATIONS OF THE MEDIA ARTIFACTS RECOVERED FROM VIBER APPLICATION

Artifacts	Location
User Photos	/sdcard/viber/media/User Photos
Sent and Received Viber Images	/sdcard/viber/media/Images
Sent and Received Viber Videos	/sdcard/viber/media/Videos

TABLE X
IMPORTANT DATABASE FILES IN VIBER

Database Files	Location	Description
viber_data	/data/data/ com.viber.voip /databases	It contains information about the user's contacts
viber_messages	/data/data/ com.viber.voip /databases	It contains the information about the application usage.

On the other hand, *"adb pull"* command was used to retrieve the database files from Viber application installed on the rooted devices. Table X presents the important details of the database files.

Although these database files do not have the extension .db files, these are the most important databases files of Viber. Table XI shows the tables in viber_data database file and Table XII shows the tables in viber_messages database file.

Even though the Viber messages are said to be encrypted, when we opened the message table in the SQLite DB browser some messages were found to be in plaintext. Fig. 3 shows the example messages exchanged.

V. CONCLUSION

The main goal of this paper is to study and analyze the most popular applications' encrypted data storage locations in Android devices. We discuss the challenges faced during data extraction from the encrypted databases. Our results show that WeChat and Viber encrypted database files can be retrieved from a rooted phone, WeChat database can be decrypted by utilizing the encryption key which can be generated by the IMEI number and unique identifier of the phone. In addition, WhatsApp messages can be retrieved from un-rooted devices. It is also possible to collect the database files for WeChat from unrooted device by degrading the application to the lower version. In case of the Telegram application, we show how this application is more secure than the other applications analyzed in this paper. However, if the phone is rooted, then it is possible to recover database artifacts from Telegram. Even though the database files are recovered, the encrypted chat database files cannot be decrypted without the encryption key.

The forensic analysis of targeted application on their current versions provides important insight to the forensic investigators as well as the researchers. This work will allow the investigators having a clear perspective about where to look for the relevant data when any of those applications involved in their case.

TABLE XI
TABLE DESCRIPTION IN VIBER_DATA

Tables	Description
Calls	It is an empty table even though calls were made from the application
phonebookcontacts	This table contains the names of all the contacts stored in the phonebook of the device
phonebookdata	This table contains the details about the phone number of the contacts in the device.

TABLE XII
TABLE DESCRIPTION IN VIBER_MESSAGES

Tables	Description
conversations	It as the name of each group with whom the receipt communicated. Unique Id is given to each group
messages	This contains the conversation messages. The address column was encrypted, which is the phone number of remote party in the conversation. The body column had the actual text which was shared

REFERENCES

[1] Statista. Global mobile os market share in sales to end users from 1st quarter 2009 to 2nd quarter 2017, 2017. Accessed: 2018-01-27.

[2] Statista. Most popular global mobile messenger apps as of january 2018, based on number of monthly active users (in millions), 2018. Accessed: 2018-01-27.

[3] Filip Karpisek, Ibrahim Baggili, and Frank Breitinger. Whatsapp network forensics: Decrypting and understanding the whatsapp call signaling messages. *Digital Investigation*, 15:110–118, 2015.

[4] Jeff Lessard and Gary C. Kessler. Android forensics: Simplifying cell phone examinations. *Small Scale digital Forensics Journal*, 4(1), 2010.

[5] Cosimo Anglano. Forensic analysis of whatsapp messenger on android smartphones. *Digital Investigation*, 11(3):201–213, 2014.

[6] Aditya Mahajan, M. S. Dahiya, and H. P. Sanghvi. Forensic analysis of instant messenger applications on android devices. *International Journal of Computer Applications*, 68(8), 2013.

[7] Cosimo Anglano, Massimo Canonico, and Marco Guazzone. Forensic analysis of the chatsecure instant messaging application on android smartphones. *Digital Investigation*, 19(Supplement C):44–59, 2016.

[8] Ibrahim; Moore, Jason; Baggili and Frank Breitinger. Find me if you can: Mobile gps mapping applications forensic analysis & snavp the open source, modular, extensible parser. *Journal of Digital Forensics, Security and Law*, 12(1), 2017.

[9] Justin Grover. Android forensics: Automated data collection and reporting from a mobile device. *Digital Investigation*, 10(Supplement):S12–S20, 2013.

[10] Yi Sun. Geo-location forensics on mobile devices.

[11] Stefan Maus, Hans Hoefken, and Marko Schuba. Forensic analysis of geodata in android smartphones. 2017.

[12] Joshua Sablatura and Umit Karabiyik. Pokémon go forensics: An android application analysis. *Information*, 8(3):71, 2017.

[13] Songyang Wu, Yong Zhang, Xupeng Wang, Xiong Xiong, and Lin Du. Forensic analysis of wechat on android smartphones. *Digital Investigation*, 21(Supplement C):3–10, 2017.

[14] G. B. Satrya, P. T. Daely, and M. A. Nugroho. Digital forensic analysis of telegram messenger on android devices. In *2016 International Conference on Information & Communication Technology and Systems (ICTS)*, pages 1–7, 2016.

[15] Cosimo Anglano, Massimo Canonico, and Marco Guazzone. Forensic analysis of telegram messenger on android smartphones. *Digital Investigation*, 2017.

[16] LP Gudipaty and KY Jhala. Whatsapp forensics: decryption of encrypted whatsapp databases on non rooted android devices. *Journal of Information Technology & Software Engineering*, 5(2):1, 2015.

[17] James Cook. Theresa may will single out messaging app telegram and call it a 'home to criminals and terrorists', 2018. Accessed: 2018-01-27.

[18] Telegram. A new era of messaging., 2018. Accessed: 2018-01-27.