



MASTER THESIS

In order to obtain the

Professional Master 2

In

MDFS option: Forensic Science

Presented and defended by:

Louaye LAMAA

On Monday July 22, 2019

Forensic Analysis of WhatsApp SQLite Databases on the Unrooted Android Phones

Supervisor:

Dr. Hasan KAZAN

Reviewers:

Colonel Albert Khoury

Major Hany Kallassy

Lebanese University - Faculty of Sciences

Acknowledgements

This thesis would not have been achievable without the support and assistance from many people. I would like to thank all of them for being part of this journey and making this thesis possible.

I would like first to express my deepest gratitude to my supervisor Dr. Hasan Kazan for his guidance, patience, and support of my study.

I would like to thank the members of the committee for offering their time to read my thesis and for being able to attend my presentation. I honestly appreciate it.

Finally, I would like to recognize the biggest support that came from my family. I am deeply thankful for their love, sacrifices, and encouragement.

Abstract

WhatsApp is the most popular instant messaging mobile application all over the world. Originally designed for simple and fast communication, its privacy features, such as end-to-end encryption, eased private and unobserved communication for criminals aiming to commit illegal acts.

In this paper, we present the forensic analysis of the artifacts left by the encrypted WhatsApp SQLite databases on the unrooted Android devices.

In order to provide a complete interpretation of the artifacts, we perform a set of controlled experiments to generate them. Once generated, we identify their storage location and databases structure on the device. Since the data is stored in an encrypted SQLite database, we first discuss its decryption. Then, we show how to analyze the artifacts and how they can be correlated to cover all the possible evidence.

The results show how to reconstruct the list of contacts, the history of exchanged textual and non-textual messages as well as the details of their contents. Furthermore, this paper shows how to determine the properties of both the broadcast and the group communications in which the user has been involved as well as of the feature called status. Finally, we show how to reconstruct the logs of the voice and video calls.

These results show that the reconstruction of the WhatsApp data from the SQLite databases is possible and this data persists on the phone even after the uninstallation of WhatsApp.

Keywords: Mobile forensics, WhatsApp messenger, Instant messaging, Android, Unrooted devices, Data recovery, SQLite databases.

Contents

Acknowledgements	i
Abstract	ii
1 Introduction	1
2 Related Works	4
3 SQLite Forensics	5
3.1 What is SQLite	5
3.2 SQLite file format	5
3.2.1 Pages	5
3.2.2 Database header	7
4 The Analysis Methodology	8
4.1 Requirements	9
4.2 Workflow	9
4.3 Sets of experiments	9
4.3.1 Experiments concerning contacts	9
4.3.2 Experiments concerning the private chat communication between user and contact	10
4.3.3 Experiments concerning the messages state	11
4.3.4 Experiments concerning the broadcast and group messages	11
4.3.5 Experiments concerning voice and video calls	12
5 Forensic Analysis of WhatsApp Messenger	13
5.1 Analysis of WhatsApp functionalities	13
5.2 Location and types of WhatsApp artifacts	14
5.3 SQLite databases decryption	15
5.4 Analysis of the contacts database "wa.db"	19
5.4.1 The structure of the contacts database "wa.db"	20
5.4.2 Reconstruction of the contacts list	20
5.4.3 Blocked contacts	22
5.5 Analysis of the chat database "msgstore.db"	24
5.5.1 The structure of the chat database "msgstore"	24
5.5.2 Determination of the chat history	25
5.5.3 Analysis of the messages content	26
Multimedia files	26
Contact cards	29
Geolocation coordinates	29
Attachments	30
5.5.4 The determination of the messages state	31
5.6 Multiple message destinations	32

5.6.1	Broadcast messages	32
5.6.2	Group chat	33
5.7	Voice and call logs	35
5.8	WhatsApp status analysis	37
5.9	Deleted data	38
6	WhatsApp Security	40
7	WhatsApp in the Court	41
7.1	Admissibility	41
7.2	Authenticity	41
7.3	Relevance	42
7.4	Digital forensics report	43
8	Conclusion	46

List of Figures

1.1	Instant Messaging applications surpass the Social Network apps.	1
1.2	WhatsApp monthly active users compared to others' IM apps.	2
3.1	SQLite page layout.	6
3.2	SQLite is set of data pages of fixed size.	6
3.3	Set of free list pages that contain deleted data.	7
3.4	The SQLite database header.	7
4.1	Workflow of the analysis methodology.	10
5.1	WhatsApp messenger artifacts.	14
5.2	The encrypted databases backups of msgstore.db stored in the internal memory.	15
5.3	The encryption process.	16
5.4	The PowerShell script used to decrypt the databases.	16
5.5	WhatsApp databases and key extraction.	17
5.6	DBs and the key are extracted from the phone.	17
5.7	WhatsApp viewer tool.	17
5.8	The database is successfully decrypted.	18
5.9	The decrypted database "msgstore" is opened in DB Browser for SQLite.	18
5.10	Damaged database: Disk image is malformed.	19
5.11	Recovered table "messages" from the damaged database "msgstore".	19
5.12	"wa_contacts" table - the individual contacts records.	21
5.13	"wa_contacts" table - the group contacts records.	21
5.14	"wa_block_list" table - the blocked contacts records.	22
5.15	Extracting the blocked contact information using a SQL query.	22
5.16	Reconstruction of the chat history.	25
5.17	The three types of a multimedia message.	26
5.18	The image message content: the sender and the recipient records.	27
5.19	LEFT JOIN Venn diagram.	27
5.20	LEFT JOIN operation between messages and message_thumbnails tables.	28
5.21	"message_thumbnails" table - the thumbnail of an image message extracted using LEFT JOIN query.	28
5.22	The contact card message.	29
5.23	The geolocation message.	29
5.24	"message_thumbnails" table - the thumbnail of a location message.	30
5.25	The attachments messages records.	30
5.26	"message_thumbnails" - the thumbnail of the document message extracted using LEFT JOIN.	31
5.27	The possible states of a message.	32
5.28	The broadcast messages records.	33
5.29	The broadcast message - recipient side.	33
5.30	The group chat records.	34
5.31	The group records created when a member leaves the group.	35

5.32	Timeline of the chronology of the group composition.	35
5.33	"call_log" table - voice and video calls records.	36
5.34	"jid" table - the phone number of the caller	36
5.35	The WhatsApp status records.	37
5.36	"message_thumbnails" table - identification of the status picture using SQL query.	38
5.37	Deleted messages: the <i>NULL</i> value.	38
5.38	The structure of the B-tree pages.	38
5.39	The output of the script used to extract deleted data.	39

List of Tables

4.1	The user contacts experiments. User 1 and User 2 are the WhatsApp users involved in the experiments.	10
4.2	Experiments concerning all the types of messages exchanged privately.	11
4.3	Message state experiments.	11
4.4	The broadcast and group messages experiments.	11
4.5	WhatsApp voice and video calls experiments.	12
5.1	The structure of the contacts database "wa.db" - table "wa_contacts" - information set by the WhatsApp system.	20
5.2	The structure of the contacts database "wa.db" - table "wa_contacts" - information set by the phonebook.	20
5.3	The structure of the chat database "msgstore.db" - table "messages" - message characteristics.	24
5.4	The structure of the chat database "msgstore.db" - table "messages" - message content.	25
5.5	The reconstruction of the WhatsApp calls history.	36

List of Abbreviations

ADB	A ndroid D ebug B ridge
AES	A dvanced E ncryption S tandard
App	A pplication
DB	D ata B ase
IM	I ntant M essaging
iOS	i phone O perating S ystem
IT	I nformation T echnology
OS	O perating S ystem
SD	S torage D evice
SMS	S hort M essage S ervice
SQL	S tructred Q uery L anguage
TB	T erabyte
RDMS	R elational D atabase M anagement S ystem

Chapter 1

Introduction

Over a decade ago, regular mobile phones offered the Short Message Service (SMS) as an alternative to the instant messaging (IM) that existed on the internet at that time. This service failed to offer the convenience of real-time texting which is available in the IM. Nevertheless, the born of the smartphones in 2007 opened the doors for real-time communication in the mobile phones, through the instant messaging applications such as WhatsApp. This alternative was more convenient than the traditional SMS services not only because of the real-time messaging feature, but also due to the much lower costs associated with their usage. IM mobile applications have also gained popularity at the expense of the social media platform, because of the more private structure they have been set upon. At the end of 2017, the instant messaging applications surpassed the social media applications in term of daily active users as shown by figure 1.1. [1]

Although there are different instant messaging applications nowadays, WhatsApp remains

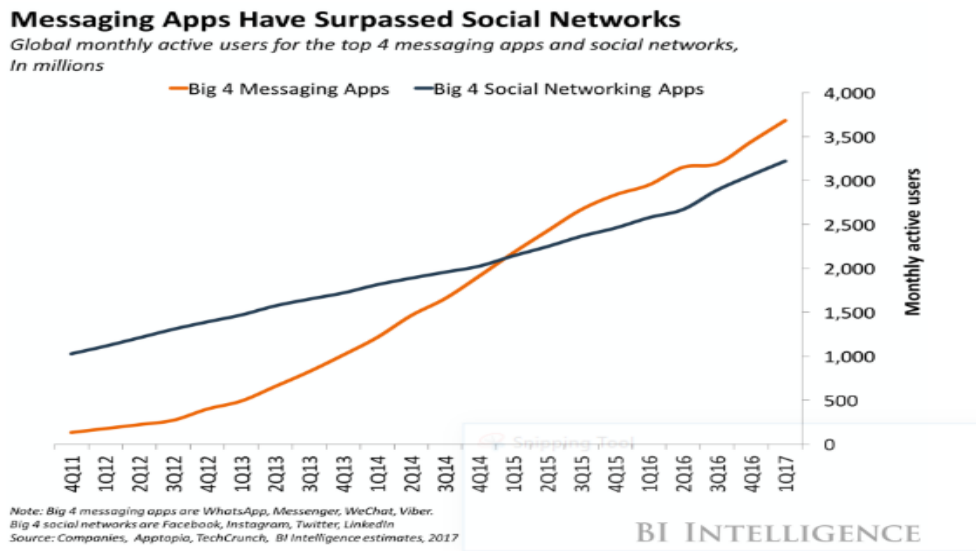


FIGURE 1.1: Instant Messaging applications surpass the Social Network apps.

the most popular of them all around the world. Currently, WhatsApp tops the list of IM applications monthly active users, with more than 1.6 billion users, as shown by Statista in figure 1.2 [2]. This huge number of users also means remarkably big data getting transferred through the app. With that in mind, the way WhatsApp handles this data is a case to investigate.

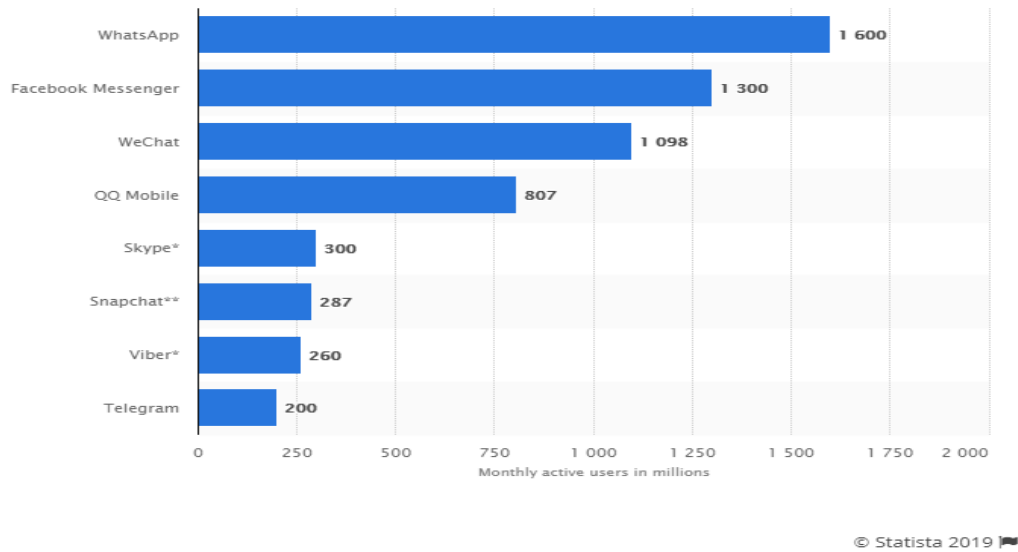


FIGURE 1.2: WhatsApp monthly active users compared to others' IM apps.

The messages exchanged on early versions of WhatsApp were kept in SQLite local databases on the devices. The database file was, in fact, not encrypted, which means that WhatsApp chat records were vulnerable to intruders, putting users' data at risk. On the other hand, the more recent versions of the application have seriously reconsidered the database security, and they have encrypted the databases following the custom Advanced Encryption Standard (AES).

The technology itself allows for both good and bad behaviors. Due to the privacy, and in most cases encryption, that come with the IM apps, they are becoming a common means for criminals to involve in illegal actions such as drug dealing, hate speech, child pornography, terrorist acts and more.

The importance and the need to retrieve electronic evidences from criminals' devices becomes a part of any forensic process [3]. Databases, like WhatsApp's, are crucial in this process, not only because average criminals don't have the technical knowledge to access/modify them, but also because that's where we are presented with the challenge of the encryption.

In this thesis, we address the forensic analysis of the artifacts left by the encrypted WhatsApp SQLite databases on the unrooted Android devices. The analysis also includes the uninstalled WhatsApp app in order to see if these artifacts persist after the deletion of the WhatsApp messenger.

Because of the limited time, the study is limited to the Android OS as according to Statista [4] in the second quarter of 2018, 88 percent of all smartphones sold to end users were phones with the Android operating system.

Several works deal with this topic. In our work, we cover the analysis of more artifacts such as the call logs and the status. In addition, we show how to decrypt the SQLite databases as we work on the unrooted phones. Furthermore, we show how to use the SQL queries to extract data from different tables. Also, since it's possible that a SQLite database may be corrupted, we discuss how to deal with this database.

In the rest of this thesis, we review existing works in chapter 2 and we explain what is the SQLite databases in forensics in chapter 3. In chapter 4, we describe the methodology, the tools and the experiments performed in this study. Then, in chapter 5, we talk about

the forensic analysis of the WhatsApp messenger. In chapter 6, we give some tips about the WhatsApp data protection. In chapter 7, we address the use of WhatsApp as evidence in the court. Finally, in chapter 8 we conclude the thesis.

Chapter 2

Related Works

The forensic analysis of smartphones takes a lot of attention in the literature. Most of papers and books in this subject focus on Android and iOS. [5] [6] explained very deeply the guide to use while working on these two OS, giving key strategies and techniques to extract and analyze the forensic artifacts from mobile phones. In this study, we benefited from this work in order to extract and analyze the data generated by WhatsApp messenger running on the Android OS.

One of the most studied domains in mobile forensics is the applications installed on the device and mainly the IM apps. The importance and the popularity of the IM apps increase the number of works published, as well as the range of applications analyzed.

Zhang et al. [7] focus on the forensic analysis of WeChat on the Android phones, Anglano [8] focuses on the analysis of Chat Secure and Walnycky [9] discusses the analysis of 20 popular IM messenger on the Android phones while Owens [10] focuses on the analysis of the IM applications on the iOS.

Anglano [11] studies the telegram messenger, providing a general methodology in the analysis of Android applications that was the support of the methodology followed in this thesis. As well, Zhang [12] works on four popular Android IM apps (WhatsApp, Facebook messenger, Line, and Hangouts) while Rath [13] discusses in-depth the analysis of the encryption of various IM apps. Azfar [14] proposes a taxonomy outlining the forensic importance of the evidence generated by the IM apps.

All these works are considering the analysis based primarily on the artifacts presented in the device and the encrypted databases generated by these apps. Their works differ from ours for both the IM apps studied and the state of the device.

In the literature, and in addition to the work of Zhang [12] which is very limited, three studies focus on the analysis of the WhatsApp messenger on the Android phones. However, Thakur [15] and Mahajan [16] focus on the analysis of a part of the artifacts left by WhatsApp (just the chat database). Anglano [17] discusses the analysis of the WhatsApp focusing on the contacts and chat databases on the rooted Android phones without aiming to study their encryption. In our study, we explain the decryption of the encrypted databases on the unrooted Android phones. In addition, we cover the new features that were not available at the time of the previous works as well as the updated stored values and tables of these databases. Furthermore, we show how to use the SQL queries to gather information from different tables to ensure that all evidence is extracted from the databases.

Chapter 3

SQLite Forensics

3.1 What is SQLite

SQLite is a special-purpose programming language designed for managing data held in a relational database management system (RDMS). SQLite differs from most databases in that other systems are client-server based, meaning that an interaction between the user computer with another computer (server), that executes any queries. SQLite, on the other hand, is installed and run as a single application or file on the user computer (device).

SQLite is the most widely deployed database in the world and can be found on most devices, running on every Android, every mac, every web browser, and most automotive multimedia systems [18].

At the end of 2018, it was estimated that the number of smartphones in the world is over 2.3 billion [19]. Almost every one of these will have SQLite databases running on it and the majority of the devices will use many more than one database. The authors of SQLite estimate that more than 1 trillion SQLite databases are in use today.

3.2 SQLite file format

A database in its simplest form is a collection of tables. Each table has a strict format with several columns or fields where data is stored. In most cases, Tables are related to each other through one of the columns, called a key, because this is the relation in a relational databases.

SQLite databases are well-normalized system (reduce data duplication and increase data integrity). This is done by indexes that are represented via Binary trees (B-trees). In fact, these B-trees are a special form of table.

In the rest of this section, we explain the structure of this database [20].

3.2.1 Pages

SQLite at the highest level is a collection of fixed size pages (figure 3.2) which each one have a size power of 2 (512 minimum to 65536 maximum). No other page size is allowed. The smallest database is a single page (512-bytes) and the largest is about 140 TB. Each page has a header and different cells where the information is stored. The structure of an SQLite page is shown in figure 3.1

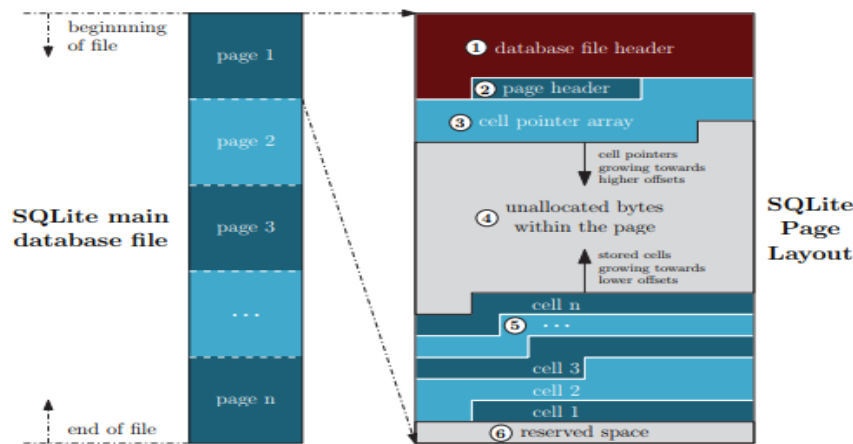


FIGURE 3.1: SQLite page layout.

There are five different types of page :

- **B-tree page:** The majority of the pages in the database. SQLite uses two types of B-tree: The "table B-tree" and the "index B-tree". Each type can be internal or leaf pages.
- **Overflow page:** If all the data can not be stored in one page, the rest of this data is stored on one or more overflow pages. These pages form a chain between the different pages on the database.
- **Freelist page:** Unused pages that can be re-used.
- **Pointer map page:** the role of this type is just to track the usage of the database. They do not contain any user data.
- **Lock-byte page:** This is a single page of 512 bytes. No forensic importance exists in this page.

The first page of the database starts with 100-bytes database header and the remaining bytes are either internal or leaf table B-tree. The rest pages do not have a header.

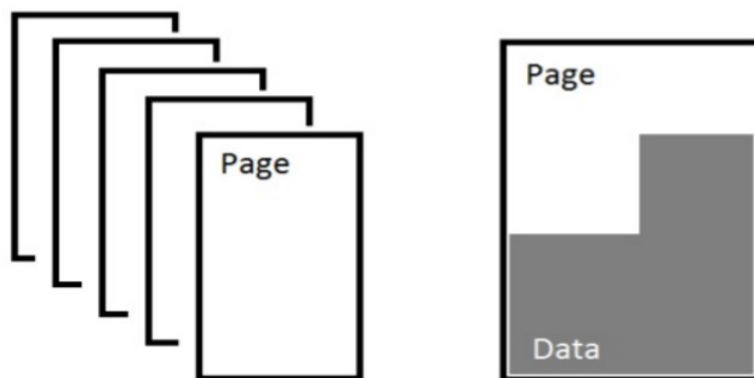


FIGURE 3.2: SQLite is set of data pages of fixed size.



FIGURE 3.3: Set of free list pages that contain deleted data.

3.2.2 Database header

The header contains the essential data that will be required to decode the database such as the page size, text encoding and journal types. The header is shown in figure 3.4. It's the first 100 bytes of the first page.

Offset	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	53	51	4C	69	74	65	20	66	6F	72	6D	61	74	20	33	00	SQLite format 3.
00000010	10	00	02	02	00	40	20	20	00	00	DC	79	00	00	7A	2A@ ..Ûy..z*
00000020	00	00	00	00	00	00	00	00	00	00	00	90	00	00	00	04
00000030	00	00	00	00	00	00	00	4B	00	00	00	01	00	00	00	01K.....
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	DC	79Ûy
00000060	00	2E	2C	50	05	00	00	00	05	0F	E7	00	00	00	6C	8C	..,P.....ç...l.
00000070	0F	FB	0F	F6	0F	F1	0F	EC	0F	E7	09	10	09	80	08	BD	.û.ö.ñ.ì.ç.....¼
00000080	08	3A	07	B3	06	98	07	6C	05	F8	04	EF	04	82	03	A7	..:'.s...l.ø.ï...\$
00000090	03	1F	01	FD	01	61	00	A4	01	2C	01	2C	00	00	00	00	...ý.a.▯.,,....
000000A0	00	00	00	00	81	05	17	07	17	21	21	01	81	55	74	61!!!..Uta
000000B0	62	6C	65	6D	65	64	69	61	5F	72	65	66	73	6D	65	64	blemedia_refsmed
000000C0	60	61	6B	73	65	66	73	61	43	50	45	41	54	45	30	54	lemedia_refsmed

FIGURE 3.4: The SQLite database header.

The autovacuum entry could be useful to determine if the deleted data could be recovered or no. This is determined in the offset 52 with 4 bytes length (highlighted in blue). In this WhatsApp database, the value at this offset is 0, which means that the WhatsApp does not support the autovacuum in his SQLite engine, makes it is possible to recover the deleted data.

Chapter 4

The Analysis Methodology

In order to accomplish this study, we performed a set of experiments. Each experiment is about different users interaction scenarios presented in details in the section 4.3 of this chapter.

The forensic data generated by WhatsApp is saved in the internal memory of the device. This is discussed in chapter 5. In that location, some data is located in inaccessible areas for the normal user, although it is accessible by rooting the phones or by using advanced commercial forensic tools. As well, this data is stored in an **encrypted** SQLite databases which also need suitable commercial tools to extract it [21] [22]. Unfortunately, these tools are expensive and we didn't have access to them.

To solve this problem, we decided to work with open-source tools and to use a powerful programming language such as Python and PowerShell that can help to achieve our goals. In addition, there is 3.4 million apps in the playstore that used SQLite databases to store their data and just 20 % of these apps are supported by these commercial tools.

Note that the understanding of how to deal with the data stored in SQLite databases manually using a free SQLite viewer is a necessity in digital forensics as the commercial tools do not provide any explanation of how the analysis is performed, nor they provide any guidance on how to correlate different pieces of evidence to completely reconstruct user activities. The tools and scripts used in this study are listed in section 4.1.

In fact, the access to these restricted locations in most of cases is not accessible without root. Rooting is the process allowing users of smartphones running the Android operating system to attain privileged control (known as "root access"). Basically rooting is performed to overcome the limitation that manufacture of Android phone put on some devices. If your Android mobile is rooted that means you are super user and you can attain this restricted location where the a part of WhatsApp data is saved that cannot be accessed on non-rooted Android.

In our work, we decided to extract the data from this location without rooting the phone. We did use unrooted Android devices for two reasons. First, the fail of the rooting process during the investigation means the re-installing of the OS which leads to overwrite the data preserved on the device. Also rooting can harm the phone with the potential risk of "Bricking". "Bricking" the device means screwing up the phone software so badly that the phone can no longer function and the data will be lost for ever. Second, the rooting becomes more and more difficult with the new versions of Android (7.0 and higher).

4.1 Requirements

1. Android smartphones:

Sony Xperia Z2 - Android 6.0.1

Sony Xperia L - Android 4.2.2

Huawei y7 prime - Android 8.0.0

2. Whatsapp messenger:

Version 2.19.53 (latest at the time of the experiments).

3. Forensic workstation (PC):

Toshiba Satellite C850-B907.

4. WhatsApp encryption key extractor:

PowerShell script written for this purpose.

5. WhatsApp Xtract:

Open source tool (python) to decode the encrypted databases backups using the key.

6. DB browser for SQLite:

SQLite DB viewer to view the data saved in the SQLite databases.

4.2 Workflow

The first step in the process of any application analysis is to identify the functionalities provided by this application. Based on these functionalities, the second step is to determine the set of experiments in order to generate the information relevant to the investigation.

The third step lies in locating the storage area on the device memory where the application stores these artifacts.

When the artifacts are located, they are analyzed and each experiment's artifacts may be correlated with the other to completely recreate the actions of the user.

The last step, which is one of the objective of the analysis of the SQLite databases, is to uninstall the WhatsApp application to see if the artifacts persist in the device even after the deletion of WhatsApp app.

The workflow of the analysis methodology is illustrated in figure 4.1.

4.3 Sets of experiments

4.3.1 Experiments concerning contacts

The goal from these experiments is to determine the list of contacts of the user as well as the operation done on it by the user. These experiments are listed in table 4.1.

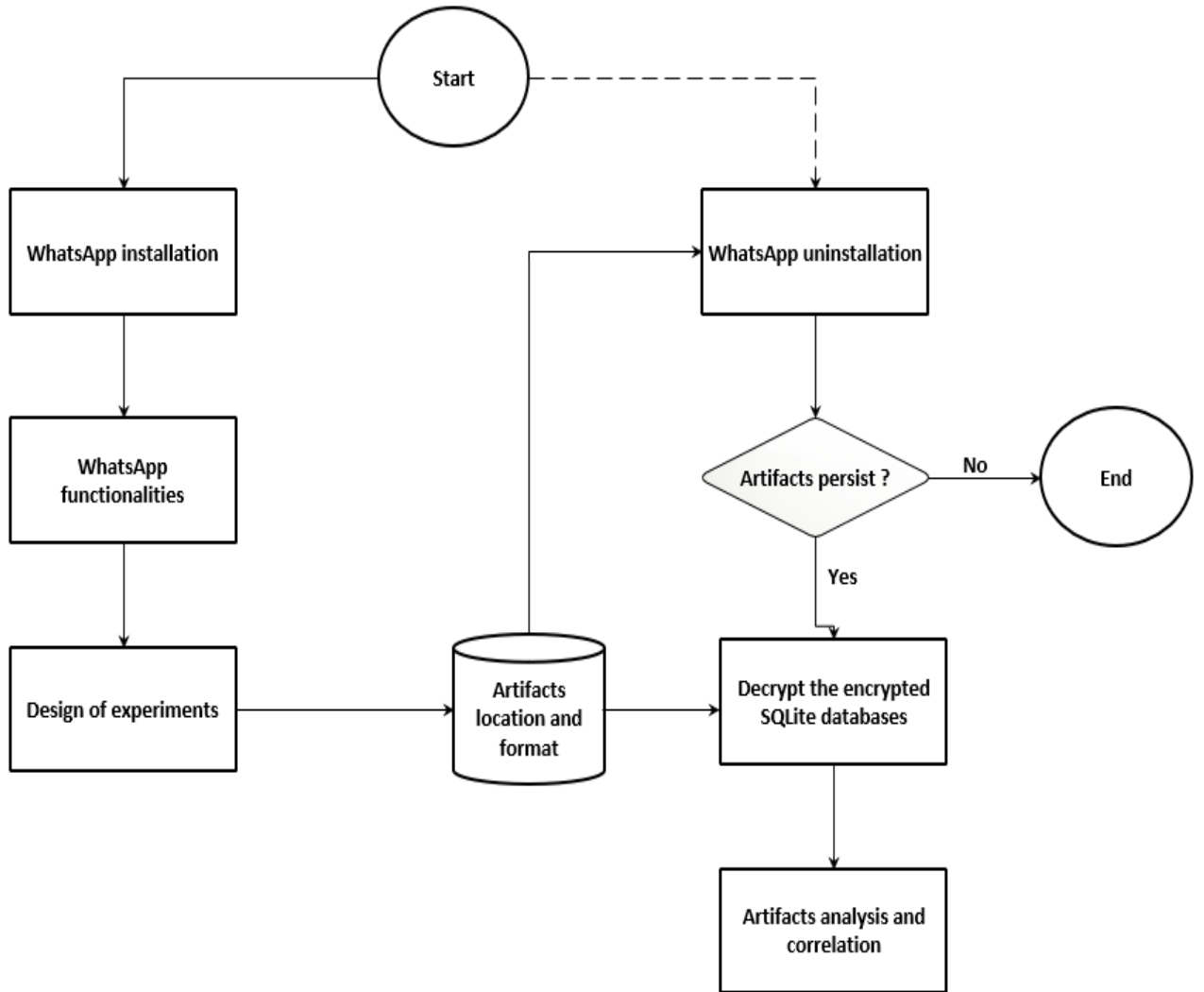


FIGURE 4.1: Workflow of the analysis methodology.

Contacts experiments	
Operation	Steps
Add contacts	User 1 adds User 2
Remove contacts	User 1 deletes User 2
Block contacts	1.User 1 blocks User 2 2.User 1 unblocks User 2

TABLE 4.1: The user contacts experiments. User 1 and User 2 are the WhatsApp users involved in the experiments.

4.3.2 Experiments concerning the private chat communication between user and contact

The goal is to reconstruct the history of messages exchanged as well as the contents of the textual and non textual messages between the user and each contact.

Messages experiments	
Operation	Steps
Textual messages exchange	1. User 1 and User 2 exchange messages
	2. User 1 and User 2 delete messages
Non-textual messages exchange	1. User 1 sends a picture to User 2
	2. User 1 sends a video to User 2
	3. User 1 sends an audio to User 2
	4. User 1 sends a contact to User 2
	5. User 1 sends a geolocation to User 2

TABLE 4.2: Experiments concerning all the types of messages exchanged privately.

4.3.3 Experiments concerning the messages state

The goal is to determine if the message has reached the server and if this message has been delivered and read by the recipient.

Messages State experiments	
Operation	Steps
Sending message, receiver offline	1. User 1 sends a message to offline User 2
	2. User 2 replies when online
Sending message, sender offline	1. User 1 is offline
	2. User 1 sends message
	3. User 1 goes online

TABLE 4.3: Message state experiments.

4.3.4 Experiments concerning the broadcast and group messages

The goal is to determine the users involved in a broadcast message and to reconstruct the chronology of a group chat as well as the events that happened within.

Broadcast and group experiments	
Operation	Steps
Broadcast messages	1. User 1 sends a broadcast message to more than two saved contacts
	2. User 1 sends a broadcast to unsaved contacts
Group messages	1. User 1 creates a group while add User 2 and User 4
	2. User 1 adds User3
	3. User 1 removes User 2, then User3 and User 4

TABLE 4.4: The broadcast and group messages experiments.

4.3.5 Experiments concerning voice and video calls

The goal is to reconstruct the chronology of the incoming and the outgoing calls of the user.

Voice and video calls experiments	
Operation	Steps
Performed voice/video call	1. User 1 calls User 2
	2. User 2 answers
	3. User 1 hangs up
Missed voice/video call	1. User 1 calls User 2
	2. User 2 does not answer
Refused voice/video call	1. User 1 calls User 2
	2. User 2 terminates the call without answer

TABLE 4.5: WhatsApp voice and video calls experiments.

Chapter 5

Forensic Analysis of WhatsApp Messenger

In this chapter, we discuss in the first section the functionalities of WhatsApp. This analysis is crucial to determine which types of experiments we should carry out in order to determine the useful artifacts from a forensic point of view.

Based on these experiments (chapter 4), we address in section 5.2 the location of the artifacts in the internal memory of the device.

In this study, we focus on the artifacts stored in SQLite databases. In order to secure and protect users data, these databases are encrypted with the strong AES encryption using cipher key with the size of 256-bit. In the section 5.3, we discuss the decryption of these databases on the unrooted Android phones.

In the following sections, we describe how the artifacts generated in the databases are decoded and analyzed. We discuss how to reconstruct the list of contact (section 5.4), how to extract the contents of the messages (section 5.5) as well as how to reconstruct the chronology of the voice and video calls (section 5.7). In addition, we discuss the broadcast and group communication (section 5.6), the status feature (section 5.8), and the deleted data (section 5.9).

5.1 Analysis of WhatsApp functionalities

As discussed in chapter 4, the first step in the mobile apps forensic analysis is to analyze the functionalities of the application in order to determine those that are important for the investigation process.

WhatsApp is an instant messaging application that allows users to exchange messages privately (one to one), in a broadcast message (one to many) or within a group of contacts (many-to-many). WhatsApp supports the textual and non textual data such as pictures, videos, audio, contacts card, and geolocation information. As well, WhatsApp enables users to place in voice and video calls privately or within a group. Recently, WhatsApp adds a new feature called status that allows you to share text, photos, videos and GIFs that disappear after 24 hours.

The functionalities that bring valuable information in the investigation are:

- **Contacts:** The communication between the users in WhatsApp is done by using the phone number. Each contact is associated with an ID, phone number and a profile picture.

The importance of the contacts list is critical in an investigation, as it provides the investigator with the possibility to determine with whom the user was in contact. It also allows to reveal the real identity of each contact as we will see in the following sections.

- **Messages exchanges:** As discussed above, WhatsApp allows the user to send and receive all types of data as well as network communications. WhatsApp organizes the messages exchanged by the user into conversations, each one corresponding to a specific chat.

The importance of the identification of the contents of messages is clear, as it allows the investigator to determine what is the data carried by the message exchanged by the user and the contacts.

- **Voice and video calls:** Starting in 2016, WhatsApp introduces the ability to place in voice and video calls. In the latest update, this feature is also available between four or less members within a group.

The reconstruction of the calls logs, with whom and how long, may constitutes a very useful evidence.

5.2 Location and types of WhatsApp artifacts

As Shakur showed [15], the artifacts are generated in different locations under several types of file formats. This is listed in figure 5.1.

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

FIGURE 5.1: WhatsApp messenger artifacts.

In this thesis, we focus on the artifacts stored in the SQLite databases where almost all data can be found such as text messages, multimedia files (images, videos, and audios), contacts, geolocation information, and the contacts of the users.

All types of data is mainly stored in two SQLite databases. Each database contains different tables where the information is stored. The two databases are:

1. Chat database **msgstore.db** and its backups.

- The main database **msgstore.db.crypt12** is stored in the internal memory **/data/data/com.whatsapp/databases**. This location is not accessed with no root privilege.
- its backups **msgstore-*<date>*.crypt12** are stored in the internal memory in **whatsapp/ databases**. This location is accessed with no root privilege.

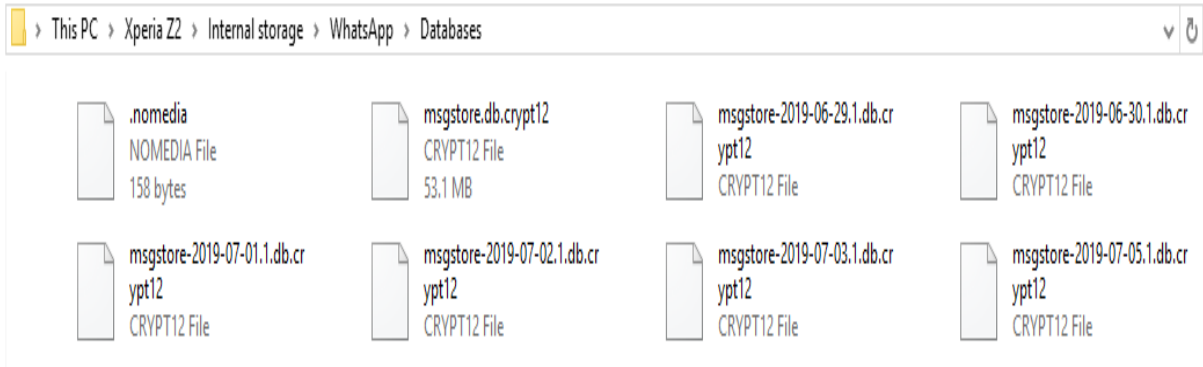


FIGURE 5.2: The encrypted databases backups of msgstore.db stored in the internal memory.

2. Contacts database **wa.db** stored at **/data/data/com.whatsapp/databases**. This file is not accessible on the unrooted devices.

After this, we uninstalled the WhatsApp application and we noted that these Databases persist. So, we continue our work as planned in the workflow (figure 4.1) in chapter 4.

These databases are encrypted, thus they need to be decoded in order to extract data from them, otherwise it's not possible to open and read their contents.

The strength of this encryption makes it unbreakable using brute force technique since it uses 256-bit key AES encryption. The only way to decrypt it is to use the encryption key, which is stored uniquely in each device and cannot be accessed without root privilege. In this thesis, we will use a method to extract this key without a root. In the next section, we explain what is the encryption as well as the method we used to extract the key to decrypt these databases.

5.3 SQLite databases decryption

Before explaining the method used to decrypt these databases, we have first to explain what is the encryption. The encryption is the process of encoding a message or information using an encryption key or code in such a way that only authorized parties can access it. This is explained in figure 5.3. Its purpose is to ensure privacy by keeping the information hidden from anyone for whom it is not intended, even those who can see the encrypted data. Database encryption can generally be defined as a process that uses a key to transform data stored in a database into a scrambled text that is incomprehensible without first being decrypted. To view the original data, we need to have the key that was used in the encryption.

To decrypt the WhatsApp databases, this key, which is stored in the "root" location in the internal memory of the device, should be extracted.

This can be done by gaining a root access to the phone. Once the phone is rooted, the superuser rights allows to get the key from `/data/data/com.whatsapp/files/key`, but this is not our case.

The root process does not lead to data loss in itself as a process [23], but in a real-life investigation the root process is not a good practice where the risk of data loss is real if it is not performed correctly as we explained in chapter 4.

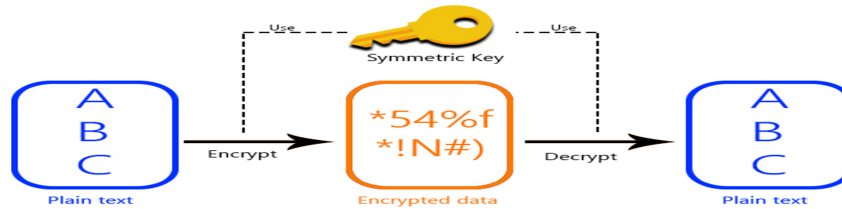


FIGURE 5.3: The encryption process.

The method used is performed by using mainly a PowerShell script. This is an open-source project published under the terms of the GNU General Public License (version 3) [24], which means that can be modified. Thus in this study, we re-write some functions in the script to be more suitable to our work. One of the most important variations we have done is that our script does not need internet connection. This is very important because it allows us to disconnect the phone from all networks prevent any remotely control of the phone. Second contribution is added support for Android API 28 (Android 9.0 Pie). The previous version support Android 7.0 Nougat.

The principle of this method consists on using the Android Debug Bridge (ADB). ADB is a command-line tool that lets us communicate with the Android device, so essentially we are running commands on the Android through the PC. The script will use these features to extract the key as well as the contact database "wa.db". Also, this script will backup the main databases in an encrypted database that will be extracted too.

FIGURE 5.4: The PowerShell script used to decrypt the databases.

This will create a directory on the PC named "extracted" that contains these DBs and the key (figure 5.6).

```
WhatsApp SQLite Databases Extractor by LOUAYE LAMAA

Installing legacy WhatsApp 2.11.431
2985 KB/s (18329558 bytes in 5.996s)
pkg: /data/local/tmp/LegacyWhatsApp.apk
Success
Install complete

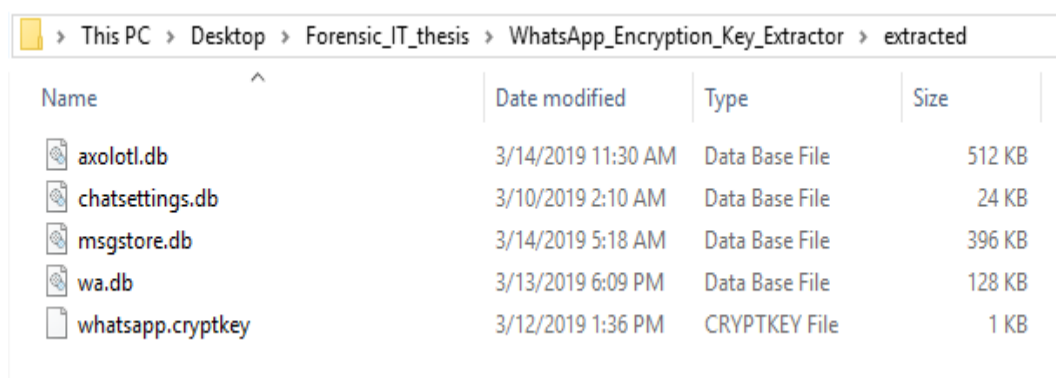
Now unlock your device and confirm the backup operation.

Please enter your backup password (leave blank for none) and press Enter:

apps/com.whatsapp/f/key
apps/com.whatsapp/db/msgstore.db
apps/com.whatsapp/db/wa.db
apps/com.whatsapp/db/axolotl.db
apps/com.whatsapp/db/chatsettings.db

Extracting whatsapp.cryptkey ...
Extracting msgstore.db ...
Extracting wa.db ...
Extracting axolotl.db ...
Extracting chatsettings.db ...
```

FIGURE 5.5: WhatsApp databases and key extraction.



Name	Date modified	Type	Size
axolotl.db	3/14/2019 11:30 AM	Data Base File	512 KB
chatsettings.db	3/10/2019 2:10 AM	Data Base File	24 KB
msgstore.db	3/14/2019 5:18 AM	Data Base File	396 KB
wa.db	3/13/2019 6:09 PM	Data Base File	128 KB
whatsapp.cryptkey	3/12/2019 1:36 PM	CRYPTKEY File	1 KB

FIGURE 5.6: DBs and the key are extracted from the phone.

Once the key is extracted, it can be used to decrypt msgstore.db.crypt12 or any crypt12 WhatsApp SQLite DB created on this device.

This is done by an open-source tool, **WhatsApp viewer** (figure 5.7 and 5.8). It is a very simple tool that associates the database with its key in order to decrypt it.

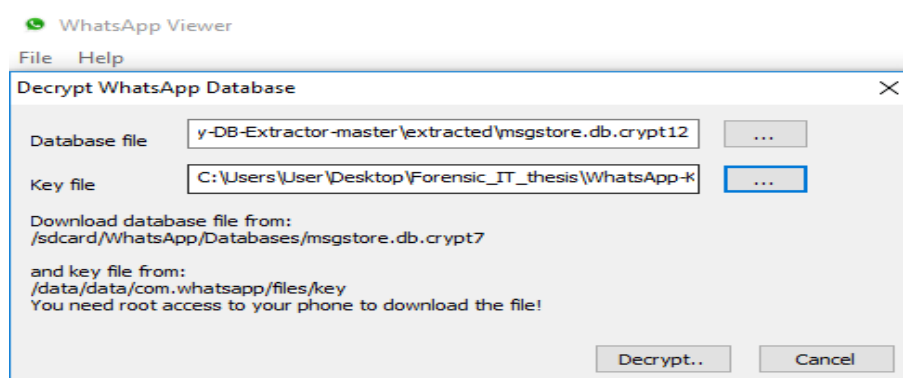
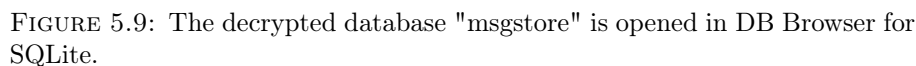
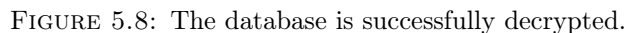


FIGURE 5.7: WhatsApp viewer tool.

Now, the decrypted databases can be manually parsed using a SQLite viewer such as DB Browser for SQLite (figure 5.9). The analysis of these databases will be addressed in details in the following sections.



SQLite is basically a highly reliable, embedded, and self contained SQL database engine. However, due to some errors, this database can be corrupted [25]. Frequently, the corruption results in this error: *Disk Image Is Malformed*.

To recover the database, by using the SQLite command-line tool, we run the following command sequence:

```
sqlite3.exe msgstore.db
.mode insert
.output msgstore_dump.sql
.dump
.exit
```

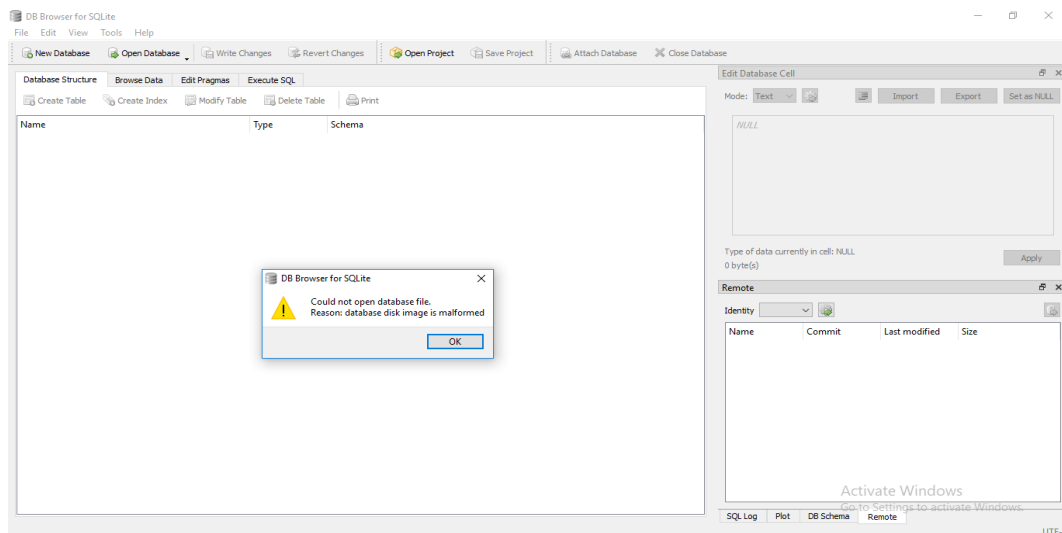


FIGURE 5.10: Damaged database: Disk image is malformed.

Now we have a SQL file (msgstore.db) with dumped database tables. We have to import the file to a new SQLite database.

In our case the database was heavily damaged, so we had to examine the file manually. We opened the SQL file in a text editor (sublime text) and save the tables of interest in separated SQL files.

We saved each table into a separated SQL file and used DB Browser for SQLite to export this SQL file into a SQLite database file.

In figure 5.11, we can just see the recovered table **messages** from the damaged database, compared to all tables as shown in figure 5.9.

For each table, this process should be respected in order to open the damaged tables of the database.

 The screenshot shows the DB Browser for SQLite application window with the "messages" table selected. The table data is displayed in a grid view. The columns are: id, status, needs_push, data, and timestamp. The data rows show various messages with their corresponding IDs and timestamps.

id	status	needs_push	data	timestamp
195283	0	0	Le2...	1548495397000
195284	0	0	NgI...	1548495409000
195285	0	0	...	1548495412000
195286	0	0	Tu...	1548495444000
195287	0	0	W...	1548495460000
195288	0	0	Enc...	1548495487000
195289	0	0	Eh...	1548495503000
195290	0	0	Fin...	1548495506000
195291	0	0	L2...	1548495514000
195292	0	0	Ez...	1548495515000
195293	0	0	Bt...	1548495519000
195294	0	0	W I...	1548495528000
195295	0	0	Hay...	1548495537000
195296	0	0	Ty...	1548495560000
195297	0	0	Y3...	1548495625000
195298	0	0	An...	1548495674000
195299	0	0	Hay...	1548495707000

FIGURE 5.11: Recovered table "messages" from the damaged database "msgstore".

5.4 Analysis of the contacts database "wa.db"

The analysis of the contacts list is very important to know with whom the user was interacting.

In this section, we determine the structure of the database and what is the information stored in its tables, then how to use this information to reconstruct the contact list and the

operations that was performed by the user. To achieve this, the experiments listed in table 4.1 were carried out.

5.4.1 The structure of the contacts database "wa.db"

The first step of the analysis of the user contacts is to study the structure of the **wa.db**. This database contains different tables. The valuable information concerning each contact is stored as a record mainly in one table, namely **wa_contacts**. This information is stored under several fields (column) based on the origin of the data (set by WhatsApp system or stored by the user in the phonebook). The structure of the table is described in table 5.1 and 5.2.

WhatsApp updated its tables by adding a new table that was not there before. We find that this new table can be useful as it contains valuable information about the blocked contacts. This table is called **wa_block_list** and it contains just one field.

Contact information from WhatsApp system	
Field name	Information presented
_id	The number of record set by SQLite
jid	WhatsApp ID of the contact containing his number
is_whatsapp_user	If the contact is an active WhatsApp user
unseen_msg_count	The number of unread message by the user
thumb_ts	Unix epoch time when the profile picture was set
wa_name	WhatsApp name of the contact set by himself

TABLE 5.1: The structure of the contacts database "wa.db" - table "wa_contacts" - information set by the WhatsApp system.

Contact information From Phonebook	
Field name	Information presented
number	The phone number of the contact
raw_contact_id	Record number
given_name	The contact name set by the user
family_name	The family name of the contact

TABLE 5.2: The structure of the contacts database "wa.db" - table "wa_contacts" - information set by the phonebook.

5.4.2 Reconstruction of the contacts list

To reconstruct the contacts list, we have to analyze the different values stored in the fields of the tables 5.1 and 5.2.

The figures 5.12 and 5.13 show the database wa.db opened in *DB browser for SQLite*. We just show two records for demonstration. .

In figure 5.12, each contact is associated with a WhatsApp ID which is stored under the **"jid"** field, with the structure *"number@whatsapp.net"*, where the *number* refers to the phone number of the contact (here 76680***). This field is used as a key to correlate this table to other tables in this database. This will be explained in the section 5.4.3. The contact is also associated with a boolean value stored under the field **"is_whatsapp_user"** indicating whether the contact is an active whatsapp user or not. The user is an active WhatsApp member if **"is_whatsapp_user = 1"**.

In addition, the full name of the contact, as it is saved in the phonebook of the user, is stored in the field **"display_name"**. The field **"given_name"** stores the name given by the user to the contact (here *Test*) and the field **"family_name"** stores the family name of the contact (here *Phone*). The field **"wa_name"** stores the name of the contact set by the contact himself (here *Louay*).

Furthermore, the "About" and its set time (previously known as status) is stored in the field **"status"** and **"status_timestamp"** respectively.

The avatar picture can link the user to his real identity if the picture displays his face or a location. The avatar picture of a contact is stored in media/pictures folder but we did not find the thumbnail of the profile picture stored in the database. The timestamp (Unix epoch time - 10 digits) stored in the field **"thumb_ts"** indicates when the contact has set its current avatar. The timestamp (13 digits) of the field **"photo_id_timestamp"** indicates the time the user opened the profile picture of the contact.

As this table stores the information about individual contacts, it stores also the information about the groups that the user has joined. In this case, the WhatsApp ID is structured under a different string such as the string *"9617029***-1555701066@g.us"* (figure 5.13). The phone number is that of the group creator and the UNIX epoch time is the group creation time. The name of the group, here is *forensic IT test*, is stored under the **"display_name"**. Note that the members of the groups are stored in this table but it is not possible to associate each contact to the corresponding group.

Table: **wa_contacts**

	jid	is_whatsapp_user	status	status_timestamp	number	display_name
	Filter	Filter	Filter	Filter	Filter	Filter
1	961766@s.whatsapp.net	1	New user!!	1553164911000	+961766	Test Phone

Table: **wa_contacts**

	thumb_ts	photo_id_timestamp	given_name	family_name	wa_name
	Filter	Filter	Filter	Filter	Filter
1	1562236228	1562357720787	Test	Phone	Louay

FIGURE 5.12: "wa_contacts" table - the individual contacts records.

Table: **wa_contacts**

	jid	is_whatsapp_user	status	status_timestamp	number	raw_contact_id	display_name
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	9617029-1555701066@g.us	1	NULL	0	NULL	NULL	Forensic IT test

FIGURE 5.13: "wa_contacts" table - the group contacts records.

5.4.3 Blocked contacts

WhatsApp allows user to block a contact, which prevents any type of communication between the user and the contact as well as getting any update of the profile picture and the about (status).

When the user blocks a contact, WhatsApp adds a record to the table **wa_block_list** of the contact database *wa.db*. This table contains one field called **"jid"**, that stores the ID (phone number) of the blocked contacts. An example is shown in figure 5.14.

Table: **wa_block_list**

	jid
1	9617089 .
2	961711 ...

FIGURE 5.14: "wa_block_list" table - the blocked contacts records.

Therefore, the other information of the blocked users can be deduced by selecting those records in table *"wa_contacts"* using the following SQL query:

```
SELECT * FROM wa_contacts WHERE jid IN (SELECT *jid* FROM wa_block_list)
```

Using this query, we can select all the information related to the blocked user in the table *wa_contacts*, or we can filter out the data needed as shown in figure 5.15.

The screenshot shows a database application interface with tabs for Database Structure, Browse Data, Edit Pragmas, and Execute SQL. Below the tabs are icons for file operations and navigation. The SQL editor shows a query that filters contacts from the *wa_contacts* table based on the *wa_block_list* table. The results table below the query shows one record for a blocked contact.

	jid	is_whatsapp_user	status	status_timestamp	display_name	given_name	family_name	wa_name
1	9617089 .	1	Hey there! I am using W...	1555185418000	Blocked contact	Blocked	contact	nbb16094

FIGURE 5.15: Extracting the blocked contact information using a SQL query.

The results of our analysis show also that when a contact is unblocked, the corresponding record in table *"wa_block_list"* is deleted. Furthermore, it is not possible to know when a contact was blocked or how many times he has been blocked or unblocked.

As a final consideration, we note that no information is stored on the side of the contact that gets blocked, so it is not possible to tell whether the user of the device under analysis

has been blocked or not by anyone of their contacts.

5.5 Analysis of the chat database "msgstore.db"

The chat database **msgstore** is the file where all messages, received and sent, are stored. This database has a very evidentiary value. Its analysis reveals evidence about the content of the messages, the time the messages have been sent or received, what type of communication the user has been involved in.

In this section, we describe the structure of this database to determine what type of information is stored in order to reconstruct the chat history, the content of the messages exchanged as well as the status of these messages. Furthermore, we discuss the case of one-many/to-many messages such as broadcast or group communication. In the last part, we discuss the deleted messages.

5.5.1 The structure of the chat database "msgstore"

The chat database contains different tables, the most important table that has evidentiary value concerning the messages exchanged is:

messages: where every message and its details are stored as a record.

The data is stored in multiple fields, which can be classified into two categories, one is the characteristics of the messages listed in table 5.3 and two is the content of the messages listed in table 5.4.

In the following sections, we discuss how to correlate the values stored in these fields to the actions taken by the user.

Messages characteristics	
Field name	Information stored
_id	Record number set by SQLite
key_remote_jid	WhatsApp ID of the contact
key_id	Message identifier
key_from_me	Message sender
status	The status of message (delivered or not)
timestamp	Time of message sending (Unix epoch format from the user device clock)
received_timestamp	Time of message receiving
receipt_server_timestamp	Time of message when reached the server
receipt_device_timestamp	Time of delivery to the contact
needs_push	Broadcast message
recipient_count	Number of recipient in a broadcast message
remote_resource	Group message characteristics

TABLE 5.3: The structure of the chat database "msgstore.db" - table "messages" - message characteristics.

Messages content	
Field name	Information stored
media_wa_type	Message type (text, media...)
data	Message content when message is a text
media_mime_type	Exact type of the media message
media_hash	Hash of the media message
media_url	URL of the media message
media_size	Size of media message
media_name	Name of the media file
media_duration	Time in sec of a media file (video, audio)
latitude	Latitude of the message (location)
longitude	Longitude of the message (location)

TABLE 5.4: The structure of the chat database "msgstore.db" - table "messages" - message content.

5.5.2 Determination of the chat history

To determine how, when, and with whom the conversation had started, we should decode the fields of the table "messages" presented in figure 5.16.

key_remote_jid	key_from_me	key_id	status	needs_push	data	timestamp
Filter	Filter	Filter	Filter	Filter	Filter	Filter
96170298	1	F9D5F1222A8...	13	0	How are you	1552408820301
96170298	0	FD661B7BE4C...	0	0	I am good	1552409050000
96170298	1	8B3E2A1519C...	13	0	NULL	1552409393603
96170298	0	BF182566136...	0	0	Oh my god	1552409419000
96170298	0	2F2B0B3CB37...	0	0	Do you know t...	1552409470000
96170298	0	C3D9208E292...	0	0	Where did yo...	1552409516000
96170298	1	8AD3A6CB0F0...	13	0	I know, don't ...	1552409617453
96170298	0	C42776722C8...	0	0	Are you crazy...	1552409651000

FIGURE 5.16: Reconstruction of the chat history.

The records presented in the figure above show that the user had a conversation with a contact who have the number 96170298*** which is clear from the field "**Key_remote_jid**". This field indicates the phone number of the contact involved in the conversation.

The field "**key_from_me**" indicates the message direction. If "**Key_from_me = 0**", the user receives the message (incoming text) and if "**Key_from_me = 1**", the user sends the message (outgoing text).

In this case, by analyzing the first record, the user started the conversation (Key_from_me = 1) by sending a text message "*How are you*" stored in the "**data**" field, which contains the content of the textual messages.

The field "**timestamp**" indicates the time the message was sent and "**received_timestamp**" indicates the time the message was received (by the user).

Here, the message "*How are you*" was sent by the device owner (Key_from_me = 1) at Tuesday 12 march, 2019 4:40:20 PM ("**timestamp**") and the contact replied at the same day at 4:44:10 pm ("**received_timestamp**") with *I am good* in the second record.

Note that the time in these fields is stored as a UNIX epoch time.

As we can see, each message (record) is associated with a unique identifier under the field **"key_id"**. This field is used usually to correlate the information stored about a message in different tables.

Since the messages are stored in the databases according to their date from the older to the newest and not according to each conversation, it is important to the investigator to extract just the information related to the conversation between the user and a specific contact. This is done using the following the SQL query :

```
SELECT * FROM messages WHERE key_remote_jid = 'contact phone number'
```

5.5.3 Analysis of the messages content

WhatsApp is used to exchange all types of information, such as text, images, videos, audios, contacts, and locations.

The types of data is determined by looking at the field **"media_wa_type"**. If **"media_wa_type = 0"**, the messages exchanged are textual and then stored in the **"data"** field. Otherwise, the messages could be a multimedia file, a contact card or a location. To determine exactly what is the type of the non-textual message, we should look at the others fields that depend on the type of the messages exchanged.

Multimedia files

When the message exchanged is media file, a record is stored in the table *messages* under different fields. First, the type of the file is determined by the field **"wa_media_type"** if the message is not a text. For the images **"wa_media_type = 1"**, for the videos **"wa_media_type = 3"**, and for audio **"wa_media_type = 2"**. The field **"wa_mime_type"** indicates exactly what is the type of the media file (format). For example, if the file is an image, the value stored is JPEG or JPG (for video is MP4 and for audio is ogg). This is shown in figure 5.17.

	data	media_url	media_mime_type	media_wa_type	media_size	media_name	media_duration
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	<i>NULL</i>	https://mmg-fna.whatsapp.net/d/f/Ah...	image/jpeg	1	204279	30593e6f-e66b...	0
2	<i>NULL</i>	https://mmg-fna.whatsapp.net/d/f/Al...	video/mp4	3	5650286	75617819-7dc...	23
3	<i>NULL</i>	https://mmg.whatsapp.net/d/f/AsUjU...	audio/ogg; codecs...	2	55696	1ee54844f51d...	23

FIGURE 5.17: The three types of a multimedia message.

Here we explain the image messages. The other media types follow the same logic. The name of the file is stored in **"media_name"** column and its size in bytes in **"media_size"**. The url, which corresponds to its location in the server (temporarily storage), is stored in **"media_url"** and the hash of the file is stored in **"media_hash"** field. These fields are shown in figure 5.18.

In the recipient side, these fields are the same except the **"media_name"**, which is empty.

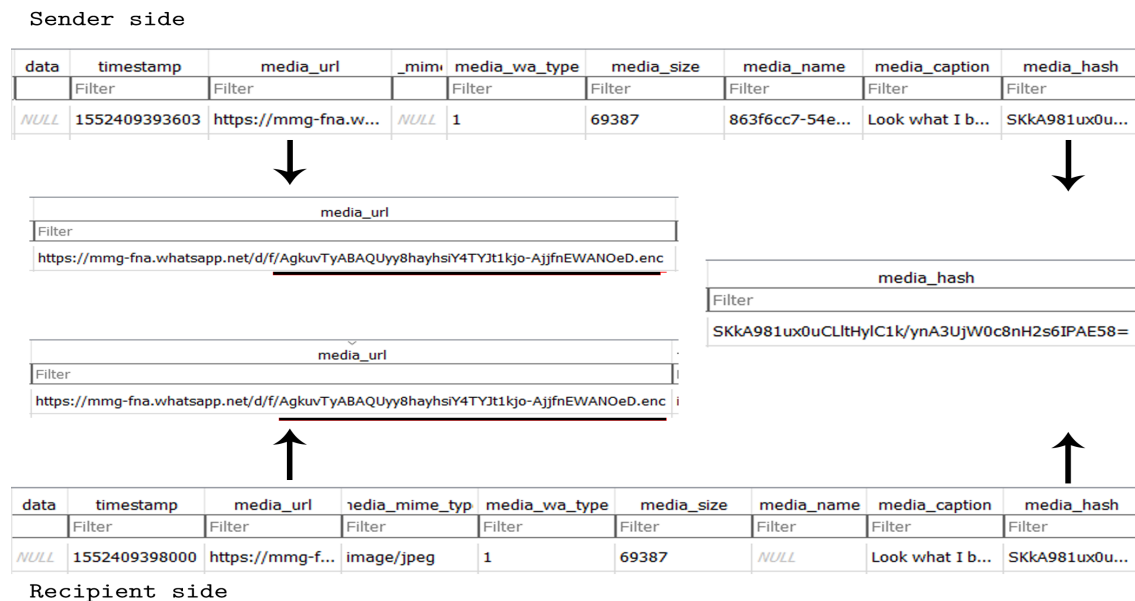


FIGURE 5.18: The image message content: the sender and the recipient records.

As we can see, the table *messages* lacks information about the thumbnail of the multimedia messages. As SQLite is a RDBMS, this table is related to another tables in this database. So, combining data from different tables is useful like adding, here, the thumbnail of the message to the table *messages*. To do this, the best way is to use the SQL queries (SQL Join). Here, we use the LEFT JOIN operation which returns all rows from the left table matched against each matching row from the right table. If there is no matching row in the right table, then the result from the right table will be NULL. The Venn diagram of this operation in figure 5.19 shows how the left join works as all left table (A) records are preserved and the matched rows from the right table (B) are extracted if the condition is satisfied.

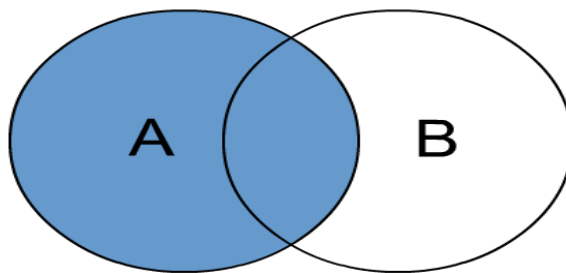


FIGURE 5.19: LEFT JOIN Venn diagram.

To decode the thumbnail of an exchanged image, we must analyze the table, the "**message_thumbnails**" which contains a field called "**thumbnail**" that stores the media thumbnails. We run the following LEFT JOIN SQL query to extract the thumbnail of a specific message by adding WHERE clause. To understand this query, the figure 5.20 explains the relation between these two tables. The field **key_id** is used to join the tables (the expression after the ON keyword in the query). We use the LEFT JOIN with *messages* table as the left table

(table to the left of the ON keyword) because we want every row from that table matched any matching row in the *message_thumbnails* table.

```
SELECT messages.key_remote_jid, message_thumbnails.thumbnail
FROM messages
LEFT JOIN message_thumbnails ON messages.key_id = message_thumbnails.key_id
WHERE message_thumbnails.key_id = 'Key_id of the image message stored under field key_id'
```

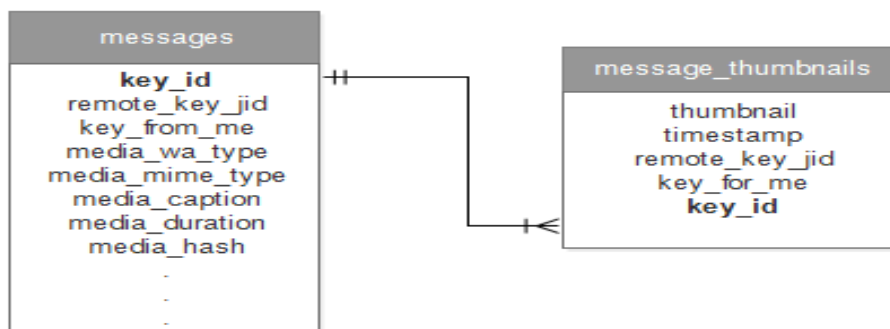


FIGURE 5.20: LEFT JOIN operation between *messages* and *message_thumbnails* tables.

This is shown in figure 5.21.

Database Structure Browse Data Edit Pragma Execute SQL

SQL 1 SQL 1

```
1 SELECT messages.key_remote_jid, message_thumbnails.thumbnail
2 FROM messages
3 LEFT JOIN message_thumbnails ON messages.key_id = message_thumbnails.key_id
4 WHERE message_thumbnails.key_id = 'F178B41A5CA9A1169A90295A1B070E21'
```

key_remote_jid	thumbnail
1 9617668 @s.whatsapp....	BLOB

Mode: Image

Type of data currently in cell: JPEG Image
90x89 pixel(s), 2.48 KiB

Remote

Identity

FIGURE 5.21: "message_thumbnails" table - the thumbnail of an image message extracted using LEFT JOIN query.

Contact cards

WhatsApp enables users to exchange contacts cards from the phonebook of the sender. In this case, the **"media_wa_type"** is "4". The messages are sent in VCARDS format and they are stored in the **"data"** field.

The number of the contact exchanged is stored in this field, here 7063****. The name of this contact (as it's saved in the phonebook) is stored in **"media_name"** field. The others fields have the same meaning and value as other type of data.

In the recipient side, all the fields are the same except the **"from_me"** is "0". The figure 5.22 shows a contact card from the sender side.

data	timestamp	media_url	media_mime_type	media_wa_type	media_size	media_name
Filter	Filter	Filter	Filter	Filter	Filter	Filter
BEGIN:VCARD...	1556752884990	NULL	NULL	4	0	Louay

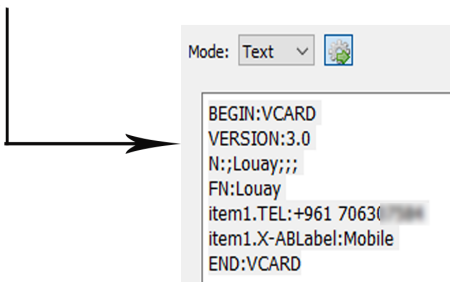


FIGURE 5.22: The contact card message.

Geolocation coordinates

WhatsApp provides the users with the ability to send their actual location or any other location on the map. The type of data in this case is marked by "5" in the field **"media_wa_type"**. An example of such record is shown in figure 5.23.

These information is stored in **"latitude"** and **"longitude"** columns. A thumbnail is stored in the other table **message_thumbnails**. To extract this thumbanil, we can use the same SQL query used to extract the image message thumbnail by replacing the key_id. This is shown in figure 5.24.

Table: messages						
key_remote_jid	key_from_me	key_id	timestamp	media_wa_type	latitude	longitude
Filter	Filter	Filter	Filter	Filter	Filter	Filter
1 9617668...	1	46848C3D40F...	1552411323149	5	33.804919862...	35.506647914...

FIGURE 5.23: The geolocation message.



FIGURE 5.24: "message_thumbnails" table - the thumbnail of a location message.

Attachments

WhatsApp allows the exchange of various types of attachments such as documents and apk.

The type of this data is indicated in the field **"media_wa_type"** where the value is "9" and the exact type of the attachment is indicated in the field **"media_mime_type"**. The figure 5.25 shows the records of different types of attachments.

As we can see in these records, the information about the content of the attachment is spread over several fields. The name (with the extension of the file) and the caption are stored in **"media_name"** and **"media_caption"** fields respectively. Other information are stored in fields that are not showed in the figure 5.25 such as the number of page for the pdf and word documents which is stored in **"media_duration"** field.

Table: messages

New Record | Delete Record

	key_id	data	media_mime_type	media_wa_type	media_name	media_caption
	Filter		Filter	Filter	Filter	Filter
1	85F1926D061A9...	NULL	application/pdf	9	Pentest.pdf	Pentest
2	FD18B79E2343A...	NULL	application/rtf	9	NULL	toxico analysis
3	5ED459401E61D...	NULL	application/msword	9	Bioweapons....	Bioweapons
4	5B45ABDB23121...	NULL	application/vnd.openxmlformats-officedocument.presentationml.presentation	9	NULL	Forensic_IT
5	3EB0060C5A2F1...	NULL	application/vnd.android.package-archive	9	zAnti3.18.apk	zAnti3.18.apk
6	66C475E03DDB...	NULL	application/vnd.openxmlformats-officedocument.spreadsheetml.sheet	9	Forensic_na...	Forensic_name..
7	3EB0767C4AAB...	NULL	text/plain	9	topics.txt	topics.txt
8	491C57A294358...	NULL	application/octet-stream	9	NULL	msgstore.db.cr...

FIGURE 5.25: The attachments messages records.

The thumbnail of the an exchanged document is its first page. This thumbnail is stored in *"message_thumbnails"*. By using the left join SQL operation, we can extract the thumbnail of the message with its information in the table *"messages"*.

```
SELECT messages.key_remote_jid, messages.key_from_me, messages.media_wa_type,
messages.media_mime_type,
message_thumbnails.thumbnail
FROM messages
LEFT JOIN message_thumbnails ON messages.key_id = message_thumbnails.key_id
WHERE message_thumbnails.key_id = 'Key_id of the document message'
```

This SQL query is illustrated in figure 5.26.

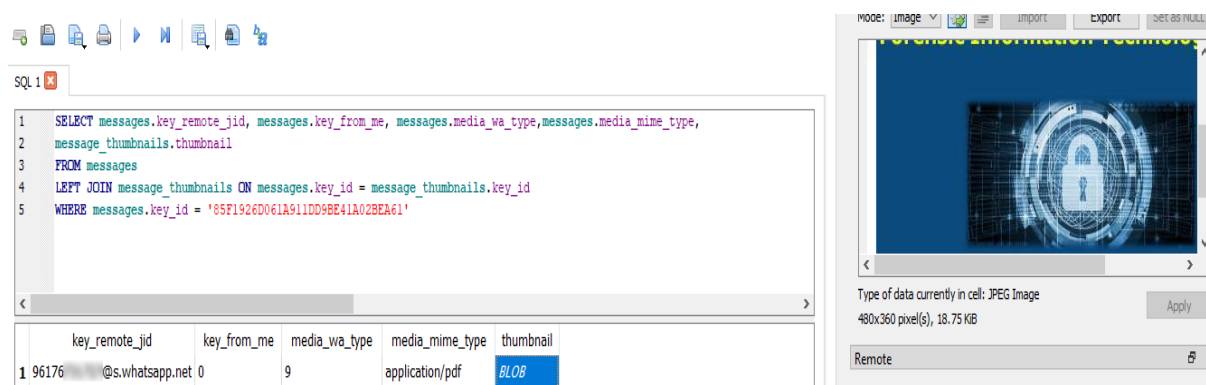


FIGURE 5.26: "message_thumbnails" - the thumbnail of the document message extracted using LEFT JOIN.

5.5.4 The determination of the messages state

Messages are not exchanged directly among communicating users, but they are first sent to the central server, that forwards them to the respective recipients if they are on-line. Otherwise, it stores them in the local server until they can be delivered. When a message is stored in the sender database, it has not necessarily been delivered to the contact. In fact, there is four possible states.

- (i) The message is sent by the user but it is not transmitted to the server (the clock sign).
- (ii) The message is sent by the user to the server but still not delivered to its recipient (one gray tick).
- (iii) The message is delivered to its recipient (two gray ticks)
- (iv) The message is read by the recipient (two blue ticks)

The analysis of the message state is very important during investigation to know if the message was delivered or not to its recipients.

To reveal this information, several fields in the "messages" table of the sender database must be analyzed.

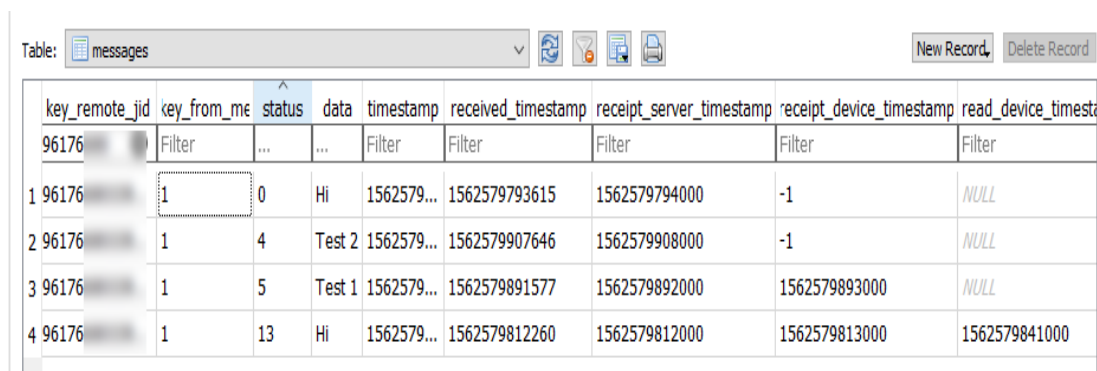
The first field is **"status"**, which will indicates whether the message has reached the server. If the message is sent but still not transmitted to the server, a record is stored in the database. In this case, the **status = 0** given that **"key_from_me = 1"**. (**status** is always zero when **key_from_me = 0**).

The field **"timestamp"** indicates the time the message was sent by the user. This is shown in the first record in figure 5.27.

If the message is transmitted to the server but it is not delivered to the recipients, the **"status = 4"** and the time when the message reaches the server is stored in **"receipt_server_timestamp"**. This is shown in the second record in figure 5.27.

When the message is delivered to its destination, the **"status = 5"** and the time the recipient receives the message is stored in the field **"receipt_device_timestamp"**. When the message is read by that recipient, the **"status = 13"** and the time the recipient reads the message is stored in **"read_device_timestamp"**. This is shown in the third and fourth records respectively in figure 5.27.

In the case of an audio message, the time the audio is heard by the contact is stored in **"played_device_timestamp"** and not in **"read_device_timestamp"**.



	key_remote_jid	key_from_me	status	data	timestamp	received_timestamp	receipt_server_timestamp	receipt_device_timestamp	read_device_timestamp
1	96176	1	0	Hi	1562579...	1562579793615	1562579794000	-1	NULL
2	96176	1	4	Test 2	1562579...	1562579907646	1562579908000	-1	NULL
3	96176	1	5	Test 1	1562579...	1562579891577	1562579892000	1562579893000	NULL
4	96176	1	13	Hi	1562579...	1562579812260	1562579812000	1562579813000	1562579841000

FIGURE 5.27: The possible states of a message.

By analyzing these fields together, the state of the messages can be revealed.

5.6 Multiple message destinations

5.6.1 Broadcast messages

WhatsApp enables users to send the same message to multiple contacts at the same time privately and the contacts' reply is shown just for the sender.

When the user sends a broadcast, a record is generated in the table *messages* for each recipient. All these records (messages) have the same identifier in the field **"key_id"**, which helps to identify the nature of the message as a broadcast (figure 5.28).

The phone number of the recipients and their IDs are stored in **"key_remote_jid"**. The user WhatsApp ID is marked with the word *broadcast*. The field **"recipient_count"** identifies the number of the recipient involved in this broadcast message. The field **"needs_push"** is always "2" in a broadcast message. The indication of this field is unknown.

On the recipient side, the received broadcast message is saved in just one record in *messages* table. This record is distinguished from other records by the presence of the %~ sign in the field **"key_id"**. This is shown in figure 5.29.

In the case of a broadcast message sent for a non saved contact, it will not be delivered to the recipient. The sender database structure would be identical to the one showed in section 5.5.4 (ii). On the receiver side, no record is generated.

Our experiment shows that if the recipient replies to the message, the record generated on both sides would not be distinguished from any other private message, and this is because the reply is sent just to the original sender as a regular private message between the user and the contact.

	key_remote_jid	key_from_me	key_id	needs_push	data	recipient_count
	Filter	Filter	Filter	Filter	F...	Filter
1	9617163...@s.what...	1	67E3364AEAAD4B154278CBD4AA2DE1FB	2	This i...	4
2	9617079...@s.what...	1	67E3364AEAAD4B154278CBD4AA2DE1FB	2	This i...	4
3	9617668...@s.what...	1	67E3364AEAAD4B154278CBD4AA2DE1FB	2	This i...	4
4	961355...@s.whats...	1	67E3364AEAAD4B154278CBD4AA2DE1FB	2	This i...	4
5	1555701619@broadcast	1	67E3364AEAAD4B154278CBD4AA2DE1FB	2	This i...	4

FIGURE 5.28: The broadcast messages records.

Table: messages					
	key_remote_jid	key_from_me	key_id	data	
	Filter	Filter	Filter	Filter	
1	9617029...	0	%~67E3364AEAAD4B154278CBD4AA2DE1FB	This is a broa...	

FIGURE 5.29: The broadcast message - recipient side.

5.6.2 Group chat

WhatsApp allows another type of chat communication, where messages are sent within a group of members and every message is shown to all of them.

As the other types, a message sent within a group is stored as a record in the table *messages*.

The analysis of this case requires the study of different fields to investigate the events that happened within the group.

To do this analysis, we created a group of four members (including the group creator). The textual messages sent contain the name of each user namely user1, user2, user3 and user4.

The first field that should be analyzed is "**key_remote_jid**" because this would give the information about the creator of the group, the creation time and the group ID. As shown in figure 5.30, this field contains, in all records, the creator's phone number and the creation time of the group as in the following string *9617029****-1555701066@g.us*. The time format is the UNIX epoch and means that the group was created at *Friday, April 19, 2019 10:11:06 PM* by the user who have the number 9617029****.

The name of this group is stored in the field "**data**", here *Forensic IT test*, where the field "**media_size = 11**" in record no.1 referring to the action of creation of the group.

The record no.2 corresponds to the message sent by the creator. On the other hand, when a member of the group sends a message, his number is stored in the field "**remote_resource**" and not in the field "key_remote_id" because this last field is always related to the admin. This is the case in record no.3 and no.4, where user2 and user3 have the numbers 9617668**** and 9617163**** respectively.

The record no.5 is the action of adding a user by the admin. To identify this action, we look at the field "**media_size**" which in this case is "12".

The record no.6 is analyzed in the same way for records no.3 and no.4. Here, the user4 have the number 9617079****.

Note that there is no such a record like record no.5 ($\text{media_size} = 12$) for the both user2 and user3, meaning that these two members were added in the same action of the group creation.

These records lack some information that are important for the investigation process. For example, no record tracks the identity of the group members that receive a specific message at any point in time. Although this information is not stored explicitly in the database, it can be deduced by examining the fields that store when a member is added and when a member leaves. This field is "**media_size**" which is the same field that stores the creation of group. We observe from the previous points that when media_size equals "11", it represents the creation of the group while when it is "12", it indicates adding a member. This field also stores a value when a member leaves/is removed.

To clarify, we did an experiment in which the users mentioned above left the group in order to reconstruct the chronology of the group composition.

What we can conclude from the figure 5.30 that the user1 creates the group, named Forensic IT test, on *Friday, April 19, 2019 at 10:11:06 PM* and he adds in the same time user2 (9617668****) and user3 (9617163****), then adds the user4 (9617079****) on the same day at 10:15:31 PM. Note that this applies to all the users databases including the group creator.

	key_remote_jid	key_from_me	status	data	timestamp	media_size	remote_resource
	Filter	Filter	F...	Filter	Filter	Filter	Filter
1	9617029-1555701066@g.us	1	6	Forensic IT test	1555701066000	11	9617029-1555701066@g.us
2	9617029-1555701066@g.us	1	13	This is user1	1555701120224	0	NULL
3	9617029-1555701066@g.us	0	0	This is user2	1555701166000	0	9617668-1555701166@g.us
4	9617029-1555701066@g.us	0	0	This is user3	1555701179000	0	9617163-1555701179@g.us
5	9617029-1555701066@g.us	1	6	NULL	1555701331000	12	9617029-1555701331@g.us
6	9617029-1555701066@g.us	0	0	This is user4	1555701386000	0	9617079-1555701386@g.us

FIGURE 5.30: The group chat records.

After that, user3 left the group. This action, in record no.1 in the figure 5.31, is stored in the database under the field "**media_size**" with value "5" indicating leaving the group. The identity of the user is reported in the field "**remote_resource**". This user left on *Wednesday, May 1, 2019 9:27:28 AM* (field timestamp). Then, by the same analysis, we know that user4 and user2 left the group on *Wednesday, May 1, 2019 4:59:27 PM* and *Thursday, May 2, 2019 2:05:00 AM* respectively.

So, this will help the reconstruction of the composition of the group and then whether a user was in the group during the conversation or not. (figure 5.32).

	key_remote_jid	key_from_me	status	data	timestamp	media_size	remote_resource
	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	9617029 -1555701066@g.us	1	6	NULL	1556692048000	5	9617163 @s.whatsapp.net
2	9617029 -1555701066@g.us	1	6	NULL	1556719167000	5	9617079 @s.whatsapp.net
3	9617029 -1555701066@g.us	1	6	NULL	1556751900000	5	9617668 @s.whatsapp.net

FIGURE 5.31: The group records created when a member leaves the group.

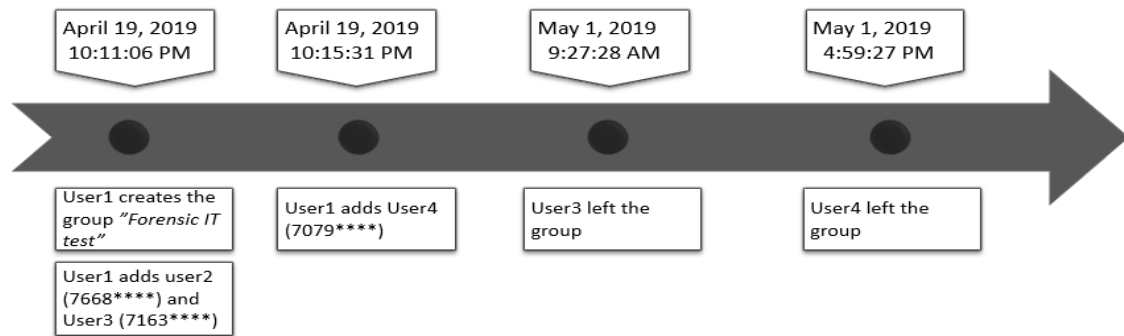


FIGURE 5.32: Timeline of the chronology of the group composition.

5.7 Voice and call logs

WhatsApp allows users to place voice and videos calls through the mobile broadband network or the Wi-Fi. In fact, when a call is performed, whatever were its results, a record will be stored in the main database "msgstore.db" but normally not in the *messages* table. Such records are saved in the table **call_logs**.

As shown in figure 5.33, WhatsApp stores in this table the following information about the calls: the call identifier in the field "**jid_row_id**", its type "**video_call**", its direction (outgoing or incoming call) in "**from_me**", its time in "**timestamp**", its duration in "**duration**", its size in "**bytes_transferred**", its result in "**call_result**" (successful, missed or refused).

Based on the experiments of the table 4.5, we have distinct records corresponding to the three possibilities presented in that table.

By looking at the field "**from_me**", its value is equal to 1 in the two records 929 and 930 which means that these two records correspond to an outgoing calls from the user to the contact who have the ID 271 ("**jid_row_id**"). The first call is a voice call because "**video_call** = 0" and the second record is a video call since "**video_call** = 1".

The field "**call_result** = 5" means that the two calls were successful, the voice call was established at 21 April 2019 1:45:30 AM (**timestamp**) and lasts for 49 seconds (**duration**). and the second was done at 21 April 2019 1:17:34 AM and last 80 seconds.

The next two records (_id 931 and 932) have the field "**from_me** = 0", and "**call_result** = 5", which means that these two calls were successful incoming calls (from the contact to the user).

The next four records from _id 933 to 936 have all the field "**call_result** = 4" and the "**duration** = 0", which means that these four records correspond to missed calls. The

analysis of other fields (timestamp and from_me) is the same as above.

The next record also shows that the call did not happen (**duration** = 0 sec), but the field "**call_result**" shows the value of "2", which means that the call was terminated by the user if the call is incoming (**from_me** = 0), or by the contact if the call is outgoing (**from_me** = 1).

As we can see, there is no field that stores the phone number of the contact, but we can extract this information from another table. The identifier of the contact involved in the call stored in the field **jid_row_id**, is stored in another table called **jid** under the field "**_id**". This table also stored the phone number of the contact under the field "**user**". To extract this phone number, we use the following SQL query to extract this information from "jid" table (figure 5.34).

```
SELECT user FROM jid WHERE _id = *user_id_number*
```

The table 5.5 resumes the chronology of the voice call logs presented in figure 5.33.

Table: call_log

	_id	jid_row_id	from_me	call_id	timestamp	video_call	duration	call_result
	Filter...	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	929	271	1	call:65E6F2F9...	1555798530353	0	49	5
2	930	271	1	call:51E57CA...	1555798623962	1	80	5
3	931	271	0	call:172F5730...	1555798801000	0	55	5
4	932	271	0	call:7233121A...	1555798894000	1	49	5
5	933	271	1	call:3806A667...	1555799064811	0	0	4
6	934	271	1	call:6136D350...	1555799116958	1	0	4
7	935	271	0	call:D56706EF...	1555799185000	0	0	4
8	936	271	0	call:977A2559...	1555799196000	1	0	4
9	937	271	1	call:1D830F10...	1555799236478	0	0	2
10	938	271	1	call:0E83FD25...	1555799293409	1	0	2
11	939	271	0	call:4B70977D...	1555799360000	0	0	2
12	940	271	0	call:92D8A7D...	1555799457000	1	0	2
13	941	271	1	call:9222894F...	1555799519549	0	0	2
14	942	271	1	call:910D6D1...	1555799541624	1	0	2
15	943	271	0	call:0DE785A...	1555799564000	0	0	2
16	944	271	0	call:A816489...	1555799586000	1	0	2

FIGURE 5.33: "call_log" table - voice and video calls records.

user
1 9617668

FIGURE 5.34: "jid" table - the phone number of the caller .

_id	type of call	The caller	The contact number	Time of call	Duration (sec)	Result
929	Voice call	The user	9617029****	1:15:30 AM	49	success
931	Voice call	The contact	96176680***	1:20:01 AM	55	success
933	Voice call	The user	967029****	1:24:24 AM	0	missed call
939	Voice call	The contact	9617668***	1:29:20 AM	0	refused

TABLE 5.5: The reconstruction of the WhatsApp calls history.

5.8 WhatsApp status analysis

WhatsApp added a new feature which allows the users to publish a text or a media file that will disappear after 24 hours. This temporary data can contain evidence, as criminals might use it instead of texting in order to be sure that their communication would be deleted and then no evidence would be left behind.

When the user posts a status, a record is generated in the table *messages*. This record is stored as **status@broadcast** under the field "key_remote_jid". The phone number of the contact sharing the status is stored in "remote_resource". This is shown in the first record in figure 5.35.

The second record in this figure shows that the user is sharing a status (key_from_me = 1). The field **status** stores the value of 13, means that the status was viewed by some users. The field "recipient_count" determines the number of contacts that viewed the status. There is no value stored that shows the phone numbers of these contacts.

WhatsApp allows the users to share a status in two ways. The status can be a textual data, in this case this text is stored in the field "data" in the *messages* table, or a media file such as an image or a video, and in this case a thumbnail is stored in the table *message_thumbnail*.

As in the two records in this figure the field data is *NULL* and the field media_wa_type is "one", the status shared is an image.

	key_remote_jid	key_from_me	key_id	status	data	timestamp	media_wa_type	remote_resource	recipient_count
1	status@broad...	0	5ED87E36C45...	0	NULL	156266908...	1	9617671...	0
2	status@broad...	1	E99A828E2AD...	13	NULL	156267136...	1		1

FIGURE 5.35: The WhatsApp status records.

To relate a status (image, video) to its thumbnail in the other table, we can use this SQL query using the LEFT JOIN operation. Here, we extract the thumbnail of the status in the second record in figure 5.35.

```
SELECT messages.key_remote_jid, message_thumbnails.thumbnail
FROM messages
LEFT JOIN message_thumbnails ON messages.key_id = message_thumbnails.key_id
WHERE message_thumbnails.key_id= key_id of the status concerned.
```

The figure 5.36 illustrates this query.



FIGURE 5.36: "message_thumbnails" table - identification of the status picture using SQL query.

5.9 Deleted data

The analysis of deleted data is based on the structure of SQLite database. If the WhatsApp messages are deleted, it is unable to view them in the "data" field. We can see the value *NULL* as shown in figure 5.37. However, according to analysis of SQLite database storage mechanism, we know that the actual message maybe still exists in the database but only its page header information is erased. The database will delete the first page header of deleted data and marked it as a free page but its data area is not deleted. So we can recover the deleted message as long as the deleted data area has not been covered by other data. Specifically, it is possible to locate and extract deleted data according to the logical structure of the database file page. In fact, all the data in SQLite is stored in the page and each page has its corresponding file structure. As we explain in section 3, there are different pages in the SQLite database. The data is stored in the B-tree pages. SQLite pages within a B-tree are classified as either internal or leaf pages. Internal pages contain pointers to other pages. Leaf pages contain the data. In general, each table has a root page which points to several leaf pages. This is illustrated in figure 5.38.

_id	key_remote_jid	key_from_me	key_id	status	needs_push	data	timestamp
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
209485	9617163	0	DCA6CEDB2F...	0	0	<i>NULL</i>	1554925475000

FIGURE 5.37: Deleted messages: the *NULL* value.

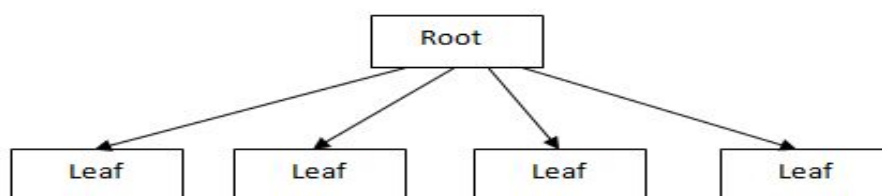


FIGURE 5.38: The structure of the B-tree pages.

All data chat records are stored in the leaf pages including the deleted messages in spaces called unallocated space (unallocated is the space after the header information and before the first cell starts where data are stored in the page (fig. 3.1)). Therefore, in order to find the deleted data, it is necessary to search the root page and use the navigation pointer to determine the leaf page of the deleted data. Finally, the deleted message can be extracted from the corresponding data area [7].

The free tools used in this study can not be used to parse the deleted data. So, and because of the expensive cost of the commercial forensic tools, we tried to use a Python script (open source project) to parse this data [26]. This script uses the logic explained above. We were able to reach these spaces but not to generate data from them. When we tried to export the content of these spaces into an excel file, the data extracted was not readable as shown in figure 5.39. The results of this script can be a support for a future work concerning the writing of an open source project to parse deleted data from WhatsApp SQLite databases specifically.

	A	B	C	D	E	F	G	H	I
1	Type	Offset	Length	Data					
2	Unallocated	12336	3573	c/					
3	Free Block	16073	23						
4	Free Block	16109	5						
5	Free Block	16162	11						
6	Free Block	16274	22						
7	Unallocated	81948	3091	ii./ qK q q qL q q \$ 99(x					
8	Unallocated	118832	2620	2					
9	Free Block	122132	60	<					
10	Free Block	122386	60	<					
11	Free Block	122570	58	:					
12	Unallocated	143456	2391	5X{ * M_p B e 7Z} ,Or !Dg 9\ .Qt"A178080774					
13	Unallocated	151560	4088	7hR\$eD%>d #X 2492 1069063I961705aribblroudahrelkbf					
14	Unallocated	159752	4088	>3'Z @u&A %+96171630720@s.whatsapp.net1440526977-					
15	Unallocated	167944	4088	siP>/mbA cL8 v m b K @ * j[# j V J 5 - r O .					
16	Unallocated	176136	4088	3t7x0f Bx P (^ 6@w :{ O #X,a4A133613492189@s.v					
17	Unallocated	184328	4088	I } {1N}}lJA '33618866387@s.whatsapp.net1440526977-32					
18	Unallocated	192520	4088	2 ` 2 V % 9Yh?d;w"IA)5 96170798031@s.whatsapp.net1					

FIGURE 5.39: The output of the script used to extract deleted data.

Note if the tables are vacuumed, there is no possibility to recover deleted data even with the best commercial tools. Fortunately, WhatsApp doesn't use an autovacuum index in their SQLite engine and therefore recovering is always possible. This what is shown in figure 3.4.

Chapter 6

WhatsApp Security

As we saw in the previous sections, the data of the WhatsApp persists on the databases even if we delete the WhatsApp application. To prevent your WhatsApp data to be retrieved and accessed, some important security measures should be taken into consideration:

1. When you delete WhatsApp, you should as well delete the databases even if the databases are encrypted. Also, you should be aware that your decryption key generated during the installation of the WhatsApp is associated to your phone number.
2. You should never backup your databases to your cloud (google drive) because these backups are not encrypted [27].
3. The screen lock is not a strong encryption, but more like you just locked the phone and many free and paid softwares can break it. The phone is encrypted meaning that the phone's data is encrypted before the phone even boots up. Almost all the new versions of Android have the possibility to encrypt the phone (internal memory where the WhatsApp and its databases are installed by default). In the older versions, it might not be present. Unfortunately, WhatsApp does not support the moving of the app to the SD card memory [28]. The SD card can be encrypted in all the versions of Android.

Chapter 7

WhatsApp in the Court

The rise of the digital tools and communications has brought new modes of the commission of the crimes. In this era of information technology, courts in all the worlds are accepting the electronic and digital communication as evidence (IM applications, social network apps, SMS).

In general, several conditions should be respected to use an electronic trace as evidence. Three major conditions are required to consider electronic communication trace as evidence:

- Admissibility under the evidence act
- Authenticity and integrity
- Relevance to the case context

7.1 Admissibility

According to the Lebanese Electronic Evidence Act, part 1 chapter II article 7, the electronic document shall be accepted as evidence and is deemed to have the same significance and power of proof as the written paper-based document if it is possible to determine the person issuing the document and if it is handled in a way to preserved its integrity [29].

The article 121 in this act defines the IT or digital evidence as the data created voluntarily or not by persons on the systems, databases, IT services and networks. IT traces are considered digital or IT evidence. IT evidence includes: hardware, software, data, applications, IT traces and the like. This definition is applicable to the evidence issued from WhatsApp SQLite databases.

if the two conditions provided by article 7 as well as the conditions in article 10 concerning the creation of copies from the evidence are respected, the WhatsApp evidence would be admissible under the Lebanese Evidence Act. The Article 15 states how to make the document more reliable.

7.2 Authenticity

Even if the evidence is admissible under the evidence act, it is still requires to pass several requirements and conditions to be accepted, mainly its authenticity.

Proving the authenticity is the most challenging aspect of submitting WhatsApp messages as evidence in court. It is important to show that the message was really sent by the person and was not fabricated. It does exist several applications that can change the conversation of the WhatsApp or created new conversation without the knowledge of the other party. These applications raise the questions about the authenticity of a screen shot or a photograph of a screen. As this thesis main goal is how to extract and reconstruct the WhatsApp messages, we are not discussing these apps and how this can affect the power of proof of WhatsApp as evidence.

To avoid the risk of tampering as excuses by the opposing side, we extract the WhatsApp messages and information directly from the SQLite databases. SQLite analysis provides a professional way to use the WhatsApp as evidence. These databases are encrypted and their decryption needs a high level of knowledge in IT and digital forensics. This is very challenging for any normal person to access these DBs and changes the information presented within. In opposite, the use of third party applications or the use of Photoshop make possible to change the screenshots easily. The data generated by the analysis of these DBs is a raw data that can be used in court by the help of an expert or a digital investigator. This will preserve the authenticity as well as the credibility of the WhatsApp evidence.

In addition of the authenticity, the integrity of the evidence should be preserved during the seizure, retention and analysis. The article 123 states that a record shall be written detailing the seizure, retention, analysis, examination or transfer thereof from one authority to another (chain of custody). The chapter VII of the Electronic Evidence Act discusses widely the rules of procedure for seizing and retaining IT evidence in order to preserve the integrity of the evidence.

Note that the final decision about an electronic evidence depends on the estimation of the court as the article 122 explains that the court may estimate, at its own discretion, the power of proof and authenticity of the digital/IT evidence, provided that such evidence is not altered in any way during seizure, retention or analysis.

7.3 Relevance

The third condition is the relevance of the WhatsApp evidence. It is not very challenging to prove if an evidence is relevant. Relevance here means that the data of the WhatsApp should be limited to the case as the article 87 of the act states " Personal data shall be collected faithfully and for legitimate, specific and explicit purposes. The data shall be appropriate, not go beyond the stated objectives, be correct and complete and remain on a daily basis as relevant as possible."

The Lebanese Evidence Act discusses widely how to protect these personal data. Part V provides comprehensive legal regulation for personal data protection. It defines the objectives and limitations of processing personal information, the cases where processing of such information is legally banned, the method by which personal information is collected, and the obligations and responsibilities of persons processing the data.

The article 88 listed the information the representative (data-processing officer) that should inform to the persons from whom the personal data are derived. This mainly information is the identity of the officer, the objectives of processing and the persons to whom the data to be sent.

In addition, some information shared through WhatsApp could not be extracted as the act determines in the article 91 the type of data that should not be extracted, collected or processed such as the health status, genetic identity or sexual life of the person concerned.

Furthermore, the section III mentions the actions required to implement processing as the article 97 states the cases where a permit or licence is required to process personal data. The article 96 explains what this permit should contain. As well, the article 94 states the cases where this permit is not required.

These three conditions (plus the personal data protection) should be carefully respected while submitting evidence in court because the other side might rely on defences to evidence admission such as chain of custody, tampering and unauthentic evidence.

7.4 Digital forensics report

To ensure that the data used will be accepted as evidence, it should be organized in digital forensics report.

Writing a report is the final step of investigation, where the findings of an investigator during the digital forensics examination is presented to the court or to the interested (impacted) persons. The report should be understood by different entities with varying IT knowledge such as the judges/juries if the report should be submitted to the court. This means that the investigator should use simple language with easy technical notions.

Most forensics reports contain a brief summary of information, tools used in the investigation process with the goal why each tool is used. Then the evidence found on the mobile phone, such as the SQLite databases or screenshots concerning the WhatsApp, is listed as a summary followed by the analysis of each portion of this evidence. The report should mention if the investigation should be continued or ceased based on the findings in the report.

Report elements and sections depend on the investigator writing style, the type of cyber crimes, and the IT skills of the persons who will read the report. In general, any effective report should contain the following sections:

1 Overview and case summary

- ***Investigator information:***

Information about who works on the case and what was his role. In our case, who handled, extracted, decrypted, and analyzed the WhatsApp SQLite databases.

- ***Case description:*** Description of the case and an explanation of the tasks requested from the investigator.

For instance, investigation of suspect's phone and specifically his WhatsApp account for possible communication with terrorist organizations.

If the examiner receives the digital evidence or the suspect's phone and he should send that evidence to another examiner, this section should contain information about the chain of custody from the moment he received the evidence.

2 Forensic acquisition and exam preparation

- ***Preservation of the evidence:***

This section is very important as it contains the steps taken to preserve and acquire the evidence. For example, before imaging (acquisition) the suspect's phone, the examiner should photograph it, documenting its information (model, brand, IMEI...), the applications installed on the phone, WhatsApp installed or uninstalled and its version, etc. The importance of these steps is related to the integrity of the digital evidence and the chain of custody.

- ***Imaging process:***

In most of the digital forensics cases, an image of the computer/phone should be made. Then, the process used when making the working copy from this forensic image of the original evidence should be described. Any additional step taken during the acquisition should be noted in this section.

3 Forensic analysis and findings

This is the longest and most detailed section of the report. It should include all artifacts that an examiner find during the analysis relating to the case.

- ***Tools (and their version) used in the analysis:***

For instance, the following tools were used to analysis of WhatsApp SQLite databases:

- WhatsApp SQLite Databases Extractor
- DB Browser for SQLite
- etc

- ***The technical procedures to extract and analyze the artifacts:***

In this part, the examiner explains what are the steps taken to analyze the artifacts and with which tools.

For example, the SQLite databases were extracted using the WhatsApp Extractor as well as their encryption key. Then, a review of the decrypted databases using DB browser for SQLite shows the following data (figure 5.9). The analysis of this data shows the "messages and contacts" of the suspect.

4 Conclusion

- ***The opinion of the investigator about the case:***

The investigator explains his opinion, based on the forensic analysis, about the result of the accusation . In our case, the conclusion should be if the suspect was in contact with terrorist persons using his WhatsApp account.

- ***Investigator's notes about the state of the investigation:***

A note to expand the investigation to other areas if needed.

5 Explanation of technical terms

Technical terms such as SQLite databases, unallocated space, and rooted and unrooted should be described in simple language with examples so a nontechnical audience can understand it.

Note that a good report will contain such qualification: the limitation of the tools used and the applicability of the technology used.

It is hard to done perfectly a report, but a good report is very important to identify the WhatsApp evidence that might be a potential concern, highlighting the relevant WhatsApp data, and their implications for the court [30] [31].

Chapter 8

Conclusion

In this thesis, we have investigated the forensic artifacts of WhatsApp messenger SQLite databases on the unrooted Android phones. We have presented a methodology based on the performing of designed experiments as well as a method to decrypt the encrypted databases by using free tools and coding scripts written for this study.

We have identified the artifacts left by the WhatsApp SQLite databases on Android phones, and we have shown their evidentiary value. We have demonstrated that it is possible to reconstruct the history of the WhatsApp by analyzing these artifacts even if the WhatsApp was uninstalled.

We have focused on how to analyze the data stored in the contact database in order to reconstruct the list of contacts of the user as well as the operation of blocking of a contact. Similarly, we have discussed how to interpret the data stored in the chat database aiming to reconstruct the state, the chronology and the content of the textual and the non-textual messages exchanged, the content of the feature called status, the communication that happened within the groups of the user, and the chronology of the voice and video calls. Furthermore, we have provided the importance to link the information stored in the different tables of the database by using SQL queries in order to assure the coverage of all information that could be missed if each table is studied in isolation.

In addition, we have discussed a method to recover the data from the damaged databases that can not be opened and parsed.

As this study was done completely by using free tools as well as Python and PowerShell scripts, we have not been able to recover the deleted data from these databases. We have tried to write a program using Python, we were able to reach the spaces where the deleted data is stored in the SQLite databases but the main problem was how to generate this data in a readable format.

First, this thesis provides a full view about the analysis of the WhatsApp messenger from the decryption of the SQLite databases to the reconstruction of the WhatsApp history and the analysis of the content exchanged by applying a methodology that can be followed in the analysis of the most of Android applications using SQLite databases. Second, this thesis shows how to manually parse the database using SQL queries to gather information from different tables easily.

As this study focuses on the SQLite databases on the Android phones, the future work should be extended to other OS such as iOS and Windows phones where the storage and the artifacts generated might be different. As well, it is important to study the network of this application side by side with the file system analysis to provide the

full picture to the analysts. Furthermore, more work is needed concerning the deleted data. As no free tools provided this efficiently yet, the creation of an open source project could be useful as a future work to recover deleted data from these databases.

Bibliography

- [1] The messaging apps report: Messaging apps are now bigger than social networks, <https://www.businessinsider.com/the-messaging-app-report-2015-11?IR=T>.
- [2] Most popular messaging apps 2019, <https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/>.
- [3] Seigfried-spellar, k.c., leshney, s.c., 2015. the intersection between social media, crime, and digital forensics: #WhoDunIt? - ScienceDirect.
- [4] Global market share held by smartphone operating systems, <https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>.
- [5] Learning iOS Forensics | PACKT Books.
- [6] Learning Android Forensics - Second Edition | PACKT Books.
- [7] Lijun Zhang, Fei Yu, and Qingbing Ji. The Forensic Analysis of WeChat Message. pages 500–503, July 2016.
- [8] Cosimo Anglano, Massimo Canonico, and Marco Guazzone. Forensic analysis of the ChatSecure instant messaging application on android smartphones. *Digital Investigation*, 19:44–59, December 2016.
- [9] Daniel Walnycky, Ibrahim Baggili, Andrew Marrington, Jason Moore, and Frank Breitingner. Network and device forensic analysis of Android social-messaging applications. *Digital Investigation*, 14:78, August 2015.
- [10] Kenneth Ovens and Gordon Morison. Forensic analysis of Kik messenger on iOS devices. *Digital Investigation*, 17:40–52, April 2016.
- [11] Marco Guazzone Cosimo Anglano, Massimo Canonico. Forensic analysis of telegram messenger on android smartphones. *Digital Investigation*, 23:31–49, December 2017.
- [12] H. Zhang, L. Chen, and Q. Liu. Digital Forensic Analysis of Instant Messaging Applications on Android Smartphones. In *2018 International Conference on Computing, Networking and Communications (ICNC)*, pages 647–651, March 2018.
- [13] K. Rath, U. Karabiyik, T. Aderibigbe, and H. Chi. Forensic analysis of encrypted instant messaging applications on Android. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pages 1–6, March 2018.
- [14] Abdullah Azfar, Kim-Kwang Raymond Choo, and Lin Liu. An Android Communication App Forensic Taxonomy. *Journal of Forensic Sciences*, 61(5):1337–1350, 2016.
- [15] Neha S Thakur. N.s. thakur. forensic analysis of whatsapp on android smartphones. master’s thesis, university of new orleans, 2013. paper 1706.

- [16] 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):38–44, April 2013.
- [17] Cosimo Anglano. Forensic Analysis of WhatsApp Messenger on Android Smartphones. *Digital Investigation*, 11(3):201–213, September 2014. arXiv: 1507.07739.
- [18] R. Hipp, P. Sanderson, H. Mahalik, B. Shavers, and E. Zimmerman. *SQLite Forensics*. Paul Sanderson, 2018.
- [19] Newzoo: Smartphone users will top 3 billion in 2018, hit 3.8 billion by 2021 | VentureBeat, <https://venturebeat.com/2018/09/11/newzoo-smartphone-users-will-top-3-billion-in-2018-hit-3-8-billion-by-2021/>.
- [20] Database File Format, <https://www.sqlite.org/fileformat2.html>.
- [21] Belkasoft Evidence Center 2018 v.8.6 Press Release, <https://belkasoft.com/bec86pr>.
- [22] Oxygen Forensics - Mobile forensic solutions: software and hardware, <https://www.oxygen-forensic.com/en/products/oxygen-forensic-detective>.
- [23] Tahani Almeahmadi and Omar Batarfi. Impact of Android Phone Rooting on User Data Integrity in Mobile Forensics. *International Journal of Advanced Computer Science and Applications*, 9(12), 2018.
- [24] GNU General Public License, <https://www.gnu.org/licenses/gpl-3.0.en.html>.
- [25] SQLite Database Disk Image Is Malformed, <https://sqliteviewer.com/blog/database-disk-image-malformed/>.
- [26] Script to recover deleted entries in an SQLite database, <https://github.com/mdegrazia/SQLite-Deleted-Records-Parser>.
- [27] WhatsApp FAQ - Backing up to Google Drive, <https://faq.whatsapp.com/en/android/28000019/>.
- [28] WhatsApp FAQ - Moving WhatsApp to an SD card, <https://faq.whatsapp.com/en/android/21068307/>.
- [29] Lebanese Evidence Act - Law No. 81 Relating to Electronic Transactions and Personal Data, https://smex.org/wp-content/uploads/2018/10/E-transaction-law-Lebanon-Official-Gazette_ENGLISH.pdf.
- [30] Understanding a Digital Forensics Report, <http://www.legalexecutiveinstitute.com/understanding-digital-forensics-report/>.
- [31] SANS Digital Forensics and Incident Response Blog | Intro to Report Writing for Digital Forensics | SANS Institute, <https://digital-forensics.sans.org/blog/2010/08/25/intro-report-writing-digital-forensics/>.