# Forensic Investigation of Social-Messaging applications on Android Devices

By

**Ayesha Arshad**

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfilment of the requirements for the degree of MS in Information Security

April 2018

# Thesis Acceptance Certificate

Certified that final copy of MS/MPhil thesis written by NS **Ayesha Arshad**, Registration No. **00000118891**, of **Military College of Signals** has been vetted by undersigned, found complete in all respect as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial, fulfillment for award of MS/MPhil degree. It is further certified that necessary amendments as pointed out by GEC members of the student have been also incorporated in the said thesis.

Signature: ______________________________
Name of Supervisor: Col Abdul Ghafoor, PhD
Date: __________________________________

Signature(HoD): _________________________
Date: __________________________________

Signature(Dean/Principal): _______________
Date: __________________________________

# Certificate

This is to certify that **Ayesha Arshad** Student of MSIS-14 Course Reg.No: **00000118891** has completed his MS Thesis title **”Forensic Investigation of Social-Messaging applications on Android Devices”** under my supervision. I have reviewed her final thesis copy and am satisfied with his work.

Thesis Supervisor,
Col Abdul Ghafoor, PhD

Date: ____________________

# Declaration

I hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification either at this institution or elsewhere.

Ayesha Arshad,
00000118891

This thesis is dedicated to *my beloved parents*

# Acknowledgments

I would like to extend my gratitude to my supervisor and Head of Department **Col Abdul Ghafoor** and to all committee members **Asst Prof Waseem Iqbal**, **Lec Narmeen Shafqat** and **Asst Prof Waleed Bin Shahid**. Without their help and support, it would have been difficult to complete this research.

I would like to give special thanks to my teacher **Waseem Iqbal**, his scholarly support and assistance in field of digital forensic throughout Master degree has been very helpful.

I would also like to extend my gratitude and appreciation to my family and friends for their moral support and words of encouragement.

# Abstract

Android Smartphones have earned massive popularity throughout the world. Social messaging application is an innovative way of sending and receiving text messages through the internet, basically it is an alternative to conventional SMS Services. These applications did not limit them to just two-person communications but group messaging, stories sharing and media sharing added charm to in-the-moment content. With increased functionality, comes complexity. The same happened to social messaging applications as the world has witnessed a huge increase in crimes. Most common social messaging applications are Whatsapp, Viber, and TelloTalk, based on feature richness and user accessibility. In this research work, these applications are forensically analyzed to see how and what artifacts can be collected at different stages; especially how much data can be recovered if data/chat or application is deleted. Moreover, general anatomy of each application is discussed, where their databases, files and messages formats are analysed. At the end, comparison of these applications is made to conclude which one provides best security to its user.

**Keywords:** *Digital Forensic, Forensic Investigation, Mobile Forensic, Social Messaging Applications, Android Framework*

# Contents

Abstract . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . viii

**1 Introduction** **1**

1.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

1.2 Problem Statement . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

1.3 Research Objectives . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.4 Reason/ Justification for the Selection of the Topic . . . . . . . . . . . . 3

1.5 Significance of Research . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.5.1 National Significance . . . . . . . . . . . . . . . . . . . . . . . 3

1.5.2 Military Significance . . . . . . . . . . . . . . . . . . . . . . . 4

1.6 Advantages . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.7 Areas of Application . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.8 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

1.8.1 Selection and installation of Messaging applications . . . . . . . 5

1.8.2 Defining Scenario and Taking Actions . . . . . . . . . . . . . . . 6

1.8.3 Rooting and Image Acquisition . . . . . . . . . . . . . . . . . . 6

1.8.4 Analysis and Reconstruction of Data . . . . . . . . . . . . . . . 6

1.9 Thesis Organization . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

**2 Literature Review** **8**

2.1 Introduction to Chapter . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.2 Android Operating System . . . . . . . . . . . . . . . . . . . . . . . . . . 8

2.2.1 Marshmallow or Android 6.x . . . . . . . . . . . . . . . . . . . . . . 9

2.2.2 Android File Structure . . . . . . . . . . . . . . . . . . . . . . . . . 9

2.3 Android Social Messaging Application . . . . . . . . . . . . . . . . . . . . 11

2.3.1 A Brief History of Messaging and Chat Applications . . . . . . . 11

2.3.2 Social Messaging Applications Vs Social Networking Applications 12

2.3.3 Reason of Popularity of Messaging Applications . . . . . . . . . . 12

2.3.4 Key Features of Messaging Applications . . . . . . . . . . . . . . 13

2.4 Selection of Messaging Applications for Research . . . . . . . . . . . . . 14

2.4.1 WhatsApp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14

2.4.2 Viber . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

2.4.3 TelloTalk . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 16

2.5 Mobile Forensics Investigation of Social Messaging Applications . . . . . 16

2.5.1 Data acquisition . . . . . . . . . . . . . . . . . . . . . . . . . . . 17

2.5.2 Identification . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

2.5.3 Evaluation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

2.5.4 Admission as evidence . . . . . . . . . . . . . . . . . . . . . . . . 19

2.6 Test environment and requirements . . . . . . . . . . . . . . . . . . . . . 19

2.6.1 Forensic Test Environment . . . . . . . . . . . . . . . . . . . . . 19

2.6.2 Mobile Forensic Tools & Techniques . . . . . . . . . . . . . . . . 20

2.7 Previous Related Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . 22

**3 Test Procedures 25**

3.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25

3.2 Research Related Social Messaging Application Scenarios . . . . . . . . 25

3.2.1 Research Related Social Messaging Application Scenarios - WhatsApp . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26

3.2.2 Research Related Social Messaging Application Scenarios - Viber 27

3.2.3 Research Related Social Messaging Application Scenarios - TelloTalk 28

3.3 Acquisition . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 28

3.3.1 Physical Acquisition . . . . . . . . . . . . . . . . . . . . . . 28

3.3.2 Logical Acquisition . . . . . . . . . . . . . . . . . . . . . . . 33

3.4 Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 33

**4 Forensic Investigation of Social Messaging Applications 34**

4.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 34

4.2 Forensic Investigation of WhatsApp Messenger . . . . . . . . . . . . 34

4.2.1 WhatsApp Anatomy . . . . . . . . . . . . . . . . . . . . . . 34

4.2.2 Analysis of Chat & media shared between participants on device - Data not Deleted . . . . . . . . . . . . . . . . . . . . . . . 45

4.2.3 Analysis of Chat & media shared between participants on Android device - Data Deleted . . . . . . . . . . . . . . . . . . . . . 49

4.2.4 Analysis of Chat & media shared between participants on device - Application Deleted . . . . . . . . . . . . . . . . . . . . . 52

4.3 Forensic Investigation of Viber . . . . . . . . . . . . . . . . . . . . 55

4.3.1 Viber Anatomy . . . . . . . . . . . . . . . . . . . . . . . . . 55

4.3.2 Analysis of Viber Chat & media shared between participants on device - Data not Deleted . . . . . . . . . . . . . . . . . . . . . 64

4.3.3 Analysis of Viber Chat & media shared between participants on Android device - Data Deleted . . . . . . . . . . . . . . . . . . 67

4.3.4 Analysis of Viber Chat & media shared between participants on device - Application Deleted . . . . . . . . . . . . . . . . . . . 69

4.4 Forensic Investigation of TelloTalk . . . . . . . . . . . . . . . . . . 70

4.4.1 TelloTalk Anatomy . . . . . . . . . . . . . . . . . . . . . . . 70

4.4.2 Analysis of TelloTalk Chat & media shared between participants on device - Data not Deleted . . . . . . . . . . . . . . . . . . . 75

4.4.3 Analysis of TelloTalk Chat & media shared between participants on Android device - Data Deleted . . . . . . . . . . . . . . . . . . 77

4.4.4 Analysis of TelloTalk Chat & media shared between participants on device - Application Deleted . . . . . . . . . . . . . . . . . . . . 79

4.5 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 80

**5 Forensic Investigation Result and Analysis 81**

5.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81

5.2 Analysis of Whatsapp Application . . . . . . . . . . . . . . . . . . . . 81

5.3 Analysis of Viber Application . . . . . . . . . . . . . . . . . . . . . . 83

5.4 Analysis of TelloTalk Application . . . . . . . . . . . . . . . . . . . . 84

5.5 Comparison of Whatsapp, Viber and TelloTalk . . . . . . . . . . . . . . 85

5.6 Guideline for Safe Deletion . . . . . . . . . . . . . . . . . . . . . . . 86

**6 Conclusion and Future Work 88**

6.1 Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88

6.2 Guidelines to Secure Android Application . . . . . . . . . . . . . . . . 88

6.3 Future Work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 89

6.4 Conclusion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90

**References 91**

# List of Figures

1.1 Research Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

2.1 NAND Division in YAFFS . . . . . . . . . . . . . . . . . . . . . . . . . . 10

4.1 WhatsApp Directory . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

4.2 Database Schema of wa.db . . . . . . . . . . . . . . . . . . . . . . . . . 36

4.3 Database Schema of msgstore.db . . . . . . . . . . . . . . . . . . . . . 40

4.4 WhatsApp Text Message format . . . . . . . . . . . . . . . . . . . . . 42

4.5 WhatsApp Media Message format . . . . . . . . . . . . . . . . . . . . 43

4.6 WhatsApp Contacts . . . . . . . . . . . . . . . . . . . . . . . . . . . . 45

4.7 WhatsApp Messages . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

4.8 WhatsApp Call Log . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46

4.9 Decrypt database file using WhatsApp Viewer . . . . . . . . . . . . . 47

4.10 Chat retrieved using WhatsApp Viewer . . . . . . . . . . . . . . . . . 47

4.11 WhatsApp database folder - Logical Acquisition . . . . . . . . . . . . 48

4.12 WhatsApp media folder - Logical Acquisition . . . . . . . . . . . . . . 48

4.13 Deleted WhatsApp conversation . . . . . . . . . . . . . . . . . . . . . 49

4.14 Manually carving of deleted WhatsApp data . . . . . . . . . . . . . . 50

4.15 Deleted group chat . . . . . . . . . . . . . . . . . . . . . . . . . . . . 50

4.16 Deleted media message . . . . . . . . . . . . . . . . . . . . . . . . . . 51

4.17 Carve deleted video by reading bytes . . . . . . . . . . . . . . . . . . 51

4.18 Extracted WhatsApp folder after deleting application . . . . . . . . . . 52

4.19 Extracted WhatsApp missed call after deleting application . . . . . . . 53

4.20 Extracted WhatsApp outgoing call after deleting application . . . . . . 53

4.21 Extracted WhatsApp messages after deleting application . . . . . . . . . 54

4.22 Directory structure of Viber data folder . . . . . . . . . . . . . . . . . . 55

4.23 Directory structure of Viber media folder . . . . . . . . . . . . . . . . . 55

4.24 Content of adx table - viber_messages Database . . . . . . . . . . . . . 57

4.25 Content of msg_info column - viber_messages Database . . . . . . . . . 58

4.26 Content of messages_call and calls table . . . . . . . . . . . . . . . . . 59

4.27 Viber Text Message format - Sent . . . . . . . . . . . . . . . . . . . . . 60

4.28 Viber media message format - Received . . . . . . . . . . . . . . . . . . 61

4.29 Viber group message format - sent . . . . . . . . . . . . . . . . . . . . . 62

4.30 Viber group message format - Received . . . . . . . . . . . . . . . . . . 62

4.31 Viber secret message format . . . . . . . . . . . . . . . . . . . . . . . . 63

4.32 Viber Conversation Details . . . . . . . . . . . . . . . . . . . . . . . . . 64

4.33 Viber one-to-one text message . . . . . . . . . . . . . . . . . . . . . . . 65

4.34 Viber one-to-one media message . . . . . . . . . . . . . . . . . . . . . . 65

4.35 Viber group message . . . . . . . . . . . . . . . . . . . . . . . . . . . . 66

4.36 Viber deleted text message . . . . . . . . . . . . . . . . . . . . . . . . . 67

4.37 Viber deleted media message . . . . . . . . . . . . . . . . . . . . . . . . 68

4.38 Viber deleted group message . . . . . . . . . . . . . . . . . . . . . . . . 68

4.39 Viber deleted call . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69

4.40 Viber deleted application marked with red cross . . . . . . . . . . . . . 69

4.41 Viber messages byte after deleting application . . . . . . . . . . . . . . 69

4.42 TelloTalk directory structure . . . . . . . . . . . . . . . . . . . . . . . . 70

4.43 TelloTalk - content of table 'awstranfer' . . . . . . . . . . . . . . . . . . 73

4.44 TelloTalk text message anatomy . . . . . . . . . . . . . . . . . . . . . . 74

4.45 TelloTalk text message anatomy . . . . . . . . . . . . . . . . . . . . . . . . 74

4.46 TelloTalk conversation - TelloTalk Viewer . . . . . . . . . . . . . . . . . . . 75

4.47 TelloTalk text messages - TelloTalk Viewer . . . . . . . . . . . . . . . . . 76

4.48 TelloTalk contacts . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76

4.49 TelloTalk deleted message . . . . . . . . . . . . . . . . . . . . . . . . . . . 77

4.50 TelloTalk - recovered text message after deleting conversation . . . . . . 78

4.51 TelloTalk - recovered media message after deleting conversation . . . . . 78

4.52 TelloTalk - Recovered a deleted contact . . . . . . . . . . . . . . . . . . . . 79

4.53 TelloTalk - Messages bytes after deleting application . . . . . . . . . . . 79

4.54 TelloTalk - Recovered messages after deleting application . . . . . . . . 80

# List of Tables

2.1 YAFFS 1 Spare data structure . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.2 YAFFS 2 Spare data structure . . . . . . . . . . . . . . . . . . . . . . . . . 10

2.3 Previous related work . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23

4.1 Structure of wa_contacts . . . . . . . . . . . . . . . . . . . . . . . . . . . 39

4.2 Keys used in end-to-end encryption . . . . . . . . . . . . . . . . . . . . . . 41

4.3 Keys used in end-to-end encryption in Viber . . . . . . . . . . . . . . . . . 60

5.1 Analysis of Whatsapp Application . . . . . . . . . . . . . . . . . . . . . . 82

5.2 Analysis of Viber Application . . . . . . . . . . . . . . . . . . . . . . . . . 83

5.3 Analysis of Tellotalk Application . . . . . . . . . . . . . . . . . . . . . . . 84

5.4 Comparison of Whatsapp, Viber and TelloTalk . . . . . . . . . . . . . . . . 86

# List of Abbreviations

| | |
|---|---|
| **IM** | Instant Messaging |
| **SMS** | Short Message Service |
| **API** | Application Programming Interface |
| **USB** | Universal Serial Bus |
| **SDK** | Software Development Kit |
| **ADB** | Android Developer Bridge |
| **OEM** | Original Equipment Manufacturer |
| **TCP** | Transmission Control Protocol |
| **CBC** | Cipher Block Chaining |
| **CTR** | Counter |
| **HMAC** | Hashed Message Authentication Code |
| **GCM** | Galois/Counter Mode |
| **SHA** | Secure Hash Algorithm |
| **AES** | Advanced Encryption Standard |

Chapter 1

# Introduction

## 1.1 Introduction

Android smartphones have earned massive popularity throughout the world. Contrary to the limited calling features, contemporary smartphones offer a vast range of features. Problems linked to social communications in the past are nowhere in the list of constraints faced by current generations. Major focus linked to improvement in mobile communication was the efficient use of available resources towards better and reliable call and text messages quality. This is not limited to receiving calls and making ones, but instant and social messaging and internet-based communication have taken over the domain of smartphones usage.

Instant messaging application is an innovative way of sending and receiving text messages through the internet, basically it is an alternative to conventional SMS Services. SMS service would simply collect complete text message and then transmit it over the channel to another party, whereas instant messaging applications convey messages character by character while they are being typed. Instant messaging applications are not just linked to text but can reliably offer secure file transfers, audio and video calls over the internet, support characters and emojis and provide clickable hyperlinks.

On the other hand, social messaging applications are quite similar to Instant Messaging applications but differ in a sense that these social messaging applications are not computer-dependent i-e these can be used directly on the smartphones. Additional to common features, the use of stickers and little-rich images have helped social messaging applications in gaining popularity in masses especially youth.

Most common social media applications are WhatsApp, Viber, and TelloTalk, based on feature richness and user accessibility. All of the aforementioned applications are available on almost all common platforms and can be downloaded freely from concerned application markets in each platform such as WhatsApp can be downloaded from Google Play store for Android. One highlighted attribute linked to popularity of social messaging applications is above mentioned cross-platform availability. These applications did not limit them to just two-persons communications but stories sharing and live videos added charm to in-the-moment content. Some social messaging applications also offer news and other important updates to restrain users from leaving the applications.

With increased functionality, comes complexity. The same happened to social messaging applications as the world has witnessed a huge 780% increase in crimes in the past four years as these applications reached in the hands of common men. This figure presents a new form of challenge; the cybercrimes through social messaging applications. These crimes include, but not limited to, social harassing, abusive message, threats, broadcasting of suicidal actions and live coverage of violent attacks.

These social messaging applications can therefore be helpful in resolving criminal situations through forensics analysis of allegedly-involved digital devices in crime scenes. Major crimes through smartphones involve cyber bullying, stalking, rumors spreading and abuse or harassment. In this thesis, we will examine three different social media applications to carry out their forensics analysis using appropriate tools and techniques. This research will lead to assisting court proceedings and spreading awareness among the common.

## 1.2 Problem Statement

- Mobile devices provide digital identity to this generation. Users expects confidentiality, integrity and availability while using native and third party applications. He assumes that the data he sends or receives through mobile devices is secure, private and accessible to only intended recipient.

- Text messages are significant component of evidence in any court proceedings. Suspect can be charged or proved innocent through digital evidence presented in the court in the form of digital messages.

## 1.3 Research Objectives

The key objectives of this research are:-

- Acquisition and forensic analysis of commonly used social messaging applications' data on Android devices.
- Complete or partial reconstruction of evidentiary traces from suspicious or deleted data after forensic analysis, exploring the properties of these applications

## 1.4 Reason/ Justification for the Selection of the Topic

The evidence recovered from the forensic analysis of smartphones and its applications can be used in court hearings and also play a vital role in investigating crimes. In recent years, many famous court cases have been decided on the basis of evidence in the form of text messages or other social communication techniques. Many popular mobile phone applications allow users to communicate via text messages at a very low rate and have, therefore, become extremely popular with users. The widespread use of these applications signifies more involvement of messages sent using these applications in investigating crimes. The users of these applications generally expect their communication to be secure; however, it is not usually the case. Despite the belief of the users that the message they are sending is received only by the expected recipient, the message or the constituents of the message can be retrieved through thorough digital forensic investigation of the mobile phone. Therefore, the study of forensic investigation of smartphone applications is beneficial not only for the advancement in field of forensic investigation but is also useful for the data security of users. The purpose of this particular research is to demonstrate the potential for acquiring digital evidence from the Android device.

## 1.5 Significance of Research

### 1.5.1 National Significance

The forensic investigation of Android devices and their messaging applications can prove to be very important for the national interests of our country. The use of social communication applications has greatly increased, which has led to a greater threat of cybercrimes. The social messaging applications allow users to stay anonymous and provide a diverse range of features, which make these applications become a tool for propagation of cybercrimes. Moreover, forensic investigation of mobile phones can greatly help the forensic analysis teams to solve cumbersome criminal cases and can also prove to be helpful for intelligence purposes.

### 1.5.2 Military Significance

Mobile phones can provide valuable information and data that can be used for military operations. These days, the military operations demand dealing with cybersecurity threats and counter-terrorism; where it is difficult to detect threats in real time. Therefore, the forensic investigation can prove to be very helpful in tracking the location and communication information from devices found at the crime scenes

## 1.6 Advantages

- Outcomes of this study will be tremendously relatable to the field of forensics.
- This research will provide meaningful assistance in many criminal investigation and court proceedings.
- This research will make users aware of the security of their communication by presenting a layer of transparency to the applications tested to eradicate their expectations about security.
- Applications and software modify constantly for improvisation, new features are added and security measures are updated. This research will help the developers of Android applications for social messaging to keep them up-to-date with the current security issues arise from weak implementation.

## 1.7 Areas of Application

- Android Mobile Forensic
- Digital forensics and incident response

- Forensic of Social Messaging Applications
- Mobile Investigations

## 1.8 Methodology

Acquisition and forensic analysis of commonly used social messaging applications' data on Android devices were carried through following research methodology:

1. Selection and Installation of Messaging Application
2. Defining Scenario and taking Actions
3. Rooting and image acquisition from Android device
4. Analysis and Reconstruction of data

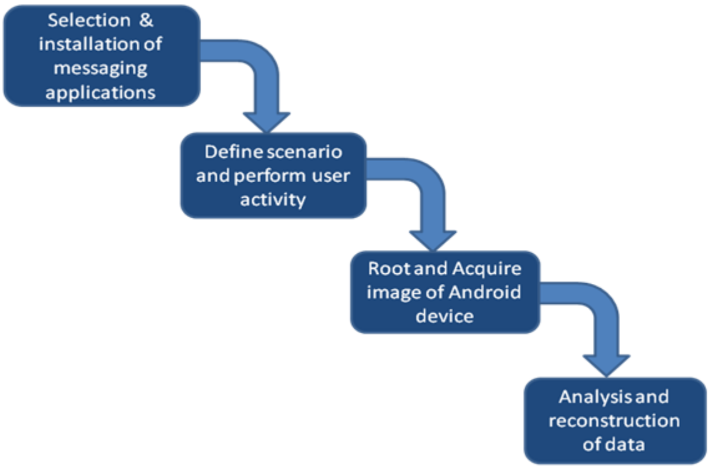

Selection & installation of messaging applications
Define scenario and perform user activity
Root and Acquire image of Android device
Analysis and reconstruction of data

**Figure 1.1:** Research Methodology

### 1.8.1 Selection and installation of Messaging applications

Google Play store provides a well-established platform for installation of applications according to the needs of users through filters and categories. For the purpose of this research, search term with keywords 'social messaging' will be used. Three major applications such as WhatsApp, Viber and Snapchat are selected on this term basis whereas the fourth application Tello was included in the list to accommodate national needs of

Pakistan, resident country of the researcher. Few other applications were overlooked because those provided almost the same communication structure as is being provided by one of the above selected social messaging applications.

### 1.8.2 Defining Scenario and Taking Actions

To keep research more precise and result oriented, some constraints were put on the scenario which included only one-on-one communications and excluding group features as well as Snapchat and WhatsApp broadcasted stories. The environment of communication was also isolated in such a way that no other applications are communicating over the networks meanwhile. It helps in avoiding of traffic overhead by applying filters for concerned social messaging application. Actions are also taken in order to understand all the activities carried out by the Android operating system on the social messaging application directly and in their stored data and cache.

### 1.8.3 Rooting and Image Acquisition

Before rooting and image acquisition, device is physically taken into custody. Rooting has been an essential need to sophisticate the forensics procedure as it helps in gaining escalated privileges over the device. It is conventionally difficult to perform forensics analysis on digital devices such as smartphones with normal or primitive usage access. Only secure and reliable rooting techniques are implemented as this rooting may exploit smartphone though efforts of ill-intended user. Once the device is rooted appropriately, logical image acquisition is performed. This process involves replicating all the data bits present in the smart phone into one or more image files. Further these image files are analyzed to filter every required or concerned file/document/database for the forensics.

### 1.8.4 Analysis and Reconstruction of Data

Final step in forensics analysis of Android based smartphones is the reconstruction and analysis report of criminal proceedings data. It is further divided into three steps; Identification, Evaluation and Admission as evidence. Identification is done to sort out and identify all important data from the smartphone image. Evaluation involves scrutinizing the data that has been identified as concerned for the crime scene. In the evaluation process, report is also generated based on the usefulness of this data. The report is then presented in the court in its evaluated form, making it worth 'a-step-forward' to the judicial proceedings.

## 1.9 Thesis Organization

The thesis is organized as follows:

- In chapter 2, literature review is delivered. Details of Android operating system, social messaging application, and selection of social messaging applications for research are discussed. Moreover, test environment and requirements that are needed to carry out the successful forensic analysis is also presented in this chapter. At the end, previous related work is discussed in tabular form.
- In chapter 3, test procedures are discussed. Scenarios that need to be executed during the research, and acquisition procedures are discussed in this chapter.
- In chapter 4, forensic investigation of all three social messaging applications is carried out and results are gathered.
- In chapter 5, results that are gathered in chapter 4 are analyzed and comparison of all three applications are also made. Moreover, guidelines for safe deletion is presented at the end.
- In chapter 6, guidelines to secure Android Application, future work and conclusion are presented.

Chapter 2

# Literature Review

## 2.1 Introduction to Chapter

In this chapter, literature review is delivered. Details of Android operating system, social messaging application, and selection of social messaging applications for research are discussed. Moreover, test environment and requirements that are needed to carry out the successful forensic analysis is also presented in this chapter. At the end, previous related work is discussed in tabular form.

## 2.2 Android Operating System

Developed by Google, Android is an operating system designed particularly for mobile phone and generally for a large range of gadgets. The basics for Android lie in extended usability of Linux Kernel in touchscreens. Most basic products that rely on Android Operating System include, but not limited to, smartphone, tablet computers, Android TV, Android Auto and Android Wear. Commercially, Android is popular because of its cost effectiveness, customizations, accessibility and ready-made nature. On the other hand, features such open-sourced nature has attracted general community as well as developers, leading towards the foundation of open projects. This predominant nature of Android has helped digital forensics teams to examine any Android-based smartphone to grab case data. Therefore, we will be looking into most common versions of Android Operating System and their nature when it comes to digital forensics.

### 2.2.1 Marshmallow or Android 6.x

Android Marshmallow or Android update version 6.x somehow managed successfully surpassed Lollipop in popularity and scalability. And on the technical grounds, it also extended challenges for digital forensics analysts. As Google moved towards Hardware-Accelerated encryption in their Marshmallow update of Android Operating System, security features such as keeping passwords in entirely unknown locations were introduced in the first place. Removable storage devices were pushed away making it harder to examine a smartphone while the device is away. Other charming security features include application permissions, supported finger print APIs etc [1].

The increasing market-share of Android Marshmallow as more and more device manufacturers started pushing it as preinstalled OS on new devices, digital forensics analysts started finding legitimate approaches to the data secured in that file system.

The vast used logical extraction process can still easily in extracting call logs, text messages and similar type of data. ADB extraction also works but it needs the device to be appropriately rooted in its nature. Recently, efforts have also been made to gain ADB access to the device under examination without the root privileges.

### 2.2.2 Android File Structure

There is a wide range of devices that depend on Android Operating System which makes digital forensics techniques vary from device to device. It is mainly due to a variant file system in each device platform. Many devices use the YAFFS File System [2] which is advantageous in providing extra security against mainstream digital forensics analysis to its contemporary competitors. Even then the NAND Internal storage makes this system pretty much weak on security as it can be exploited with the root user access. Later on, text messages, communication logs and other similar data can be extracted easily from the same YAFFS File System.

YAFFS has been brought for use in Linux Operating Environment, but with the time, its evolution extended its incorporation in other operating environments as well. NAND Division and chunks of 512 bytes and 6 bytes have been graphically shown in Figure 2.1.

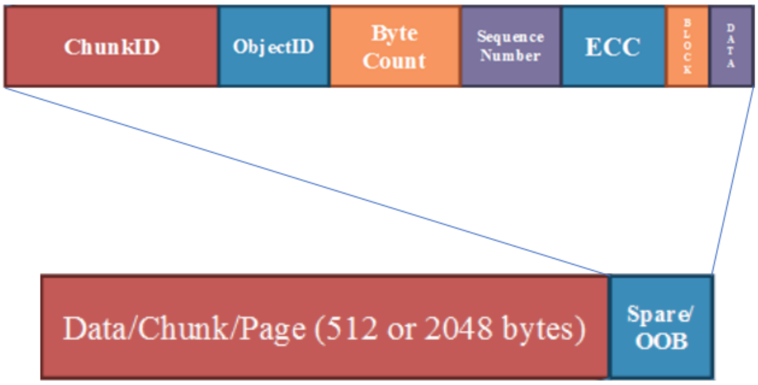


**Figure 2.1:** NAND Division in YAFFS

YAFFS1 was the first milestone in evolution of YAFFS. YAFFS2 is a more recent file system as it contains 2048 bytes and 64 bytes of spare, where as YAFFS1 is limited to 512 bytes and 16 bytes of spare. Following tables help us understand basic differences between both the versions of YAFFS.

| **YAFFS 1 Spare (16 bytes) consists of** |
|---|
| 8 bytes tags |
| 6 bytes Error Correction Code (ECC) |
| 1 byte block status (damaged) |
| 1 byte data status (dirty) |

**Table 2.1:** YAFFS 1 Spare data structure

| **YAFFS 2 Spare (64 bytes) consists of** | |
|---|---|
| Bytes | |
| 4 | Chunk ID (20) (if 0 is a header (directory entry) if > 1 is data and position) |
| 4 | Object ID (0 if unused) |
| 2 | nBytes, number of bytes used in the chunk, 0x 00 08 = 0x0800 = 2048 = full |
| 4 | Sequence number |
| 3 | ECC for tags |
| 24 | ECC for data |
| 1 | block status (damaged) |
| 1 | data status (dirty) |

**Table 2.2:** YAFFS 2 Spare data structure

## 2.3 Android Social Messaging Application

Android, as owns feature rich attributes, is also popular for social messaging applications. These applications are free, and in some cases very low-cost alternatives to the Short Message Service aka SMS. Smartphones are one of the most daily used gadgets by normal users and these can also be significantly helpful in criminal investigations and proceedings in courts in form of digital evidences.

### 2.3.1 A Brief History of Messaging and Chat Applications

Internet based messaging applications mark their basics to the early computing when highly technical users needed computer-based messaging using primitive software. With the advent of technological era, more and more users relied on electronic messaging through computers and other similar devices [3].

In the start of 21st century, messaging was restricted to either electronic messages through computers or SMS text messages through simple phones and Personal Digital Assistants (PDA) [4]. As such methods became popular in this century, general public jumped to smartphones and social networks, later merging both of them together into a common platform.

Initially those innovative alternative messaging applications for the smartphones were limited in nature. But later on, they emerged in the form of full-fledged multimedia platforms, making media a complementary add-on to textual communication and conversations.

With the better resource consumption and performance in mobile devices, social messaging applications managed their pace accordingly. To date, there are more than hundred smartphone applications that assist its users in social messaging. Contemporary features in such applications include but not limited to video calls, audio calls, group messaging, stickers and content sharing.

Apparently, WhatsApp manages monopoly rather anomaly in social messaging industry, by providing end-to-end encryption for the text messages.

### 2.3.2 Social Messaging Applications Vs Social Networking Applications

Social messaging and social networking applications basically provide a platform for people to connect with each other and share multimedia content. Actually, both have a grey line of difference between them. Social messaging applications are primarily used for direct communication between two or few parties. These can have temporary or long-lasting conversations, kept privately and intended for a specific audience.

Whereas social networks are meant to help in one-to-many and many-to-many communications. These applications are fairly efficient enough to provide durability and produce long-lasting network effects. Content on social network applications remain public in most of the cases therefore we can comfortably call them broadcast applications.

Considering it the need of the hour, major social applications both from messaging and network domains are converging by blurring the grey line of difference. This statement can be easily supported by a social media network named Instagram that later in December 2013 introduced Instagram Direct to incorporate messaging feature between Instagram users [5].

### 2.3.3 Reason of Popularity of Messaging Applications

There are several reasons that promoted such messaging applications in a wider circle of audience. Most prominent is the privacy; users prefer apparently secure messaging applications for their daily life communications. Youth is found more inclined towards the notion of security and privacy by controlling their online personas, hiding activities from parents and other authorities and preventing schools from monitoring their day-to-day activities.

Another major factor is the disposability of messaging applications as these now offer multiple accounts or feature to switch between accounts. A smooth change in persona and group is always an appreciable activity.

Different people from variant locations can communicate through vast range of devices yet they achieve a seamless connectivity and availability throughout. It has been made as such through the scalable feature of messaging applications.

Comparing social messaging applications to social network based applications, we can easily avoid spamming and unwanted communications. For example, social network Facebook can be awkward sometimes when it floods one's timeline with acquaintances' posts and activities than the closed-friends'.

### 2.3.4 Key Features of Messaging Applications

Basic messaging features include:

- Textual Chat
- Online and Offline Messages Access
- History and log maintenance
- Activity Status of the users

Groups Messaging features are:

- Customizable Chatrooms
- Message Broadcast
- Electronic Conferences

Content and Data Features are:

- No file size constraints
- Parallel Browsing
- Grabbing Screenshots
- Remote Accessibility

Enhanced Chat features:

- Audio chat
- Visual Chat
- Whiteboards
- Multi-protocols of communication

Complementary Features:

- Compatibility
- Antivirus
- Spell-Check
- Updates

## 2.4 Selection of Messaging Applications for Research

Android is supported by Google Playstore as its marketplace for additional applications to increase accessibility of Android Operating System based smartphones. Doing a simple search in the store with the keywords like social messaging, we came across a list of 3 major application designed for the same very purpose. The list includes applications based on user space and ratings; WhatsApp, Viber, and TelloTalk.

This work only focuses one-on-one and group communication features provided by aforementioned applications which do not include WhatsApp broadcasted stories etc. All such applications were individually monitored as in a controlled environment to make it sure no other traffic gets analyzed meanwhile as smartphones have several services running simultaneously. This research also carries work performed on the Android operating system of smartphone to know what that does with social messaging applications and the data stored in application folder, cache and other locations inside the device.

### 2.4.1 WhatsApp

WhatsApp [6] is available across all contemporary platforms. It has a user base of more than 1.2 billion active accounts [7]. It was later bought by Facebook and seems to stay for more years to come.

Similar to other applications, WhatsApp artifacts are efficiently valuable in several examinations linked to recovering evidence and in other similar investigations [8] [9] [10]. As WhatsApp has proven to ensure end-to-end encryption, it is beneficial for the digital forensics analyst having same data available on both victim and suspect's smartphone.

Thus artifacts from WhatsApp can help in resolving many criminal cases which will be unfolded in coming chapters.

In Android smartphones, WhatsApp maintains two securely encrypted files as databases which trigger the investigators; wa.db and msgstore.db. The former contains all the details about a user's contacts whereas the latter is linked to storing chat conversations for all the contacts.

For the last few months, WhatsApp has been emphasizing on end to end encryption. Claiming that privacy and security are in their DNA. With the help of end to end encrypted communication, all data messages are properly created to keep them safe from adversarial attacks. Only constraint in end-to-end encryption is that WhatsApp application for PC provides encryption in latest versions only. It means that is one of the devices is not updated properly, other device communicating to it would not adopt any end to end encryption feature. WhatsApp is keeping end to end encryption always activated which means that there is no option available return of this encryption technique.

WhatsApp uses tables in its database files in such a clean way that everything appears so organized in a tray for the investigators. With data and time stamps, it has also been observed that WhatsApp keeps record of geocoordinates in the form of longitude and latitude while some message is sent or received.

After completion of the end-to-end encryption scheme by WhatsApp, the company itself claim that even they do not have access to the content of communication between two parties.

After the third quarter of year 2013 WhatsApp started and greeting its conversation and communication files in a new format with an extension .crypt12. This compression and encryption technique is basically a spongy castle algorithm extension from cryptographic libraries of Android. The idea of decrypting the database is better made simple. In this way, user makes backup of messages and then uninstall the current version of WhatsApp, the user then install the older version of WhatsApp which would support no encryption.

### 2.4.2 Viber

Viber is social messaging application with over 900 million users [11]. Similar to WhatsApp, Viber also maintains SQLite database to record all secure and private communications between user and his contacts. If a forensics analyst dumps a Viber SQL database, he can view all details in CSV, Text and Html format with not much effort involved. An existing word list is forced on the database which extracts out word patterns. Then these word patterns are matched to general user behaviors to generate a table of conversations against all contacts.

### 2.4.3 TelloTalk

Till now we have been looking into different messaging applications that are globally available now I will look into a messaging application named TelloTalk which is localized in Pakistan [12]. This application is pretty much similar to WhatsApp as it provides person to person chat, group chat sharing content and location and few other multimedia features. This messaging application also comes with the building feature to edit your last message this feature of Editing your last message can help cause analysis to know that phone is keeping editable copy of the message on both ends; the sender and the receiver.

## 2.5 Mobile Forensics Investigation of Social Messaging Applications

As a new field, smartphone forensic is proving itself to be an interesting topic for forensics community. Law enforcement Agencies have started investigation through Smartphones and these investigations appear to be cheaper, easier and less time consuming. With every passing day, highly scientific techniques help digital evidence extraction to a greater level. In most cases, removal of the memory card from the smart phones has helped in a way that all data is stored in one place; the internal memory of the smartphone.

As the crime rate is increasing in the world, researchers have started to bring more affordable investigation techniques through smartphones. Here the question appears why is mobile forensics so important?

The answer to above question is pretty much simple as devices like mobile phones, Gadgets and other similar connectivity devices have started replacing the computers and these devices stay connected to our bodies like personal area network most of the day. Smartphone devices carry repository of data that can be of personal nature related to the owner. This data includes text messages, phone call logs, audio recordings, video recordings, user location, banking transactions and much more. These details are efficiently enough for the investigators to resolve the crime scene.

### 2.5.1 Data acquisition

Data acquisition is a legal process in which the forensic investigators collect data from the Android smartphone on some legal basis. This process has been divided into further 4 domains. It starts with acquisition of device physically then penetrating into it to gather data and then data is transformed into the information which is further provided to the concerned authorities as an evidence.

We will be looking into all of these in different paragraphs:

#### 2.5.1.1 Physical Acquisition

Physical acquisition is the starting page in which the device is taken into custody or anything that carries the multimedia linked to the Crime Scene is held by the investigators. It is mandatory that the acquisition in physical case must be done through proper channel by the approval of concerned higher authorities.

#### 2.5.1.2 Logical Acquisition

Logical acquisition is linked to the data that we need in particular case. Generally speaking mobile phone carry too much data and this data has more of the stuff that it is irrelevant to the investigation. Therefore, the investigators have to logically divide the data into the useful one and the useless one. It can be considered as the phone calls are divided from pictures or simple text messages are divided from videos if the videos on unconcerned for the case.

#### 2.5.1.3 Collection objective

At this time, we have all the text messages and all the phone call logs that are important for our investigation. Here Digital forensic team divide the calls that are concerned with the case investigation and random calls made by the victim or suspect. At this stage, a parallel team starts logging the report of their Crime scene investigation through the forensic of a smartphone.

#### 2.5.1.4 Root or not to root

With increasing security provided by the applications for social messaging, it is becoming difficult for forensic analyst to carry on Digital forensic of the smartphone with primitive and normal user privileges. Security providing agencies that incorporate the security methods within messaging applications have started hiding their files into the root directory. Therefore, the forensic analyst has to go through the rooting of the mobile phone device if possible to gather as much information as available for the investigation. Rooting is not a mandatory step in the investigation process of data acquisition but to the investigation team it does wonders that were not possible within an privileged user access, it provides an extra information.

If the suspect Android device is not rooted, there are chances that forensics team will fail to acquire any data, information or settings reserved by the preinstalled Social Messenger Application in the root directory of the Android operating system. Therefore, rooting comes with its advantages and its limitations in context to digital forensics analysis. The limitations of rooting a device may lead to an international Malware installation or privileged access to unauthorized users or making system vulnerability to penetrations by hackers.

During forensic, analyst team prefers not to root the device as rooting device may delete all the data and file system directories. Only secure and reliable rooting techniques should be implemented as this rooting may exploit your phone though efforts of ill-intended user.

### 2.5.2 Identification

Identification is a complete process of obtaining data, then segregating it, converting it into information and later reporting that information as an evidence.

### 2.5.3 Evaluation

Evaluation phase consists of placing useful data in a manner that it can further be used to record information in relevant direction.

### 2.5.4 Admission as evidence

When the data has been evaluated, it is moved to the evidence department where they decide that the data is worth an evidence to the Crime digital investigation through forensics.

## 2.6 Test environment and requirements

Test environment for Smartphone forensic analysis needs highly qualified investigators, Software and Hardware tools and the digital device under investigation. From now on we will be looking into all the major building blocks/ components that are required to carry out the successful forensic analysis of a smartphone.

### 2.6.1 Forensic Test Environment

To gather data, all the test scenarios are performed on Huawei p8 lite running Android 6.0.1 marshmallow. For forensic analyses, I have used a Samsung 300E4V laptop machine running 64-bit Microsoft Windows 10 operating system. Machine has Intel Core i3 processor that runs at 2.50GHz and 4GB RAM.

All the tools mentioned in next section are installed and configured to system. Android device is rooted using Kingo Root. Android Debug Bridge and Data duplicator is used to capture physical and logical image of device. Rest of tools mentioned in next section are used to analyze captured data.

### 2.6.2 Mobile Forensic Tools & Techniques

#### 2.6.2.1 Kingo Root

Kingo root is the most advanced open source rooting tool for the Android smartphones [13]. As already mentioned that smart phone rooting is not a mandatory step in data acquisition for forensic analysis but it can also be helpful, in case the security provided by the application is strong. Efficiency of the application lies in its feature of one click/ step rooting. It can be downloaded from KingoRoot website [14].

#### 2.6.2.2 Android Debug Bridge (ADB)

Sometimes, it happens if the computer forensics tool does not get directly integrated into the smartphone, we have to depend on connectivity to the phone via Android debug Bridge. Through simple Android debug Bridge [15], we can access all the contact details, call logs, messages and all forensics related data, even it has been deleted several days ago, through SQL forensics tools.

There are techniques available which can even penetrate into the phone to extract username and passwords of the social applications being used by the owner of that smartphone. By default, the mobile phone manufacturers in Android industry try to restrict ADB access to keep the owner safe from intentional or unintentional Malware attacks. ADB USB debugging mode requires that forensic team at least for once turn on debugging mode from the settings and further development settings of the smartphone [16].

#### 2.6.2.3 Data Duplicator (DD)

Data Duplicator is the simple method of replicating an operating system with all its files and folders. Through data duplication one can easily copy all the data from the suspect smartphone therefore if the phone misbehaves or gets corrupted, a complete backup of the data is still available with the forensics team. There are several tools available for data duplication but the most common used by the forensic analysts is DD the data duplicator [17].

#### 2.6.2.4 Cellebrite - UFED Physical Analyzer

Cellebrite is one of the most important companies when it comes to forensic analysis of digital equipment [18]. Therefore, we have significantly considered using UFED physical analyzer to analyze data from Android smartphone. For the purpose of this research we requested a license of trial version of UFED physical analyzer from the aforementioned company [19]. On industrial scale, the premium version of UFED digital analyzer is available for digital forensics.

#### 2.6.2.5 WhatsApp Viewer

WhatsApp viewer is a small tool for the PC that is used to view Android backups made by WhatsApp application in its supported versions like crypt 5, crypts 7, crypt 8 and crypt 12. It can be downloaded for github [20]. Main features of the WhatsApp viewer are listed below:

a. It allows you to see your WhatsApp chats on PC
b. It keeps a backup of your Android WhatsApp application on your PC
c. It also supports older version of conversation encryption used by WhatsApp
d. It provides search feature
e. Even the backup from crypt to TXT, HTML, and JSON conversion is available

#### 2.6.2.6 DB Browser for SQLite database

DB Browser [21] is an essential component as we have already seen in above paragraph that almost all the major messaging applications on Android devices and give their conversation into an SQLite DB file. Besides providing features like creating and compacting database files, defining records, DB browser also enables forensics team to view the database that has been produced by social media application such as WhatsApp, and Viber.

#### 2.6.2.7 WinHex

It is hexadecimal editor used in field of computer forensic and data recovery [22]. It is used to inspect all kind of files pulled from all kind of digital devices. Features of WinHex that are used in this research are:

- Inspect/analyze files byte-by-byte
- Search hexadecimal and text values
- Compare and analyze two different files

## 2.7 Previous Related Work

Following research works have contributed towards the forensic investigation of social messaging applications on Smartphones:

| **Ser No** | **Title** | **Brief Description** |
|---|---|---|
| a. | Forensic analysis of social networking applications on mobile devices [23]. | The authors have forensically analyzed three social networking applications named Facebook, Twitter and MySpace on BlackBerrys, iPhones, and Android phones. This research intended to find data on internal memory of device if generated by activities performed on these applications. |
| b. | iForensics: Forensic Analysis of Instant Messaging on Smart Phones [24]. | The authors have forensically examined three instant messaging applications on iPhone. These applications are AIM, Yahoo! Messenger and Google Talk. This paper aimed to investigate traces left by these applications on mobile devices. |
| c. | Forensic Analysis of WhatsApp Messenger on Android Smartphone [25]. | The author has analyzed the WhatsApp Messenger application to trace the artifacts left behind on software-emulated Android devices. |

| | | |
|---|---|---|
| d. | Forensic Analysis of the ChatSecure Instant Messaging Application on Android Smartphones [26]. | The author has forensically analyzed the artifacts generated on Android device by ChatSecure, a secure Instant Messaging application that provides strong encryption (AES-256). If the secret passphrase that is selected by user at the initial step is known, the author has been able to decrypt the encrypted messages. |
| e. | Android forensics analysis: Private chat on social messenger [27]. | The author of this paper has discussed the process of acquisition, analysis and interpretation of secret messages from Telegram, Line, and KakaoTalk. |
| f. | Forensic Analysis of Instant Messenger Applications on Android Devices [28]. | The authors have carried out a forensic analysis of WhatsApp and Viber; two most widely used applications for social communication. The goal of this research is to define the data and information that can be found on the devices internal memory. |

**Table 2.3:** Previous related work

This research work however will aim at forensic analysis of the social messaging applications specifically used by the users of Pakistan. Contrary to the previous researches that forensically analysed limited features of very few widely used messaging applications, this research will analyse three widely used Android applications in a broader perspective.

TelloTalk is first Pakistani social messaging application and has not be forensically analyzed yet, so this research would be first of its kind. Though WhatsApp and Viber have been analyzed before but focus of previous research works were mostly on data that is not deleted but this research will not only try to analyse different features when data is present but also it will try to find out that what kind of data can still be extracted after deletion of message, conversation and application itself.

Chapter 3

# Test Procedures

## 3.1 Introduction

The test procedure consisted of three stages: scenarios, acquisition, and analysis. The following sections describe each stage in details.

## 3.2 Research Related Social Messaging Application Scenarios

User activates that are common on social messaging application are carried out in this stage. Applications are installed if they are not already installed on Android device. Hence this stage consists of two scenarios:

1. Pre-Installed applications
2. Installing applications on demand and performing activities

**Pre-Installed applications:** Bloatware applications: most of the vendors in the mobile industry really come up with the devices that have pre-installed applications. Any general statistical analysis you come to the conclusion that these applications are more used by the users than other applications installed from "Google Play Store" or other similar sources. Among such pre-installed applications, WhatsApp is one of the most common application for messaging and social communication.

**Installing applications and performing activities:** It is significant here to mention that some of the applications for pre-installed and few others were installed manually through Google play store such as Viber and TelloTalk.

Major reasons behind choosing this application is their availability and accessibility as standalone application for all possible platforms. During the research process several accounts were created on all above mentioned application to conduct social messaging experiment in the smartphones. These accounts were logged into for the research purpose.

During the experimentation several predefined processes and activities were carried out in all applications based on the features they offer. Most common applications were communicating through text messages sending and receiving audios, photos, videos etc. Realizing different activities performed by various applications on smartphones regarding messaging and social media communication almost all the features were tested.

Forensic analysis will be conducted on three different levels for each application:

1. Application installed and working
2. Application installed and working but data has been deleted
3. Application and data both have been deleted

At level one, all data is available and application is in use. At second level, researcher will delete partial or complete data but the application will remain in the phone. At third level, both data and application will be deleted from the mobile device.

### 3.2.1 Research Related Social Messaging Application Scenarios - WhatsApp

Following user activities will be performed and analysis on WhatsApp application on Android device for research purpose:

1. Contacts and their status
2. Groups and their information
3. Message Sent and received

4. Message Sent and received in group
5. Audio, video and image sent and received
6. Document sent and received
7. Location sent and received
8. Contact information sent and received
9. Call history

All these scenarios will be tested on three different levels as mentioned above.

### 3.2.2 Research Related Social Messaging Application Scenarios - Viber

Following user activities will be performed and analysis on Viber application on Android device for research purpose:

1. Contacts and their information
2. Groups and their information
3. Message Sent and received
4. Message Sent and received in group
5. Audio, video and image sent and received
6. Document sent and received
7. Location sent and received
8. Contact information sent and received
9. Secret chat sent and received

All these scenarios will be tested on three different levels as mentioned above.

### 3.2.3 Research Related Social Messaging Application Scenarios - TelloTalk

Following user activities will be performed and analysis on TelloTalk application on Android device for research purpose:

1. Contacts and their information
2. Message sent and received
3. Audio, video and image sent and received
4. Document sent and received
5. Location sent and received
6. Contact information sent and received

All these scenarios will be tested on three different levels as mentioned above.

## 3.3 Acquisition

The two most important phases of this process is acquisition of physical image from the internal memory of the smartphone. This position was carried out in an environment perfectly control for forensics analysis in order to maintain integrity of the data images and their potential acceptability in court. There is a greatest chance that some file which remain inactive for longer time may be skipped during the acquisition process. It is appropriately similar to computer forensics where logical extraction skips data present in the stack space.

### 3.3.1 Physical Acquisition

When we have to physically acquire the data, we face the choice of whether rooting the mobile device or not. As mentioned before rooting the device may affect the integrity of data. Below, we have mentioned two methods that will be used to physically acquire the data. First one is by rooting the device and second one is by booting through recovery mode. If we captured a mobile during the investigation that is already rooted, we can move to the imaging part.

#### 3.3.1.1 Enable USB debugging

One of the most important features link to Android devices is USB debugging mode. Android software development kit (SDK) accommodates a computer to make a connection with Android device. USB debugging mode can be enabled by simply connecting your Android device and turning the future on in Android settings.

USB debugging mode enables an administration level system access of the smart phone. Generally USB debugging mode is used by the developers to test and try their new application before releasing the final version. It also gives the administrator access to the device directly from the computer. It allows you to run terminal commands to ADB right on your smartphone from your computer. In this way we are able to run forensics tools on the smartphone remotely.

Step to follow to allow USB debugging are:

1. Go to **About phone** in **Settings** of your system..
2. Tap **build number** seven times. “You are now a developer” message will appear.
3. Go back to Settings.
4. Now you can see new menu **Developer options**, select it.
5. In Developer option menu, first slide **Developer options** on and then **USB debugging**.

#### 3.3.1.2 Unlock Bootloader

Unlocking bootloader is necessary before flashing custom recovery image because bootloader verifies signature of recovery image with company’s official signature and as signature of custom recovery image doesn’t match, user has to unlock bootloader before flashing custom recovery image. Some devices do not provide root access; in that case user has to install custom recovery with su binaries and this is not possible without unlocking bootloader.

Before starting the unlocking process, get the OEM unlock code from Huawaei website. [29]

Following are steps to unlock boot loader of Huawaei P8 Lite:

1. Connect your mobile device to system
2. Enable USB debugging
3. Open up a terminal window with administrative rights. If you have installed adb and it is in your system's PATH, type the following:
    a. adb device
    b. Note: if system PATH is not set, first move to the folder where adb.exe reside using cd command and the enter the above command.
4. On mobile device, "Allow USB debugging" prompt will appear. Click OK to allow.
5. Type following commands
    a. adb reboot bootloader
    b. fastboot devices
    c. fastboot oem unlock [Insert code here that you get from Huawaei website]
    d. fastboot reboot
6. This will succesfully unlock your bootloader.

#### 3.3.1.3 Rooting a device

Following are steps to root Android device using KingoRoot:

1. Connect your mobile device with system using USB cable.
2. Enable USB debugging mode on mobile device.
3. Open KingoRoot software on your system and let it detect your device.
4. Once your device is detected, a "Root" button will appear.
5. Click on Root button. KingoRoot will run many exploits and this process can take several minutes to complete.

**To check if your device has been rooted properly or not.**

1. Install Root Checker Basic app on mobile

2. Click on Verify Root Button.

3. If your device is rooted, "Congratulations! Root access is proper installed on this device!" will be shown.

#### 3.3.1.4 Recovery Partition

Small devices like mobiles phones, tablets and television, where Android framework generally deploys, has no secure boot. Though these Android gadgets come with recovery partitioning that is used for recovery purposes. Normal boot process is skipped when Android device starts in "Recovery mode" and image stored in recovery partition is loaded.

Many studies are carried out to build custom recovery image, one such study ensures integrity of user data [30].

Custom recovery image can be loaded by flashing recovery partition or part of ROM. Many Android devices store details of recovery partition in /proc/mtd directory.

Sometimes, NAND flashing becomes significant which is also performed from computer on the smartphone using Fastboot utility over USB. There are different methods to enter into the Fastboot mode, varying from vendors to vendors and even within similar products. Once the device enters the Fastboot mode, it is ready to accept and run remote commands from computer.

Presence in the Fastboot can be verified using following line of command:

```
./fastboot devices
```

A successful indication of a device in Fastboot mode enables the user to flash a custom recovery partition. If the new recovery image is named as recovery-file.img then command goes this way:

```
fastboot flash recovery modified-recovery-image.img
```

#### 3.3.1.5 Taking Image of device

Following steps are taken to acquire the physical image of Android device.

1. Connect your device to the system.
2. Open up a terminal window with administrative rights. If you have installed ADB and it is in your system's PATH, type the following:
    a. adb device
    b. Note: if system PATH is not set, first move to the folder where adb.exe reside using cd command and the enter the above command.
3. On mobile device, "Allow USB debugging" prompt will appear. Click OK to allow.
4. On terminal window, type "adb shell". It creates a shell session that enable you to communicate with your device by typing commands. Now all the commands that run on this shell will execute on your device.
5. Now type "su", this will enable super user (root).

dd command is helpful in modifying block files inside the device; reading and writing concerned files. Whereas, netcat command accommodates in forwarding all communication across appropriate ports. Using both of the these simultaneously, users can read and write device files remotely from the computer via already established USB Connection.

Replicating an image of the device needs some precise commands. It is done through two separate shells running on the host computer; one shell manages session to the smartphone whereas the other shell runs commands on the computer. Connect one terminal with adb access that communicates with the device and other terminal inside windows shell in order to locate saving location for the to-be created image.

6. Now, open another terminal window with administrator rights and choose the directory where you want to save the image and type the following:
    a. adb forward tcp:8888 tcp:8888
    b. If adb PATH is not defined use: C:\Users\XXXX\AppData\Local\Android\sdk\platform-tools\adb.exe forward tcp:8888 tcp:8888
    c. Now adb can listen/communicate to 8888 tcp port through netcat.

7. Now back to the shell of your mobile device and type following command:

    a. dd if=/dev/block/mmcblk0 | busybox nc - 1 - p 8888

    b. This command reads the contents of /dev/block/mmcblk0 (the head block of my device) and writes it via port 8888 across adb using netcat.

8. Finally, back in the shell to the computer, type the following:

    a. nc 127.0.0.1 8888 > device_ image.dd

    b. This command saves the output of the contents across port 8888 (which will be the results of reading /dev/block/mmcblk0 on the device, or the complete image of the device) to the file device_ image.dd.

### 3.3.2 Logical Acquisition

Most of the digital forensics analysts prefer logical data acquisition methods in the first place due to the accessibility and result-oriented nature of such tools and techniques. One of technique to perform the logical acquisition is creating a backup of device. ADB provides following command to backup the device.

adb backup -all -f [Path_To_Store_Backup].ab
This command will take the full backup of device and save it on desired location.

## 3.4 Analysis

Third stage involved performing forensic examination on the acquired logical images in order to determine if the traces of all activities conducted on the device were present on the devices' internal storage. If the above stayed true, the amount, location and significance of the data found and acquired from the devices were determined. The examinations were carried out manually with a number of tools to view the acquired images, determine unique headers or signatures in each structure search for data related to the social messaging applications, and to determine the ways these data were stored in each device. In next chapter, we will analysis the selected applications in detail.

Chapter 4

# Forensic Investigation of Social Messaging Applications

## 4.1 Introduction

In this chapter, WhatsApp, Viber and TelloTalk are forensically analyzed and results are gathered.

## 4.2 Forensic Investigation of WhatsApp Messenger

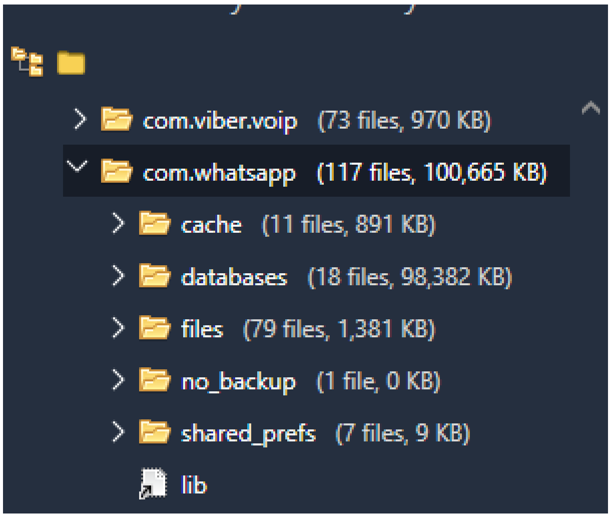
In this section, WhatsApp artifacts that are important in forensic investigation are discussed. This section also explains ways to extract WhatsApp data from Android device.

### 4.2.1 WhatsApp Anatomy

Data generated by the WhatsApp app is stored in the internal device memory, which is normally inaccessible by users. Data folder is located in /data/data/com.whatsapp/ directory as shown in Figure 4.1. This directory has five folders named: cache, databases, files, no_backup, shared_pref.

- Media files are stored in /data/media/0/WhatsApp/Media directory.
- Backup of messages databases are stored in /data/media/0/WhatsApp/Databases directory.

- Backup of WhatsApp Application is stored in /data/media/0/WhatsApp/Backups directory.



```
> com.viber.voip   (73 files, 970 KB)
v com.whatsapp   (117 files, 100,665 KB)
    > cache   (11 files, 891 KB)
    > databases   (18 files, 98,382 KB)
    > files   (79 files, 1,381 KB)
    > no_backup   (1 file, 0 KB)
    > shared_prefs   (7 files, 9 KB)
      lib
```

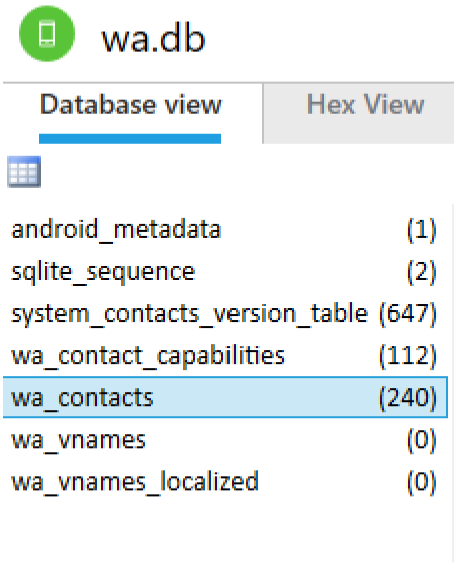
**Figure 4.1:** WhatsApp Directory

#### 4.2.1.1 WhatsApp Databases

WhatsApp application stores generated data in databases and these databases files are located in the /data/data/com.whatsapp/databases directory. The database that holds the user chat data are encrypted with crypt-12 algorithm and can be decrypted using the key that is placed in /data/data/com.whatsapp/files/key. There are 7 databases in the WhatsApp, listed below:

- axolotl
- chatsettings
- chatsettingsbackup
- hsmpacks
- msgstore
- wa
- web_sessions

*axolotl.db* stores public keys and will be discussed later in this chapter. If user is connected to WhatsApp through web browser, *web_sessions.db* holds the information about this session including browser_type, token, operating system, latitude, longitude, place name, last active timestamp, and expiry date of current session. *wa.db* stores contacts and groups information and *msgstore.db* stores messages related data. In next section, I have discussed these two databases in detail.



| | |
|---|---|
| android_metadata | (1) |
| sqlite_sequence | (2) |
| system_contacts_version_table | (647) |
| wa_contact_capabilities | (112) |
| wa_contacts | (240) |
| wa_vnames | (0) |
| wa_vnames_localized | (0) |

**Figure 4.2:** Database Schema of wa.db

**4.2.1.1.1 Anatomy of wa.db** WhatsApp stores contact information in wa.db file and it is located in the /data/data/com.whatsapp/databases/ directory. As shown in Figure 4.2, there are seven tables in this database: android_metadata, sqlite_sequence, system_contact_version_table, wa_contact_capabilities, wa_contacts, wa_vnames, wa_vname _localized.

Table wa_contacts stores crucial data that might help during forensic investigation. Other tables store housekeeping information and hold no evidentiary information. Structure of wa_contacts is explained in Table 4.1.

| **Field name** | **Meaning** | **Comments** |
|---|---|---|
| _id | Auto Incremented sequence number of record; also primary key of this table. | |

| | | |
|---|---|---|
| Jid | WhatsApp id of contact, group or broadcast message. Contact id is saved in x@s.whatsapp.net format where x is the phone number of contact, group id is stored in a-b@g.us where a is number of contact who created this group and b is unique number assigned to it, broadcast message is stored in status@broadcast format. | |
| is_whatsapp_user | If contact represents actual Whatsapp user, its value is '1'. Otherwise its value is '0'. | |
| status | Status text of the contact (as set in his/her profile) | |
| status_timestamp | Time when status was set | |
| number | Phone number of user stored in phonebook of device. | Data in this field is coming from phonebook of device. |
| raw_contact_id | Contact id of user, this field is retrieved from phonebook of device. | Data in this field is coming from phonebook of device. |
| display_name | Name of user stored in phonebook of device. | Data in this field is coming from phonebook of device. |
| phone_type | Type of phone | Data in this field is coming from phonebook of device. |

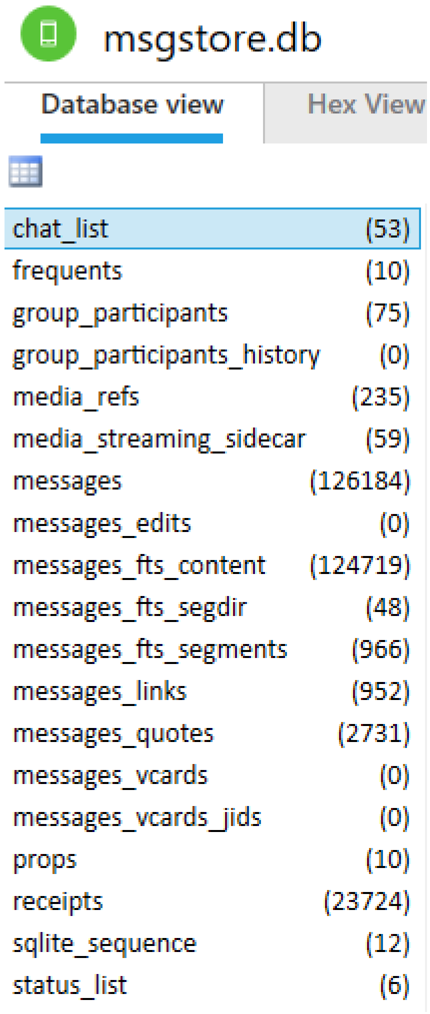
| | | |
|---|---|---|
| phone_label | If phone_type is 0, value will be mobile. If contact is a group, value represents the epoch timestamp of when the group was created. Otherwise this column is assigned null value. | Data in this field is coming from phonebook of device. |
| unseen_msg_count | Number of messages that are unseen | |
| photo_ts | | |
| thumb_ts | Unix epoch timestamp (10 digits); indicated when user set his display image. | |
| photo_id_timestamp | Unix millisecond epoch timestamp (13 digits); indicates when display image is downloaded in user device. | |
| given_name | Given name assigned to user | Data in this field is coming from phonebook of device |
| family_name | Family name assigned to user | Data in this field is coming from phonebook of device. |
| wa_name | WhatsApp name of the contact (as set in his/her prole) | |
| sort_name | name of the contact used in sorting operations | |
| Nickname | Nick name assigned to user | Data in this field is coming from phonebook of device. |

| | | |
|---|---|---|
| Company | Company of user | Data in this field is coming from phonebook of device. |
| Title | Title assigned to user | Data in this field is coming from phonebook of device. |
| status_autodownload_disabled | Auto downloading of status is disabled or enabled. | |

**Table 4.1:** Structure of wa_contacts

**4.2.1.1.2 Anatomy of msgStore.db** One of the most important database file from the forensic point of view is msgstore.db. It stores actual messages, chat and group information. As shown in 4.3, there are 20 tables in this database, and are explained as follow:

1. chat_list - table that stores the list of all chats in which user participated.
2. frequents - it holds the message count of frequent chats.
3. group_participants - Information of groups in which user has participated, this also holds the admin column that indicates that user is group admin or not.
4. group_participants_history -
5. media_refs - It holds the path and reference count of media that was shared in chats.
6. media_streaming_sidecar
7. messages - actual messages that are received or sent are stored in this table, each column of this table is discussed in Table 3.
8. messages_edits - messages that was edited
9. messages_fts - it is a virtual table (temporary fts table) and holds the copy of messages. This table is created to do the full text searches more efficiently.
10. messages_vcards - it keeps the data of vcards and sender of this card.
11. receipts - keeps the delivered and read timestamp of message.
12. sqlite_sequence - It keeps housekeeping data like Total messages, or group. WhatsApp uses this data internally for housekeeping.
13. status_list - List of status uploaded by WhatsApp users



| | |
|---|---|
| chat_list | (53) |
| frequents | (10) |
| group_participants | (75) |
| group_participants_history | (0) |
| media_refs | (235) |
| media_streaming_sidecar | (59) |
| messages | (126184) |
| messages_edits | (0) |
| messages_fts_content | (124719) |
| messages_fts_segdir | (48) |
| messages_fts_segments | (966) |
| messages_links | (952) |
| messages_quotes | (2731) |
| messages_vcards | (0) |
| messages_vcards_jids | (0) |
| props | (10) |
| receipts | (23724) |
| sqlite_sequence | (12) |
| status_list | (6) |

**Figure 4.3:** Database Schema of msgstore.db

#### 4.2.1.2 Cryptography elements in WhatsApp

**Database Security** To encrypt database, latest version of Whatspp uses crypt-12 algorithm. Crypt-12 is symmetric-key algorithm that uses same key to both encrypt and decrypt like AES. To decrypt databases, key is essential. Crypt-12 key is named as ‘key’ and stored in /data/media/0/WhatsApp/files directory. Unless you have key, decrypting WhatsApp database is very hard. One can use brute force attack to unlock database, but this method is quite unrealistic.

**End-to-End Encryption** WhatsApp client that is released after March 31, 2016 uses end-to-end encryption [1]. It means that WhatsApp messages, voice and video calls are encrypted all the way from sender to receiver; only these two parties can read the message. WhatsApp uses many keys to create session and encrypt/decrypt messages, audio, video calls, group messages, and live location information. Public keys are stored in *axolotl.db* file. These cryptographic keys are shown in Table 4.2

| **No.** | **Key Name** | **Key Type** | **Database name** | **Table name** | **Comments** |
|---|---|---|---|---|---|
| 1 | Identity Key Pair | Public | axolotl.db | identities | Long-term Curve25519 key pair; generated at install time |
| 2 | Signed Pre Key | Public | axolotl.db | signed_prekeys | Curve25519 key pair; first generated at install time but changes on a periodic timed basis. |
| 3 | One-Time Pre Keys | Public | axolotl.db | prekeys | Curve25519 key pair; first generated at install time but changes if required. |
| 4 | Root Key | Session | | | Curve25519 key pair; 32-byte value; chain key is derived from this key. |
| 5 | Chain Key | Session | | | 32-byte value; message key is derived from this key. |
| 6 | Message Key | Session | | | It is 80-byte key to encrypt message. Out of 80 bytes, AES-256 key takes 32 bytes, HMAC-SHA256 key takes 32 bytes and IV takes 16 bytes. [1] |

**Table 4.2:** Keys used in end-to-end encryption

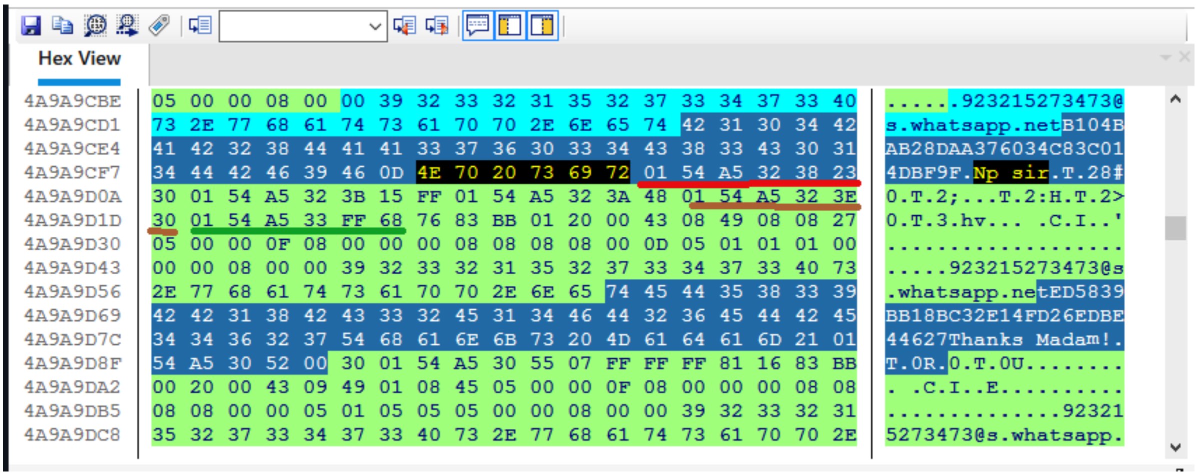
All three session keys are established during session initialization phase using public keys. WhatsApp uses AES256 in CBC mode to encrypt messages and HMAC-SHA256 for authentication.

There are few attacks on WhatsApp's end-to-end encryption which basically reads the keys from database and decrypt messages captured over network [2].

#### 4.2.1.3 WhatsApp Text Message Anatomy

One other way to retrieve a message is through manually analysis of bytes. Format of WhatsApp messages is:

- 27 bytes - chat participant info
- 30 bytes - Platform Info
- X bytes - Message
- 6 bytes - Timestamp of message sent
- 6 bytes - Timestamp of message delivered
- 6 bytes - Timestamp of message read



```
4A9A9CBE  05 00 00 08 00 00 39 32 33 32 31 35 32 37 33 34 37 33 40  ......923215273473@
4A9A9CD1  73 2E 77 68 61 74 73 61 70 70 2E 6E 65 74 42 31 30 34 42  s.whatsapp.netB104B
4A9A9CE4  41 42 32 38 44 41 41 33 37 36 30 33 34 43 38 33 43 30 31  AB28DAA376034C83C01
4A9A9CF7  34 44 42 46 39 46 0D 4E 70 20 73 69 72 01 54 A5 32 38 23  4DBF9F.Np sir.T.28#
4A9A9D0A  30 01 54 A5 32 3B 15 FF 01 54 A5 32 3A 48 01 54 A5 32 3E  0.T.2;...T.2:H.T.2>
4A9A9D1D  30 01 54 A5 33 FF 68 76 83 BB 01 20 00 43 08 49 08 08 27  0.T.3.hv... .C.I..'
4A9A9D30  05 00 00 0F 08 00 00 00 08 08 08 08 00 0D 05 01 01 01 00  ...................
4A9A9D43  00 00 08 00 00 39 32 33 32 31 35 32 37 33 34 37 33 40 73  .....923215273473@s
4A9A9D56  2E 77 68 61 74 73 61 70 70 2E 6E 65 74 45 44 35 38 33 39  .whatsapp.netED5839
4A9A9D69  42 42 31 38 42 43 33 32 45 31 34 46 44 32 36 45 44 42 45  BB18BC32E14FD26EDBE
4A9A9D7C  34 34 36 32 37 54 68 61 6E 6B 73 20 4D 61 64 61 6D 21 01  44627Thanks Madam!.
4A9A9D8F  54 A5 30 52 00 30 01 54 A5 30 55 07 FF FF FF 81 16 83 BB  T.0R.0.T.0U........
4A9A9DA2  00 20 00 43 09 49 01 08 45 05 00 00 0F 08 00 00 00 08 08  . .C.I..E..........
4A9A9DB5  08 08 00 00 05 01 05 05 05 00 00 08 00 00 39 32 33 32 31  ..............92321
4A9A9DC8  35 32 37 33 34 37 33 40 73 2E 77 68 61 74 73 61 70 70 2E  5273473@s.whatsapp.
```

**Figure 4.4:** WhatsApp Text Message format

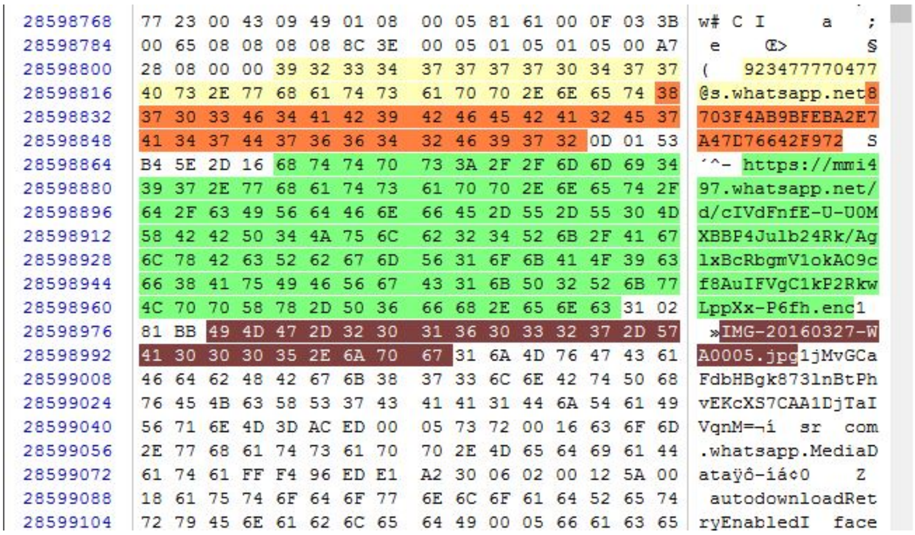
In Figure 4.4,

- Chat participant information is highlighted in cyan
- Platform information is highlighted in Blue
- Actual Message is highlighted in Black
- Message sent timestamp is underlined in red
- Message delivered timestamp is underlined in brown
- Message read timestamp is underlined in green.

#### 4.2.1.4 WhatsApp Media Message Anatomy

Like message, media data can be scraped from bytes too. Format of WhatsApp media message is:

- 27 byte - Chat participant Info
- 30 bytes - Platform Info
- 106 bytes - Attachment details
- 23 bytes - Attachment name
- 50 bytes - Attachment Path
- X bytes - Message body
- 6 bytes - Timestamp of message sent
- 6 bytes - Timestamp of message delivered
- 6 bytes - Timestamp of message read



```
28598768  77 23 00 43 09 49 01 08  00 05 81 61 00 0F 03 3B  w# C I     a   ;
28598784  00 65 08 08 08 08 8C 3E  00 05 01 05 01 05 00 A7   e     Œ>       §
28598800  28 08 00 00 39 32 33 34  37 37 37 37 30 34 37 37  (   923477770477
28598816  40 73 2E 77 68 61 74 73  61 70 70 2E 6E 65 74 38  @s.whatsapp.net8
28598832  37 30 33 46 34 41 42 39  42 46 45 42 41 32 45 37  703F4AB9BFEBA2E7
28598848  41 34 37 44 37 36 36 34  32 46 39 37 32 0D 01 53  A47D76642F972   S
28598864  B4 5E 2D 16 68 74 74 70  73 3A 2F 2F 6D 6D 69 34  ´^- https://mmi4
28598880  39 37 2E 77 68 61 74 73  61 70 70 2E 6E 65 74 2F  97.whatsapp.net/
28598896  64 2F 63 49 56 64 46 6E  66 45 2D 55 2D 55 30 4D  d/cIVdFnfE-U-U0M
28598912  58 42 42 50 34 4A 75 6C  62 32 34 52 6B 2F 41 67  XBBP4Julb24Rk/Ag
28598928  6C 78 42 63 52 62 67 6D  56 31 6F 6B 41 4F 39 63  lxBcRbgmV1okAO9c
28598944  66 38 41 75 49 46 56 67  43 31 6B 50 32 52 6B 77  f8AuIFVgC1kP2Rkw
28598960  4C 70 70 58 78 2D 50 36  66 68 2E 65 6E 63 31 02  LppXx-P6fh.enc1
28598976  81 BB 49 4D 47 2D 32 30  31 36 30 33 32 37 2D 57   »IMG-20160327-W
28598992  41 30 30 30 35 2E 6A 70  67 31 6A 4D 76 47 43 61  A0005.jpg1jMvGCa
28599008  46 64 62 48 42 67 6B 38  37 33 6C 6E 42 74 50 68  FdbHBgk873lnBtPh
28599024  76 45 4B 63 58 53 37 43  41 41 31 44 6A 54 61 49  vEKcXS7CAA1DjTaI
28599040  56 71 6E 4D 3D AC ED 00  05 73 72 00 16 63 6F 6D  VqnM=¬í  sr  com
28599056  2E 77 68 61 74 73 61 70  70 2E 4D 65 64 69 61 44  .whatsapp.MediaD
28599072  61 74 61 FF F4 96 ED E1  A2 30 06 02 00 12 5A 00  ataÿô-íá¢0    Z
28599088  18 61 75 74 6F 64 6F 77  6E 6C 6F 61 64 52 65 74   autodownloadRet
28599104  72 79 45 6E 61 62 6C 65  64 49 00 05 66 61 63 65  ryEnabledI  face
```

**Figure 4.5:** WhatsApp Media Message format

In Figure 4.5,

- Participant is shown in yellow. This image is sent to WhatsApp user.
- Platform information is highlighted in orange
- Attachment details is highlighted in green
- Attachment name is highlighted in maroon
- Timestamps

#### 4.2.1.5 Other Important Artifacts

Some important xml files that might help in artifact gathering during investigation are stored in /data/media/0/WhatsApp/shared_pref directory. These files are:

1. com.whatsapp_preferences.xml

    (a) This file keeps settings and preferences of WhatsApp application. Some of the most important preference include:

    i. ph - WhatsApp registered using this phone number
    ii. registration_jid - phone number assigned as registration unique number
    iii. data_usage_last_sync_date - timestamp of last sync using data
    iv. gdrive_account_name - email id of user if he/she keeps backup in gdrive
    v. gdrive_already_uploaded_bytes - number of bytes store in gdrive as backup, if it sets as 0, we can conclude that no backup was performed.
    vi. gdrive_last_successful_backup_timestamp:[email_id] - timestamp; when last backup was taken
    vii. gdrive_last_successful_backup_total_size:[email_id] - total size of last backup in bytes.
    viii. client_version_upgrade_timestamp - timestamp; when WhatsApp was last updated
    ix. push_name - name of user.
    x. phoneid_last_sync_timestamp - timestamp of last sync.

2. registration.RegisterPhone.xml

    - Users are registered on Whatspp through phone number. This file contains info like phone number, country code, and verification status.

3. keystore.xml

    - To connect to server, WhatsApp uses following two keys to perform handshake:

        - server_static_public
        - client_static_keypair

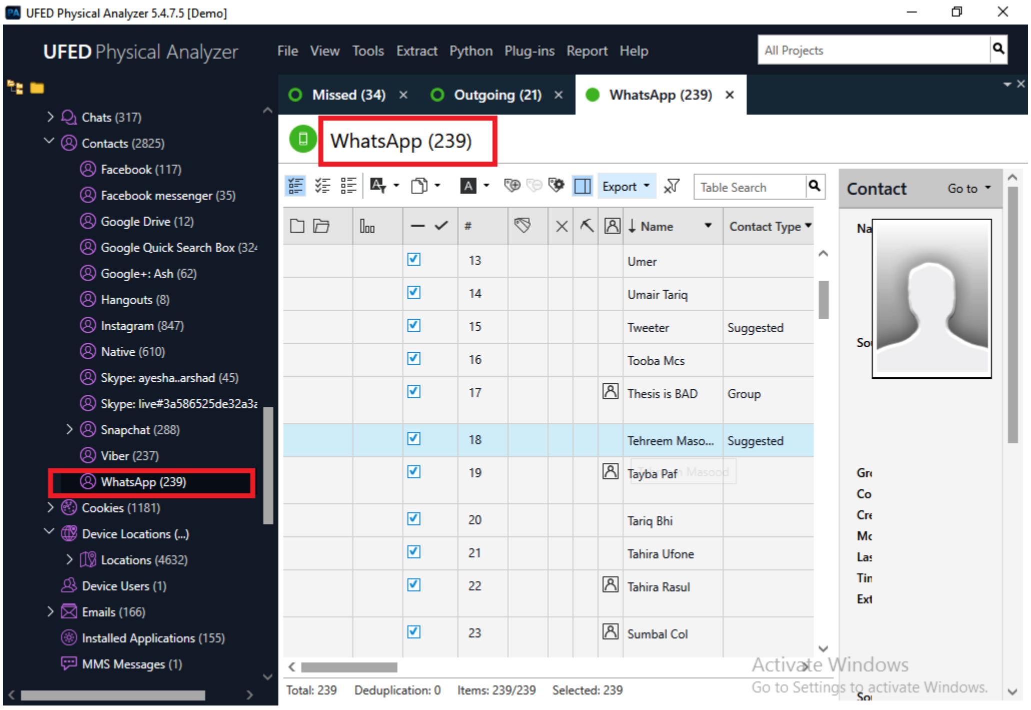
### 4.2.2 Analysis of Chat & media shared between participants on device - Data not Deleted

#### 4.2.2.1 Physical Acquisition

All the data can be retrieved from WhatsApp including chat messages, contacts, images, videos, group info. Many tools are available through we can retrieve WhatsApp data. In this research, I used three tools, Cellbrite UFED, WhatsApp Imager, Autopsy to get the data.

**4.2.2.1.1 Cellebrite UFED Physical Analyzer** Data that are retrieved using Cellebrite UFED Physical Analyzer are discussed in following sub sections.

**Contacts** All the contacts can be retrieved as shown in Figure 4.6. In Figure 4.6, it is shown that 239 WhatsApp contacts can be retrived. These contacts are mapped to database named wa.db.

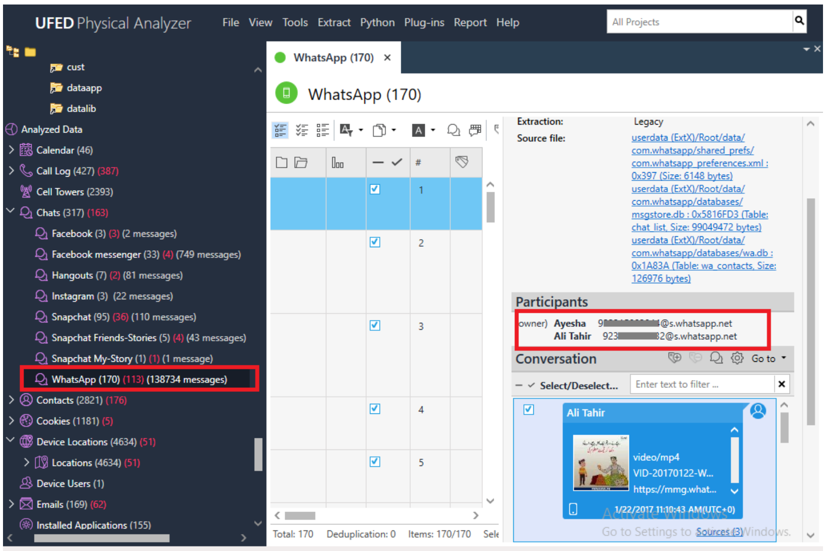


|  | # | Name | Contact Type |
|---|---|---|---|
| ☑ | 13 | Umer | |
| ☑ | 14 | Umair Tariq | |
| ☑ | 15 | Tweeter | Suggested |
| ☑ | 16 | Tooba Mcs | |
| ☑ | 17 | Thesis is BAD | Group |
| ☑ | 18 | Tehreem Maso... | Suggested |
| ☑ | 19 | Tayba Paf | |
| ☑ | 20 | Tariq Bhi | |
| ☑ | 21 | Tahira Ufone | |
| ☑ | 22 | Tahira Rasul | |
| ☑ | 23 | Sumbal Col | |

Total: 239 Deduplication: 0 Items: 239/239 Selected: 239

**Figure 4.6:** WhatsApp Contacts

**Messages** Messages including text, video, audio can be retrieved shared among a single receipt or in group. Figure 4.7 shows that total 138734 messages are retrieved, there are total 170 chats and out of these 170 chats 113 chats were deleted, but it can still be retrieved. So deleted messages can also be retrieved using Cellebrite UFED Physical Analyzer. Figure 4.7 also shows video message that is shared between Ayesha (Owner) and Ali Tahir.


UFED Physical Analyzer
File View Tools Extract Python Plug-ins Report Help
All Projects
cust
dataapp
datalib
Analyzed Data
Calendar (46)
Call Log (427) (387)
Cell Towers (2393)
Chats (317) (163)
Facebook (3) (3) (2 messages)
Facebook messenger (33) (4) (749 messages)
Hangouts (7) (2) (81 messages)
Instagram (3) (22 messages)
Snapchat (95) (36) (110 messages)
Snapchat Friends-Stories (5) (4) (43 messages)
Snapchat My-Story (1) (1) (1 message)
WhatsApp (170) (113) (138734 messages)
Contacts (2821) (176)
Cookies (1181) (5)
Device Locations (4634) (51)
Locations (4634) (51)
Device Users (1)
Emails (169) (62)
Installed Applications (155)
WhatsApp (170)
WhatsApp (170)
#
1
2
3
4
5
Total: 170 Deduplication: 0 Items: 170/170 Sele
Participants
Conversation
Go to
Select/Deselect...
Enter text to filter ...
Activate Windows
Go to Settings to activate Windows.

Extraction: Legacy

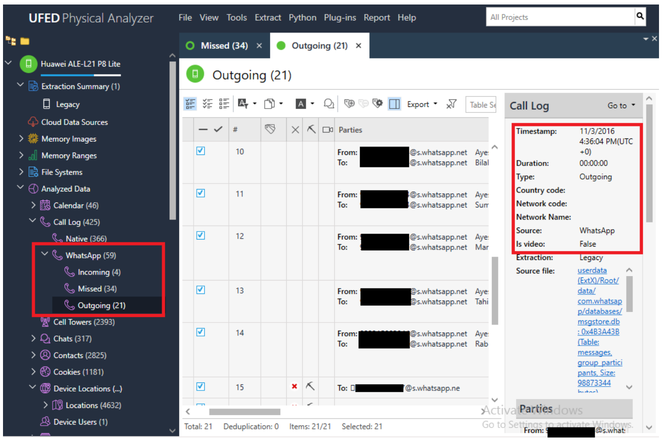
Source file: userdata (ExtX)/Root/data/com.whatsapp/shared_prefs/com.whatsapp_preferences.xml : 0x397 (Size: 6148 bytes) userdata (ExtX)/Root/data/com.whatsapp/databases/msgstore.db : 0x5816FD3 (Table: chat_list, Size: 99049472 bytes) userdata (ExtX)/Root/data/com.whatsapp/databases/wa.db : 0x1A83A (Table: wa_contacts, Size: 126976 bytes)

owner) Ayesha 9[redacted]@s.whatsapp.net
Ali Tahir 923[redacted]2@s.whatsapp.net

Ali Tahir
video/mp4
VID-20170122-W...
https://mmg.what...
1/22/2017 11:10:43 AM(UTC+0)
Sources (3)

**Figure 4.7:** WhatsApp Messages

**Calls** Using WhatsApp application, audio and video calls can also be placed. Data of when a call is placed and to whom user called or received a call is also important in forensic investigation. Cellebrite UFED Physical Analyzer can retrieve detailed data of incoming, outcome and missed call including timestamp, participants, duration, type (incoming, outgoing, missed) and mode (audio or video) as shown in Figure 4.8.



10 From: [redacted]@s.whatsapp.net Aye
To: @s.whatsapp.net Bilal

11 From: [redacted]@s.whatsapp.net Aye
To: @s.whatsapp.net Sum

12 From: [redacted]@s.whatsapp.net Aye
To: @s.whatsapp.net Mar

13 From: [redacted]@s.whatsapp.net Aye
To: @s.whatsapp.net Tahi

14 From: [redacted]@s.whatsapp.net Aye
To: @s.whatsapp.net Rab

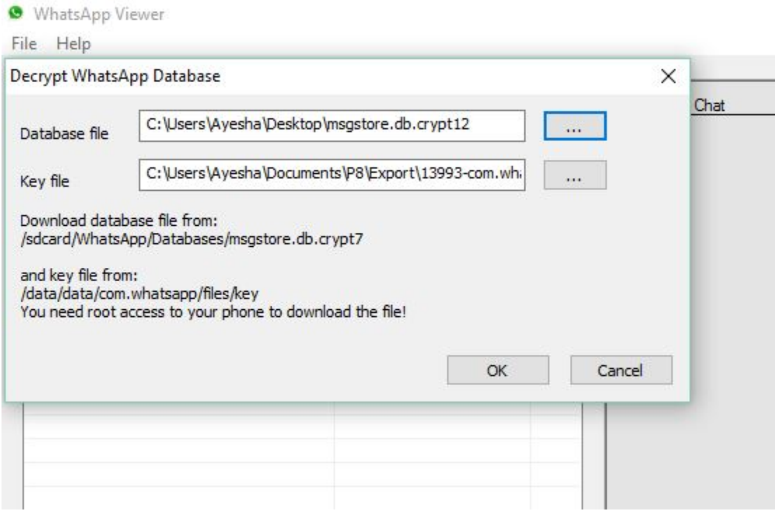
15 To: [redacted]@s.whatsapp.ne

Timestamp: 11/3/2016 4:36:04 PM(UTC+0)
Duration: 00:00:00
Type: Outgoing
Country code:
Network code:
Network Name:
Source: WhatsApp
Is video: False
Extraction: Legacy
Source file: userdata (ExtX)/Root/data/com.whatsapp/databases/msgstore.db : 0x4B3A43B (Table: messages, group_participants, Size: 98873344 bytes)

Parties
From: [redacted]@s.what

**Figure 4.8:** WhatsApp Call Log

**4.2.2.1.2 WhatsApp Viewer** To view data stored in Android backups made by WhatsApp application, WhatsApp Viewer tool can be used. It takes database file and key file to unlock encrypted database files as shown in Figure 4.9. After decrypting , it creates a decrypted database that can be open in WhatsApp Viewer. Chats are shown Figure 4.10. In left panel of chat window, we can see all the chats, their names and last message received or sent. In right panel, we can see messages.



Decrypt WhatsApp Database

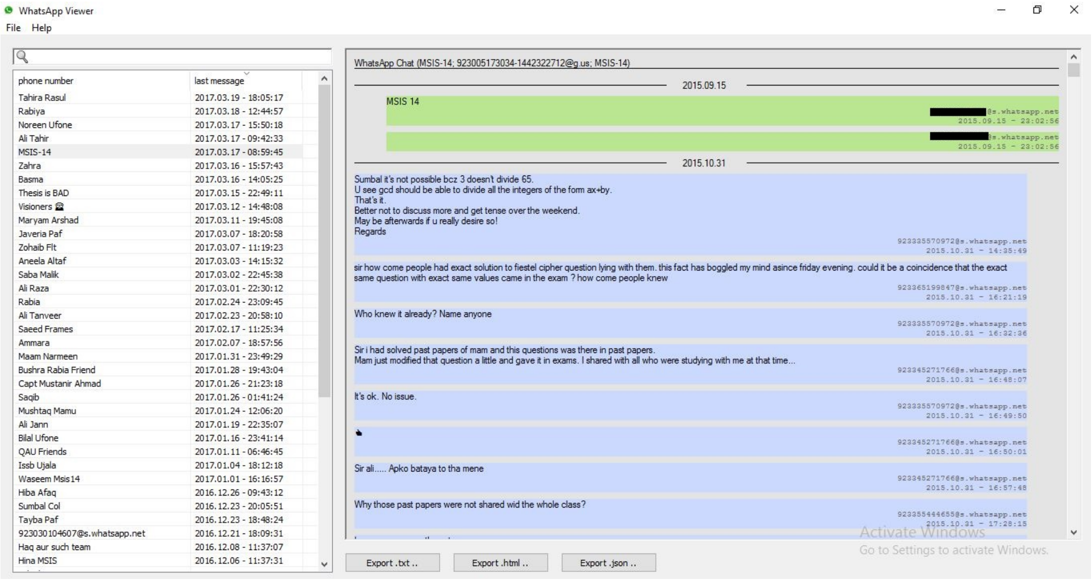
Database file: C:\Users\Ayesha\Desktop\msgstore.db.crypt12

Key file: C:\Users\Ayesha\Documents\P8\Export\13993-com.whi

Download database file from:
/sdcard/WhatsApp/Databases/msgstore.db.crypt7

and key file from:
/data/data/com.whatsapp/files/key
You need root access to your phone to download the file!

**Figure 4.9:** Decrypt database file using WhatsApp Viewer



| phone number | last message |
|---|---|
| Tahira Rasul | 2017.03.19 - 18:05:17 |
| Rabiya | 2017.03.18 - 12:44:57 |
| Noreen Ufone | 2017.03.17 - 15:50:18 |
| Ali Tahir | 2017.03.17 - 09:42:33 |
| MSIS-14 | 2017.03.17 - 08:59:45 |
| Zahra | 2017.03.16 - 15:57:43 |
| Basma | 2017.03.16 - 14:05:25 |
| Thesis is BAD | 2017.03.15 - 22:49:11 |
| Visioners ☎ | 2017.03.12 - 14:48:08 |
| Maryam Arshad | 2017.03.11 - 19:45:08 |
| Javeria Paf | 2017.03.07 - 18:20:58 |
| Zohaib Flt | 2017.03.07 - 11:19:23 |
| Aneela Altaf | 2017.03.03 - 14:15:32 |
| Saba Malik | 2017.03.02 - 22:45:38 |
| Ali Raza | 2017.03.01 - 22:30:12 |
| Rabia | 2017.02.24 - 23:09:45 |
| Ali Tanveer | 2017.02.23 - 20:58:10 |
| Saeed Frames | 2017.02.17 - 11:25:34 |
| Ammara | 2017.02.07 - 18:57:56 |
| Maam Narmeen | 2017.01.31 - 23:49:29 |
| Bushra Rabia Friend | 2017.01.28 - 19:43:04 |
| Capt Mustanir Ahmad | 2017.01.26 - 21:23:18 |
| Saqib | 2017.01.26 - 01:41:24 |
| Mushtaq Mamu | 2017.01.24 - 12:06:20 |
| Ali Jann | 2017.01.19 - 22:35:07 |
| Bilal Ufone | 2017.01.16 - 23:41:14 |
| QAU Friends | 2017.01.11 - 06:46:45 |
| Issb Ujala | 2017.01.04 - 18:12:18 |
| Waseem Msis14 | 2017.01.01 - 16:16:57 |
| Hiba Afaq | 2016.12.26 - 09:43:12 |
| Sumbal Col | 2016.12.23 - 20:05:51 |
| Tayba Paf | 2016.12.23 - 18:48:24 |
| 923030104607@s.whatsapp.net | 2016.12.21 - 18:09:31 |
| Haq aur such team | 2016.12.08 - 11:37:07 |
| Hina MSIS | 2016.12.06 - 11:37:31 |

WhatsApp Chat (MSIS-14; 923005173034-1442322712@g.us; MSIS-14)

2015.09.15

MSIS 14
[redacted]@s.whatsapp.net
2015.09.15 - 23:02:56

[redacted]@s.whatsapp.net
2015.09.15 - 23:02:56

2015.10.31

Sumbal it's not possible bcz 3 doesn't divide 65.
U see gcd should be able to divide all the integers of the form ax+by.
That's it.
Better not to discuss more and get tense over the weekend.
May be afterwards if u really desire so!
Regards
923335570972@s.whatsapp.net
2015.10.31 - 14:35:49

sir how come people had exact solution to fiestel cipher question lying with them. this fact has boggled my mind asince friday evening. could it be a coincidence that the exact same question with exact same values came in the exam ? how come people knew
923365199847@s.whatsapp.net
2015.10.31 - 16:21:19

Who knew it already? Name anyone
923335570972@s.whatsapp.net
2015.10.31 - 16:32:36

Sir i had solved past papers of mam and this questions was there in past papers.
Mam just modified that question a little and gave it in exams. I shared with all who were studying with me at that time...
923452717660@s.whatsapp.net
2015.10.31 - 16:48:07

It's ok. No issue.
923335570972@s.whatsapp.net
2015.10.31 - 16:49:50

923452717660@s.whatsapp.net
2015.10.31 - 16:50:01

Sir ali..... Apko bataya to tha mene
923452717660@s.whatsapp.net
2015.10.31 - 16:57:48

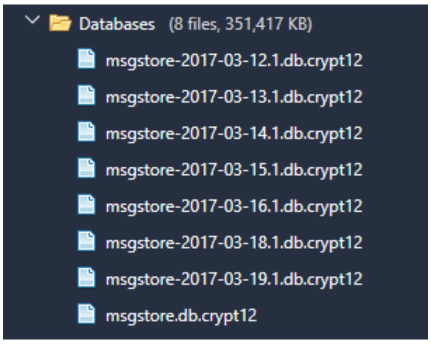
Why those past papers were not shared wid the whole class?
923355444655@s.whatsapp.net
2015.10.31 - 17:28:15

**Figure 4.10:** Chat retrieved using WhatsApp Viewer

#### 4.2.2.2 Logical Acquisition

One of technique to perform the logical acquisition is creating a backup of device. In last chapter, we have explained how to get backup of device using ADB commands. Biggest drawback of logical acquisition is that it does not hold deleted data, so deleted data cannot be retrieved using this method.

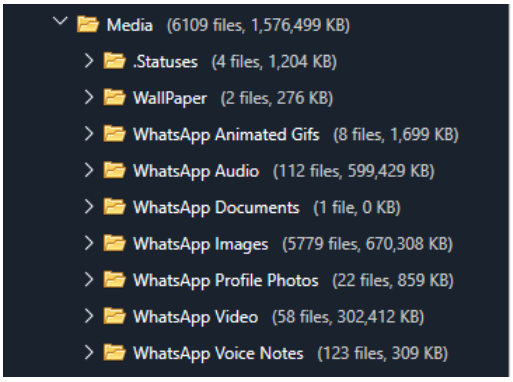
WhatsApp's media files and messages databases can be retrieved using this method. Database is encrypted using crypt12 algorithm. You need to have the key of your device to decrypt it. In Figure 4.11, msgstore databases are shown, these databases are encrypted using crypt12 algorithm and it's label shows the date of when the backup was taken.



```
Databases (8 files, 351,417 KB)
    msgstore-2017-03-12.1.db.crypt12
    msgstore-2017-03-13.1.db.crypt12
    msgstore-2017-03-14.1.db.crypt12
    msgstore-2017-03-15.1.db.crypt12
    msgstore-2017-03-16.1.db.crypt12
    msgstore-2017-03-18.1.db.crypt12
    msgstore-2017-03-19.1.db.crypt12
    msgstore.db.crypt12
```

**Figure 4.11:** WhatsApp database folder - Logical Acquisition

Media including status, wallpapers, animated gifs, audio, documents, images, profile photos, videos, voice notes can be retrieved using physical acquisition technique as shown in Figure 4.12.



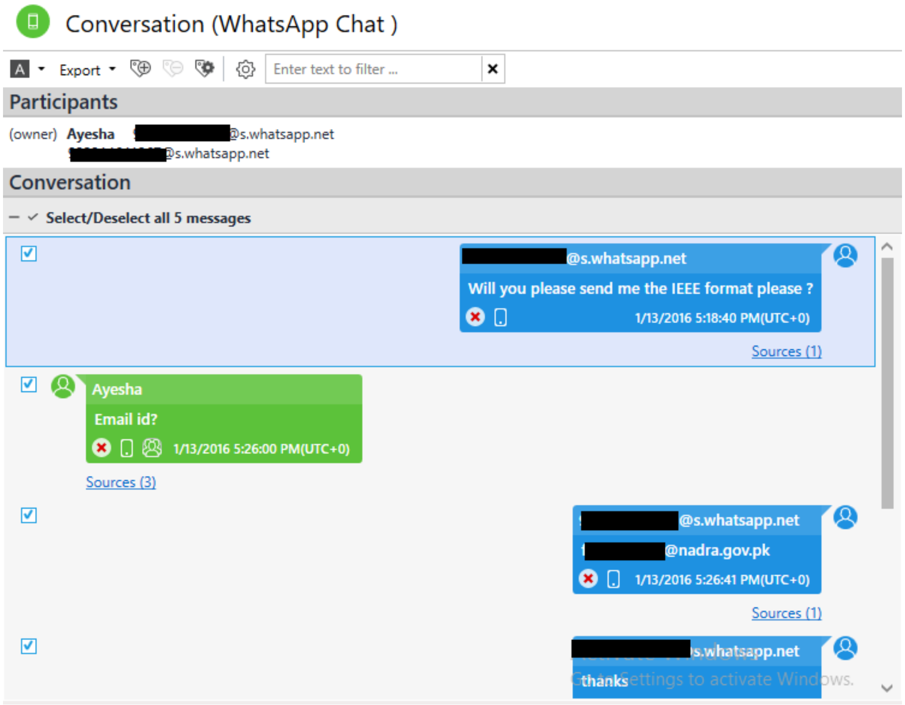
```
Media (6109 files, 1,576,499 KB)
  > .Statuses (4 files, 1,204 KB)
  > WallPaper (2 files, 276 KB)
  > WhatsApp Animated Gifs (8 files, 1,699 KB)
  > WhatsApp Audio (112 files, 599,429 KB)
  > WhatsApp Documents (1 file, 0 KB)
  > WhatsApp Images (5779 files, 670,308 KB)
  > WhatsApp Profile Photos (22 files, 859 KB)
  > WhatsApp Video (58 files, 302,412 KB)
  > WhatsApp Voice Notes (123 files, 309 KB)
```

**Figure 4.12:** WhatsApp media folder - Logical Acquisition

### 4.2.3 Analysis of Chat & media shared between participants on Android device - Data Deleted

Carving deleted data is very important as most of time it contains significant information that is crucial in forensic investigation. Deleted message sent or received to user is goldmine in investigation, and investigator must carve these messages during evidence gathering. During this research, we are able to find the deleted messages, contacts, and call records of WhatsApp messenger.

**Deleted Text Messages** Deleted text and media message can be extracted using Cellebrite UFED Physical Analyzer. As shown in Figure 4.13, complete detail of chat can be carved including participant info, timestamps, and actual text.



Conversation (WhatsApp Chat )

Participants

(owner) Ayesha [redacted]@s.whatsapp.net
[redacted]@s.whatsapp.net

Conversation

☑ [redacted]@s.whatsapp.net
Will you please send me the IEEE format please ?
1/13/2016 5:18:40 PM(UTC+0)
Sources (1)

☑ Ayesha
Email id?
1/13/2016 5:26:00 PM(UTC+0)
Sources (3)

☑ [redacted]@s.whatsapp.net
[redacted]@nadra.gov.pk
1/13/2016 5:26:41 PM(UTC+0)
Sources (1)

☑ [redacted]s.whatsapp.net
thanks

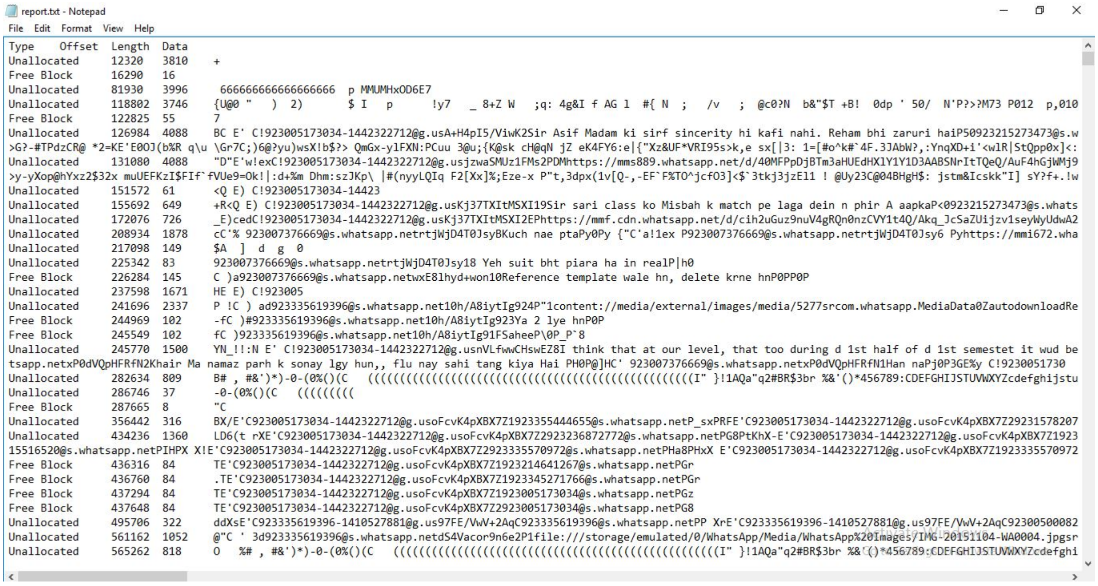
**Figure 4.13:** Deleted WhatsApp conversation

Deleted messages can be manually carved too. We can parse databases file to extract data from unallocated and free spaces. For this research, I used sqlparser tool to parse msgstore.db, this tool carves all the data from free and unallocated space. One of the drawbacks of manually extracting data is that we need to analysis file byte by byte to get the meaningful data. Extract meaningful data from free and unallocated data is time consuming.

To get data using sqlparser, we used this command:

```
sqlparse_cli.exe -f msgstore.db -o report.txt
```

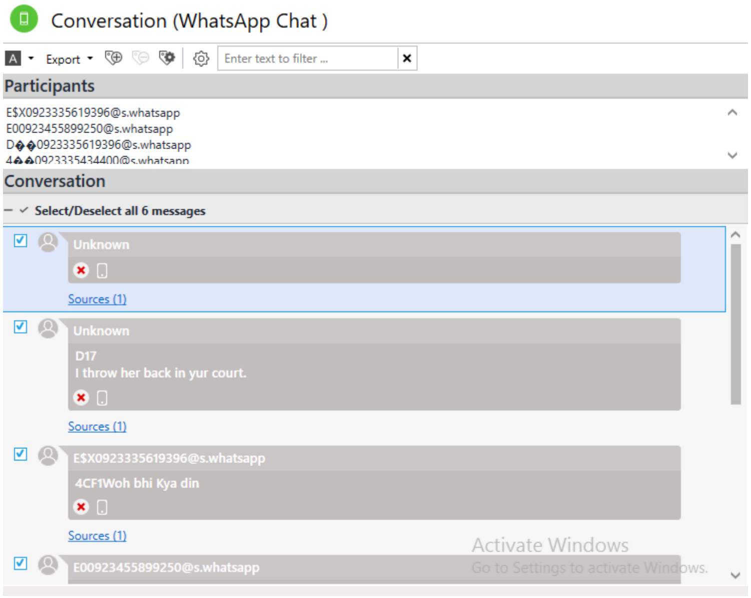
It creates a file as shown in Figure 4.14.



```
Type      Offset  Length  Data
Unallocated     12320   3810     +
Free Block      16290   16
Unallocated     81930   3996      666666666666666666  p MMUMHxOD6E7
Unallocated     118802  3746    {U@0 "    )  2)       $ I   p       !y7    _ 8+Z W    ;q: 4g&I f AG l  #{ N  ;    /v    ;  @c0?N  b&"$T +B!  0dp ' 50/  N'P?>?M73 P012  p,010
Free Block      122825  55      7
Unallocated     126984  4088    BC E' C!923005173034-1442322712@g.usA+H4pI5/ViwK2Sir Asif Madam ki sirf sincerity hi kafi nahi. Reham bhi zaruri haiP50923215273473@s.w
>G?-#TPdzCR@ *2=KE'E0OJ(b%R q\u \Gr7C;)6@?yu)wsX!b$?> QmGx-ylFXN:PCuu 3@u;{K@sk cH@qN jZ eK4FY6:e|{"Xz&UF*VRI95s>k,e sx[|3: 1=[#o^k#`4F.3JAbW?,:YnqXD+i'<wlR|StQpp0x]<:
Unallocated     131080  4088    "D"E'w!exC!923005173034-1442322712@g.usjzwaSMUz1FMs2PDMhttps://mms889.whatsapp.net/d/40MFPpDjBTm3aHUEdHXlY1Y1D3AABSNrItTQeQ/AuF4hGjWMj9
>y-yXop@hYxz2$32x muUEFKzI$FIf`fVUe9=Ok!|:d+%m Dhm:szJKp\ |#(nyyLQIq F2[Xx]%;Eze-x P"t,3dpx(1v[Q-,-EF`F%TO^jcfO3]<$`3tkj3jzEl1 ! @Uy23C@04BHgH$: jstm&Icskk"I] sY?f+.!w
Unallocated     151572  61      <Q E) C!923005173034-14423
Unallocated     155692  649     +R<Q E) C!923005173034-1442322712@g.usKj37TXItMSXI19Sir sari class ko Misbah k match pe laga dein n phir A aapkaP<0923215273473@s.whats
Unallocated     172076  726     _E)cedC!923005173034-1442322712@g.usKj37TXItMSXI2EPhttps://mmf.cdn.whatsapp.net/d/cih2uGuz9nuV4gRQn0nzCVY1t4Q/Akq_JcSaZUijzv1seyWyUdwA2
Unallocated     208934  1878    cC'% 923007376669@s.whatsapp.netrtjWjD4T0JsyBKuch nae ptaPy0Py {"C'a!1ex P923007376669@s.whatsapp.netrtjWjD4T0Jsy6 Pyhttps://mmi672.wha
Unallocated     217098  149     $A  ]  d  g  0
Unallocated     225342  83      923007376669@s.whatsapp.netrtjWjD4T0Jsy18 Yeh suit bht piara ha in realP|h0
Free Block      226284  145     C )a923007376669@s.whatsapp.netwxE8lhyd+won10Reference template wale hn, delete krne hnP0PP0P
Unallocated     237598  1671    HE E) C!923005
Unallocated     241696  2337    P !C ) ad923335619396@s.whatsapp.net10h/A8iytIg924P"1content://media/external/images/media/5277srcom.whatsapp.MediaData0ZautodownloadRe
Free Block      244969  102     -fC )#923335619396@s.whatsapp.net10h/A8iytIg923Ya 2 lye hnP0P
Free Block      245549  102     fC )923335619396@s.whatsapp.net10h/A8iytIg91FSaheeP\0P_P`8
Unallocated     245770  1500    YN_!!:N E' C!923005173034-1442322712@g.usnVLfwwCHswEZ8I think that at our level, that too during d 1st half of d 1st semestet it wud be
tsapp.netxP0dVQpHFRfN2Khair Ma namaz parh k sonay lgy hun,, flu nay sahi tang kiya Hai PH0P@]HC' 923007376669@s.whatsapp.netxP0dVQpHFRfN1Han naPj0P3GE%y C!9230051730
Unallocated     282634  809     B# , #&')*)-0-(0%()(C   ((((((((((((((((((((((((((((((((((((((((((((((((((((((I" }!1AQa"q2#BR$3br %&'()*456789:CDEFGHIJSTUVWXYZcdefghijstu
Unallocated     286746  37      -0-(0%()(C   (((((((((
Free Block      287665  8       "C
Unallocated     356442  316     BX/E'C923005173034-1442322712@g.usoFcvK4pXBX7Z1923355444655@s.whatsapp.netP_sxPRFE'C923005173034-1442322712@g.usoFcvK4pXBX7Z29231578207
Unallocated     434236  1360    LD6(t rXE'C923005173034-1442322712@g.usoFcvK4pXBX7Z2923236872772@s.whatsapp.netPG8PtKhX-E'C923005173034-1442322712@g.usoFcvK4pXBX7Z1923
15516520@s.whatsapp.netPIHPX X!E'C923005173034-1442322712@g.usoFcvK4pXBX7Z2923335570972@s.whatsapp.netPHa8PHxX E'C923005173034-1442322712@g.usoFcvK4pXBX7Z1923335570972
Free Block      436316  84      TE'C923005173034-1442322712@g.usoFcvK4pXBX7Z1923214641267@s.whatsapp.netPGr
Free Block      436760  84      .TE'C923005173034-1442322712@g.usoFcvK4pXBX7Z1923345271766@s.whatsapp.netPGr
Free Block      437294  84      TE'C923005173034-1442322712@g.usoFcvK4pXBX7Z1923005173034@s.whatsapp.netPGz
Free Block      437648  84      TE'C923005173034-1442322712@g.usoFcvK4pXBX7Z2923005173034@s.whatsapp.netPG8
Unallocated     495706  322     ddXsE'C923335619396-1410527881@g.us97FE/VwV+2AqC923335619396@s.whatsapp.netPP XrE'C923335619396-1410527881@g.us97FE/VwV+2AqC92300500082
Unallocated     561162  1052    @"C ' 3d923335619396@s.whatsapp.netdS4Vacor9n6e2P1file:///storage/emulated/0/WhatsApp/Media/WhatsApp%20Images/IMG-20151104-WA0004.jpgsr
Unallocated     565262  818     O   %# , #&')*)-0-(0%()(C   (((((((((((((((((((((((((((((((((((((((((((((((((((((((I" }!1AQa"q2#BR$3br %&'()*456789:CDEFGHIJSTUVWXYZcdefghi
```

**Figure 4.14:** Manually carving of deleted WhatsApp data

**Deleted Group Text Message** Message shared in group can also be carved through Cellebrite UFED Physical Analyzer as shown in Figure 4.15.



E$X0923335619396@s.whatsapp
E00923455899250@s.whatsapp
D��0923335619396@s.whatsapp
4��0923335434400@s.whatsapp

☑ Unknown
✖
Sources (1)

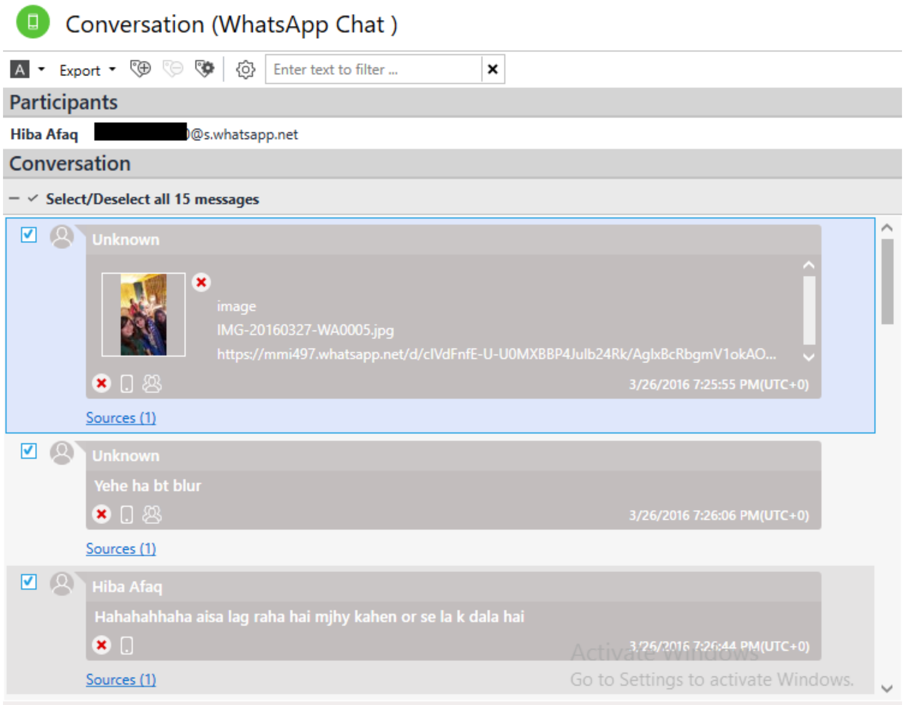
☑ Unknown
D17
I throw her back in yur court.
✖
Sources (1)

☑ E$X0923335619396@s.whatsapp
4CF1Woh bhi Kya din
✖
Sources (1)

☑ E00923455899250@s.whatsapp

**Figure 4.15:** Deleted group chat

**Deleted media message** Deleted media message can also be retrieved via Cellebrite UFED Physical Analyzer as shown in Figure 4.16.



Conversation (WhatsApp Chat )

Participants

Hiba Afaq [redacted]@s.whatsapp.net

Conversation

☑ Select/Deselect all 15 messages

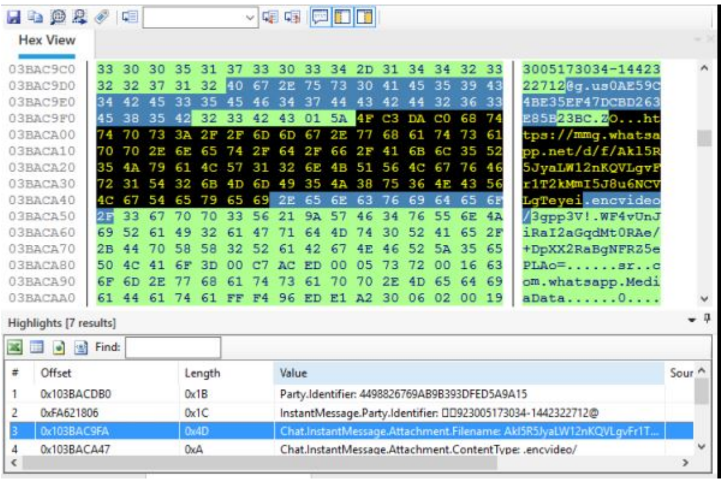
☑ Unknown
image
IMG-20160327-WA0005.jpg
https://mmi497.whatsapp.net/d/clVdFnfE-U-U0MXBBP4Julb24Rk/AglxBcRbgmV1okAO...
3/26/2016 7:25:55 PM(UTC+0)
Sources (1)

☑ Unknown
Yehe ha bt blur
3/26/2016 7:26:06 PM(UTC+0)
Sources (1)

☑ Hiba Afaq
Hahahahhaha aisa lag raha hai mjhy kahen or se la k dala hai
3/26/2016 7:26:44 PM(UTC+0)
Sources (1)

**Figure 4.16:** Deleted media message

Deleted media file can also be retrieved by reading the bytes as shown in Figure 4.17.



```
03BAC9C0  33 30 30 35 31 37 33 30 33 34 2D 31 34 34 32 33  3005173034-14423
03BAC9D0  32 32 37 31 32 40 67 2E 75 73 30 41 45 35 39 43  22712@g.us0AE59C
03BAC9E0  34 42 45 33 35 45 46 34 37 44 43 42 44 32 36 33  4BE35EF47DCBD263
03BAC9F0  45 38 35 42 32 33 42 43 01 5A 4F C3 DA C0 68 74  E85B23BC.ZO...ht
03BACA00  74 70 73 3A 2F 2F 6D 6D 67 2E 77 68 61 74 73 61  tps://mmg.whatsa
03BACA10  70 70 2E 6E 65 74 2F 64 2F 66 2F 41 6B 6C 35 52  pp.net/d/f/Akl5R
03BACA20  35 4A 79 61 4C 57 31 32 6E 4B 51 56 4C 67 76 46  5JyaLW12nKQVLgvF
03BACA30  72 31 54 32 6B 4D 6D 49 35 4A 38 75 36 4E 43 56  r1T2kMmI5J8u6NCV
03BACA40  4C 67 54 65 79 65 69 2E 65 6E 63 76 69 64 65 6F  LgTeyei.encvideo
03BACA50  2F 33 67 70 70 33 56 21 9A 57 46 34 76 55 6E 4A  /3gpp3V!.WF4vUnJ
03BACA60  69 52 61 49 32 61 47 71 64 4D 74 30 52 41 65 2F  iRaI2aGqdMt0RAe/
03BACA70  2B 44 70 58 58 32 52 61 42 67 4E 46 52 5A 35 65  +DpXX2RaBgNFRZ5e
03BACA80  50 4C 41 6F 3D 00 C7 AC ED 00 05 73 72 00 16 63  PLAo=......sr..c
03BACA90  6F 6D 2E 77 68 61 74 73 61 70 70 2E 4D 65 64 69  om.whatsapp.Medi
03BACAA0  61 44 61 74 61 FF F4 96 ED E1 A2 30 06 02 00 19  aData......0....
```

Highlights [7 results]

| # | Offset | Length | Value | Sour |
|---|---|---|---|---|
| 1 | 0x103BACDB0 | 0x1B | Party.Identifier: 4498826769AB9B393DFED5A9A15 | |
| 2 | 0xFA621806 | 0x1C | InstantMessage.Party.Identifier: □□923005173034-1442322712@ | |
| 3 | 0x103BAC9FA | 0x4D | Chat.InstantMessage.Attachment.Filename: Akl5R5JyaLW12nKQVLgvFr1T... | |
| 4 | 0x103BACA47 | 0xA | Chat.InstantMessage.Attachment.ContentType: .encvideo/ | |

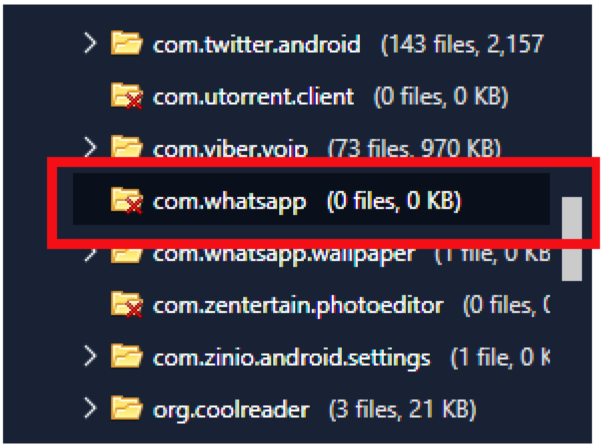
**Figure 4.17:** Carve deleted video by reading bytes

Similarly, we can extract audio/video call history and document/vcard/location shared between sender and receiver.

### 4.2.4 Analysis of Chat & media shared between participants on device - Application Deleted

Many times culprit deletes the app and thinks that he has removed all the evidence; most of time this type of thinking is not true. Much of this data can be extracted by craving the memory. Many tools are available in market that can carve data from free and unallocated space and represent in meaningful information. In this research, I have tried to extract the data after deleting WhatsApp application from Android device.

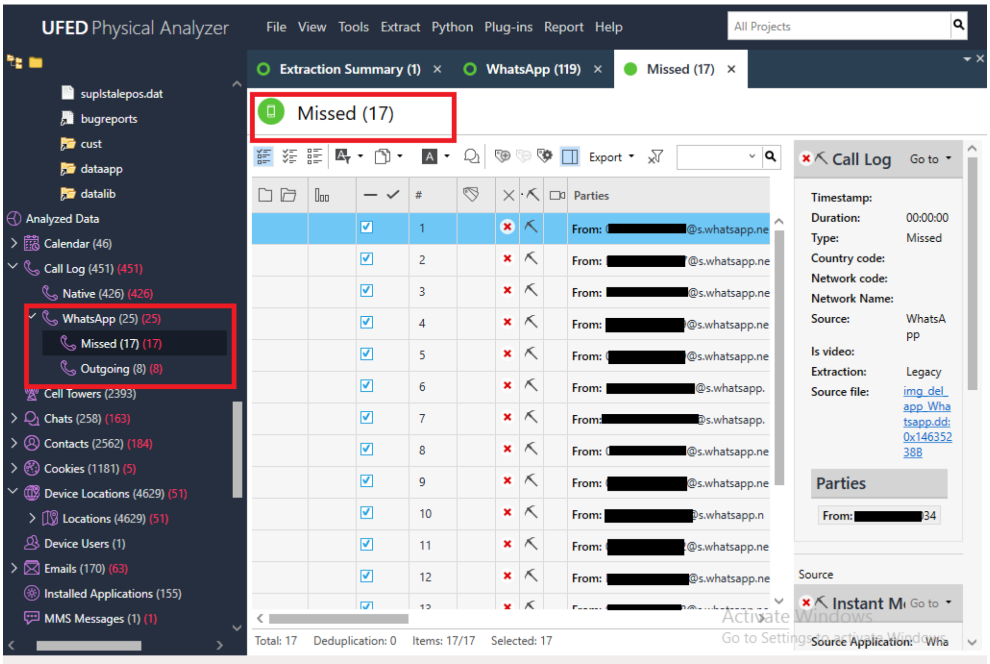
**WhatsApp Application Folder** WhatsApp folder can easily be retrieved, many tools are available that can carve data stored in directory hierarchy and present in meaningful form. In Figure 4.18, we can see deleted WhatsApp folder marked with red cross.



```
> com.twitter.android (143 files, 2,157
  com.utorrent.client (0 files, 0 KB)
> com.viber.voip (73 files, 970 KB)
  com.whatsapp (0 files, 0 KB)
> com.whatsapp.wallpaper (1 file, 0 KB
  com.zentertain.photoeditor (0 files, (
> com.zinio.android.settings (1 file, 0 K
> org.coolreader (3 files, 21 KB)
```

**Figure 4.18:** Extracted WhatsApp folder after deleting application

**Extracting deleted Calls** All calls including incoming, outgoing, missed can be carved from memory. Deleted missed calls are shown in Figure 4.19 and outgoing calls are shown in 4.20.



| # | Parties |
|---|---|
| 1 | From: [redacted]@s.whatsapp.ne |
| 2 | From: [redacted]@s.whatsapp.ne |
| 3 | From: [redacted]@s.whatsapp.ne |
| 4 | From: [redacted]@s.whatsapp.ne |
| 5 | From: [redacted]@s.whatsapp.ne |
| 6 | From: [redacted]@s.whatsapp. |
| 7 | From:[redacted]s.whatsapp. |
| 8 | From: [redacted]@s.whatsapp.ne |
| 9 | From: [redacted]@s.whatsapp.ne |
| 10 | From: [redacted]s.whatsapp.n |
| 11 | From: [redacted]@s.whatsapp.ne |
| 12 | From: [redacted]@s.whatsapp.ne |

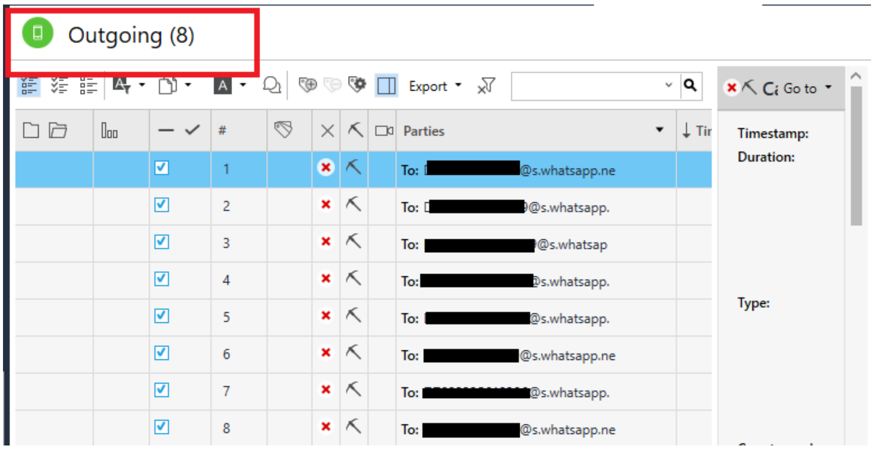
Timestamp:
Duration: 00:00:00
Type: Missed
Country code:
Network code:
Network Name:
Source: WhatsApp
Is video:
Extraction: Legacy
Source file: img_del_app_Whatsapp.dd:0x146352388

From: [redacted]34

Source Application: Wha

Total: 17 Deduplication: 0 Items: 17/17 Selected: 17

**Figure 4.19:** Extracted WhatsApp missed call after deleting application

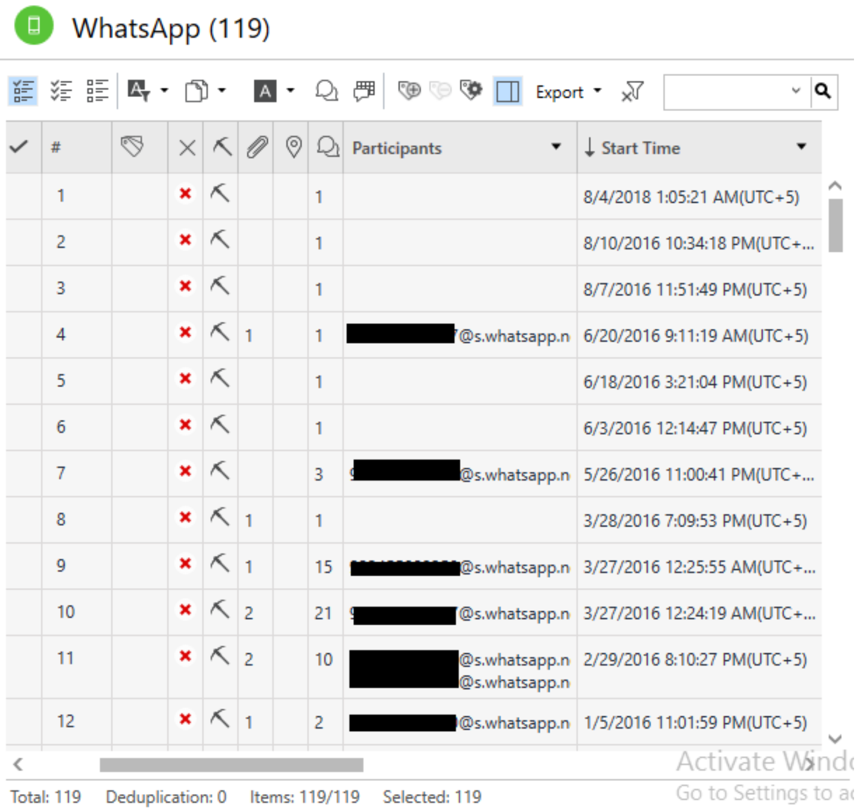


| # | Parties |
|---|---|
| 1 | To: [redacted]@s.whatsapp.ne |
| 2 | To: [redacted]@s.whatsapp. |
| 3 | To: [redacted]@s.whatsap |
| 4 | To:[redacted]s.whatsapp. |
| 5 | To: [redacted]@s.whatsapp. |
| 6 | To: [redacted]@s.whatsapp.ne |
| 7 | To: [redacted]@s.whatsapp. |
| 8 | To: [redacted]@s.whatsapp.ne |

Timestamp:
Duration:
Type:

**Figure 4.20:** Extracted WhatsApp outgoing call after deleting application

**Extracting deleted message** Like calls, messages can also be extracted using Cellebrite UFED Physical Analyzer or by byte by byte scanning of msgstore.db as discussed previously. Deleted messages are shown in Figure 4.21.



WhatsApp (119)

| ✓ | # | 🏷 | ✕ | ⛏ | 📎 | 📍 | 🗨 | Participants | ↓ Start Time |
|---|---|---|---|---|---|---|---|---|---|
| | 1 | | ✖ | ⛏ | | | 1 | | 8/4/2018 1:05:21 AM(UTC+5) |
| | 2 | | ✖ | ⛏ | | | 1 | | 8/10/2016 10:34:18 PM(UTC+... |
| | 3 | | ✖ | ⛏ | | | 1 | | 8/7/2016 11:51:49 PM(UTC+5) |
| | 4 | | ✖ | ⛏ | 1 | | 1 | [redacted]@s.whatsapp.n | 6/20/2016 9:11:19 AM(UTC+5) |
| | 5 | | ✖ | ⛏ | | | 1 | | 6/18/2016 3:21:04 PM(UTC+5) |
| | 6 | | ✖ | ⛏ | | | 1 | | 6/3/2016 12:14:47 PM(UTC+5) |
| | 7 | | ✖ | ⛏ | | | 3 | [redacted]@s.whatsapp.n | 5/26/2016 11:00:41 PM(UTC+... |
| | 8 | | ✖ | ⛏ | 1 | | 1 | | 3/28/2016 7:09:53 PM(UTC+5) |
| | 9 | | ✖ | ⛏ | 1 | | 15 | [redacted]@s.whatsapp.n | 3/27/2016 12:25:55 AM(UTC+... |
| | 10 | | ✖ | ⛏ | 2 | | 21 | [redacted]@s.whatsapp.n | 3/27/2016 12:24:19 AM(UTC+... |
| | 11 | | ✖ | ⛏ | 2 | | 10 | [redacted]@s.whatsapp.n<br>@s.whatsapp.n | 2/29/2016 8:10:27 PM(UTC+5) |
| | 12 | | ✖ | ⛏ | 1 | | 2 | [redacted]@s.whatsapp.n | 1/5/2016 11:01:59 PM(UTC+5) |

Total: 119    Deduplication: 0    Items: 119/119    Selected: 119

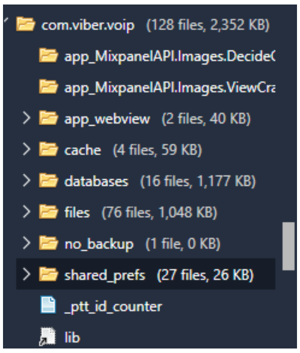
**Figure 4.21:** Extracted WhatsApp messages after deleting application

## 4.3 Forensic Investigation of Viber

In this section, artifacts that can be extracted from Viber and are important in forensic investigation are discussed.

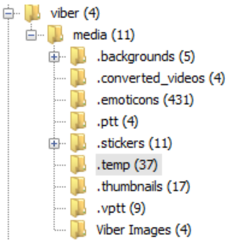
### 4.3.1 Viber Anatomy

Viber application creates a folder on internal memory of device to store application and user related data, and it is named as *com.viber.voip*. Path to this folder is /data/data/com.viber.voip and its directory structure is shown in Figure ??. Viber stores media including audio and video in /data/media/0/viber/media folder as shown in 4.23. These two folders hold important Viber related data and it is very crucial in forensic investigation.



```
com.viber.voip   (128 files, 2,352 KB)
    app_MixpanelAPI.Images.DecideC
    app_MixpanelAPI.Images.ViewCra
  > app_webview   (2 files, 40 KB)
  > cache   (4 files, 59 KB)
  > databases   (16 files, 1,177 KB)
  > files   (76 files, 1,048 KB)
  > no_backup   (1 file, 0 KB)
  > shared_prefs   (27 files, 26 KB)
    _ptt_id_counter
    lib
```

**Figure 4.22:** Directory structure of Viber data folder



```
viber (4)
  media (11)
    .backgrounds (5)
    .converted_videos (4)
    .emoticons (431)
    .ptt (4)
    .stickers (11)
    .temp (37)
    .thumbnails (17)
    .vptt (9)
    Viber Images (4)
```

**Figure 4.23:** Directory structure of Viber media folder

#### 4.3.1.1 Viber Databases

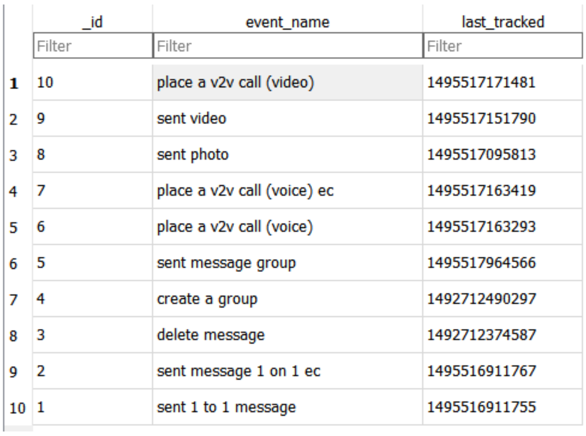
Viber stores user and application related data in databases and these are found in /data/data/com.viber.voip/databases directory. One most important thing to be noted is that Viber does not encrypt its database and data can easily be retrieved. Following are list of Viber database:

- appboy
- apptimize
- apptimize_tmp
- google_app_measurement_local
- mixpanel
- viber_data
- viber_messages
- viber_prefs

Most important databases are viber_data and viber_messages and will be discussed in detail in next section.

**4.3.1.1.1 Anatomy of viber_data** viber_data stores information related to user's contacts including phone book contacts, Viber contacts, and block numbers. Apart from that, this database also stores call history log. List of tables and their description are discussed as follow:

1. blockednumbers - contact numbers that have been blocked.
2. call - stored information of incoming, outgoing and missed call. Important columns are:
   - number - contact number of callee/caller.
   - viber_call - True if call type is Viber otherwise false.
   - Viber_call_type - It is assigned 1 if Viber user calls other Viber user; it is assigned 2 if Viber user calls non-viber user.
   - duration - Duration of call in seconds.
   - date - This value represents the epoch timestamp of when the call was placed.
3. phonebookcontact - When the first time user opens Viber, it scrapes all the phone book contacts and stores them in this database. Important columns are:
   - display_name - Name assigned to user
   - viber - True if Viber's user otherwise false
   - joined_data - epoch timestamp of when the user joined the viber
4. phonebookdata - This table basically holds the phone number of user that has one to one relationship with phonebookcontact table.
5. vibernumber - This table stores information of User's contacts who use Viber application. It stores unique id assigned to each Viber member, other important data is as follow:
   - canonized_number - Phone number of Viber user.
   - photo - Unique id of user's display picture.
   - member_id, viber_id - Unique Id assigned to each Viber user.



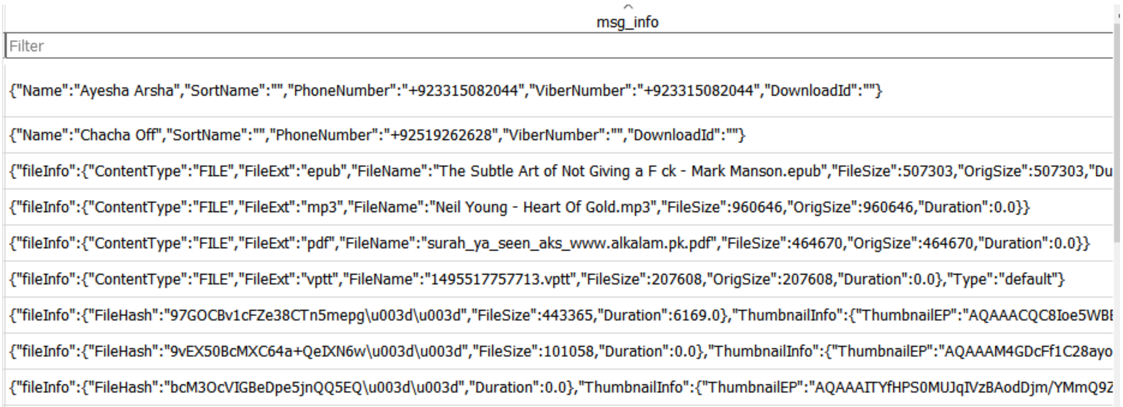
| | _id | event_name | last_tracked |
|---|---|---|---|
| | Filter | Filter | Filter |
| 1 | 10 | place a v2v call (video) | 1495517171481 |
| 2 | 9 | sent video | 1495517151790 |
| 3 | 8 | sent photo | 1495517095813 |
| 4 | 7 | place a v2v call (voice) ec | 1495517163419 |
| 5 | 6 | place a v2v call (voice) | 1495517163293 |
| 6 | 5 | sent message group | 1495517964566 |
| 7 | 4 | create a group | 1492712490297 |
| 8 | 3 | delete message | 1492712374587 |
| 9 | 2 | sent message 1 on 1 ec | 1495516911767 |
| 10 | 1 | sent 1 to 1 message | 1495516911755 |

**Figure 4.24:** Content of adx table - viber_messages Database

**4.3.1.1.2 Anatomy of viber_messages** This database schema stores information of chat, groups, and messages that are sent or received. From forensic point of view, this database has significant respect as it contains valuable artifacts that might help in investigation. Detail of each table is discussed below:

1. adx - It is most important table as it contains timestamp of all events. These events are shown in Figure 4.24.
2. applications - It stores information of Viber application like version, last updated date.
3. conversation - Conversation can be between two participants, or a group. This table stores information of each unique conversation including group id, recipient and date.
4. messages - It contains all messages that are sent or received. Detailed information is:

    - _id - Unique Id of message
    - address - Unique Viber id assigned to each user; This column is mapped to *viber_id* column of vibernumber table in viber_data database.
    - date - Linux epoch timestamp of when message was sent/received.
    - type - If message type is outgoing, its value is 1. If message type is incoming, its value is 0.
    - body - actual message body; it can be text message, uri of media and map, contact information, and audio/video call record.
    - extra_uri - uri of media.
    - extra_mime - Type of media; audio, sound, video, image, file, location, call, share_contact, notif, text, deleted.



```
{"Name":"Ayesha Arsha","SortName":"","PhoneNumber":"+923315082044","ViberNumber":"+923315082044","DownloadId":""}
{"Name":"Chacha Off","SortName":"","PhoneNumber":"+92519262628","ViberNumber":"","DownloadId":""}
{"fileInfo":{"ContentType":"FILE","FileExt":"epub","FileName":"The Subtle Art of Not Giving a F ck - Mark Manson.epub","FileSize":507303,"OrigSize":507303,"Du
{"fileInfo":{"ContentType":"FILE","FileExt":"mp3","FileName":"Neil Young - Heart Of Gold.mp3","FileSize":960646,"OrigSize":960646,"Duration":0.0}}
{"fileInfo":{"ContentType":"FILE","FileExt":"pdf","FileName":"surah_ya_seen_aks_www.alkalam.pk.pdf","FileSize":464670,"OrigSize":464670,"Duration":0.0}}
{"fileInfo":{"ContentType":"FILE","FileExt":"vptt","FileName":"1495517757713.vptt","FileSize":207608,"OrigSize":207608,"Duration":0.0},"Type":"default"}
{"fileInfo":{"FileHash":"97GOCBv1cFZe38CTn5mepg\u003d\u003d","FileSize":443365,"Duration":6169.0},"ThumbnailInfo":{"ThumbnailEP":"AQAAACQC8Ioe5WB
{"fileInfo":{"FileHash":"9vEX50BcMXC64a+QeIXN6w\u003d\u003d","FileSize":101058,"Duration":0.0},"ThumbnailInfo":{"ThumbnailEP":"AQAAAM4GDcFf1C28ayo
{"fileInfo":{"FileHash":"bcM3OcVIGBeDpe5jnQQ5EQ\u003d\u003d","Duration":0.0},"ThumbnailInfo":{"ThumbnailEP":"AQAAAITYfHPS0MUJqIVzBAodDjm/YMmQ9Z
```

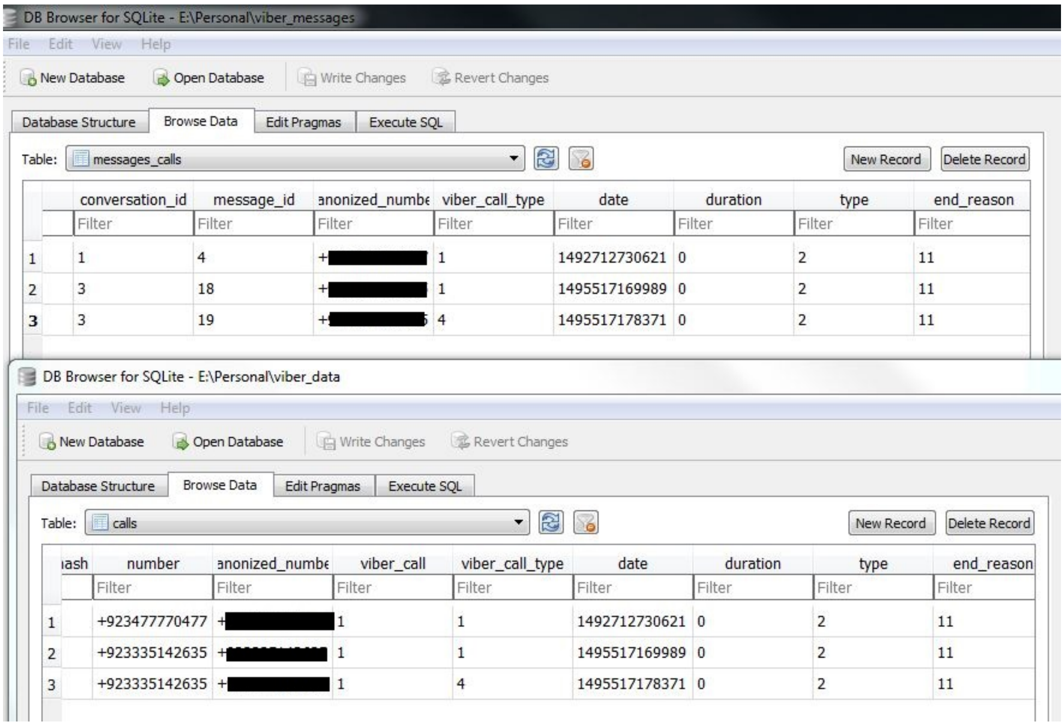
**Figure 4.25:** Content of msg_info column - viber_messages Database

- msg_info - Extra information of shared media as shown in Figure 4.25.
- group_id - This is populated if message is shared in group and its value corresponds to unique id of group.
- conversation_id - Unique id of conversation; mapped to *id* column of conversations table.
- read_message_time - Numeric value + message sent timestamp = Linux epoch timestamp of when message was read.

5. messages_call - It has same information as *calls* table in viber_data database as shown in Figure 4.26. Call data is redundant so if it is corrupted at one location, it can still be recovered from other location.
6. participants_info - It stores information of all users that are part of conversation, either in group or between two participants. It gets most of its data from viber_data database.
7. participants - It maps each participant to conversations in which they have contributed.

#### 4.3.1.2 Cryptography elements in Viber

**Database Security** Viber stores most of its information in database files, but these files are not encrypted. If attacker can get hold of these files, alot of valuable information can be leaked. But if forensic investigator seizes these files, he can bring out a great deal of meaningful information without wasting time on decryption process.



| | conversation_id | message_id | anonized_numbe | viber_call_type | date | duration | type | end_reason |
|---|---|---|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | 1 | 4 | +[redacted] | 1 | 1492712730621 | 0 | 2 | 11 |
| 2 | 3 | 18 | +[redacted] | 1 | 1495517169989 | 0 | 2 | 11 |
| 3 | 3 | 19 | +[redacted] | 4 | 1495517178371 | 0 | 2 | 11 |

| | ıash | number | anonized_numbe | viber_call | viber_call_type | date | duration | type | end_reason |
|---|---|---|---|---|---|---|---|---|---|
| | | Filter | Filter | Filter | Filter | Filter | Filter | Filter | Filter |
| 1 | | +923477770477 | +[redacted] | 1 | 1 | 1492712730621 | 0 | 2 | 11 |
| 2 | | +923335142635 | +[redacted] | 1 | 1 | 1495517169989 | 0 | 2 | 11 |
| 3 | | +923335142635 | +[redacted] | 1 | 4 | 1495517178371 | 0 | 2 | 11 |

**Figure 4.26:** Content of messages_call and calls table

**End-to-End Encryption** Viber has provided feature of end-to-end encryption from version 6.0, and Viber claims that all calls, one-on-one messages, group messages, media sharing and secondary devices are all secure from end-to-end [2].

Viber's cryptographic protocol uses concept of "double ratchet". During installation phase, each primary Viber device is assigned a single 256-bit Curve-25519 key-pair known as *ID Key*. Private key of this pair is stored in user's device and public key is stored in Viber's server.

In addition to ID key, each Viber client also generates a series of *PreKeys*. Each PreKey has two 256-bit Curve-25519 key-pairs called *Handshake Key* and *Ratchet Key*. Like ID key, private keys of are stored on device and public key is uploaded to Viber's server.

Elliptic-Curve Diffie-Hellman key-exchange algorithm is used to exchange keys between participants.

To encrypt a message, keys shown in Table 4.3 are derived or used.

| No. | Key Name | Keys involved | Algorithm used |
|---|---|---|---|
| 1 | Rootkey | ID key and Handshake key of both participants are used to derive this key. | SHA-256 |
| 2 | TempKey | RootKey and Ratchet key of both participants are used to derive this key. | HMAC_SHA256 |
| 3 | SessionKey | It takes tempKey and message | HMAC_SHA256 |

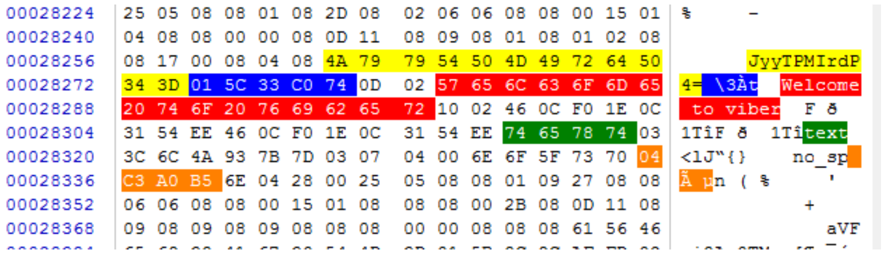
**Table 4.3:** Keys used in end-to-end encryption in Viber

**Media files** - Each media file is encrypted with symmetric Salsa20 key, this key is stored in encryption_params column of messages table in viber_messages database.

**Group messages** - A common secret key (salsa20) is used among all members of group to secure messages.

### 4.3.1.3 Viber Text Message Anatomy - Sent

Another way to retrieve a message is through manually analyzing of bytes. Format of Viber messages is shown in Figure 4.27.



```
00028224  25 05 08 08 01 08 2D 08  02 06 06 08 08 00 15 01  %     -
00028240  04 08 08 00 00 08 0D 11  08 09 08 01 08 01 02 08
00028256  08 17 00 08 04 08 4A 79  79 54 50 4D 49 72 64 50        JyyTPMIrdP
00028272  34 3D 01 5C 33 C0 74 0D  02 57 65 6C 63 6F 6D 65  4= \3Àt  Welcome
00028288  20 74 6F 20 76 69 62 65  72 10 02 46 0C F0 1E 0C   to viber  F ð
00028304  31 54 EE 46 0C F0 1E 0C  31 54 EE 74 65 78 74 03  1TîF ð  1Tîtext
00028320  3C 6C 4A 93 7B 7D 03 07  04 00 6E 6F 5F 73 70 04  <lJ"{}     no_sp
00028336  C3 A0 B5 6E 04 28 00 25  05 08 08 01 09 27 08 08  Ã µn ( %     '
00028352  06 06 08 08 00 15 01 08  08 08 00 2B 08 0D 11 08            +
00028368  09 08 09 08 09 08 08 08  00 00 08 08 08 61 56 46              aVF
```

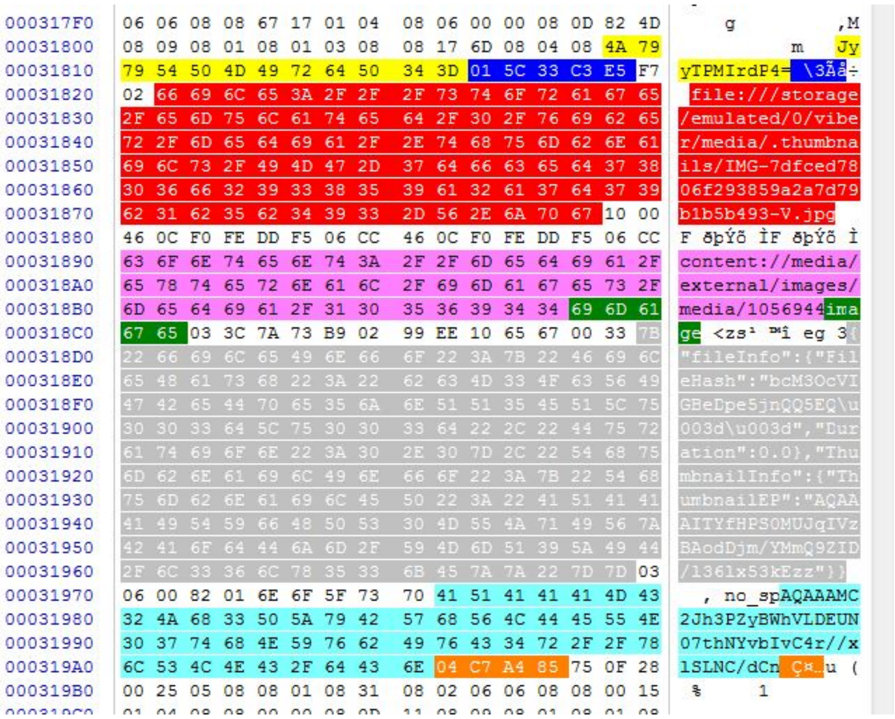
**Figure 4.27:** Viber Text Message format - Sent

In Figure 4.27,

- Message participant address is highlighted in yellow.
- Message received timestamp is highlighted in blue.
- Message body is highlighted in red.
- Extra mime information is underlined in green.
- Message read timestamp is highlighted in orange.

#### 4.3.1.4 Viber Media Message Anatomy

Like text messages, media messages can also be carved from memory, its format is shown in figure 4.28.



```
000317F0  06 06 08 08 67 17 01 04  08 06 00 00 08 0D 82 4D      g        ,M
00031800  08 09 08 01 08 01 03 08  08 17 6D 08 04 08 4A 79            m   Jy
00031810  79 54 50 4D 49 72 64 50  34 3D 01 5C 33 C3 E5 F7  yTPMIrdP4= \3Ãå÷
00031820  02 66 69 6C 65 3A 2F 2F  2F 73 74 6F 72 61 67 65   file:///storage
00031830  2F 65 6D 75 6C 61 74 65  64 2F 30 2F 76 69 62 65  /emulated/0/vibe
00031840  72 2F 6D 65 64 69 61 2F  2E 74 68 75 6D 62 6E 61  r/media/.thumbna
00031850  69 6C 73 2F 49 4D 47 2D  37 64 66 63 65 64 37 38  ils/IMG-7dfced78
00031860  30 36 66 32 39 33 38 35  39 61 32 61 37 64 37 39  06f293859a2a7d79
00031870  62 31 62 35 62 34 39 33  2D 56 2E 6A 70 67 10 00  b1b5b493-V.jpg
00031880  46 0C F0 FE DD F5 06 CC  46 0C F0 FE DD F5 06 CC  F ðþÝõ ÌF ðþÝõ Ì
00031890  63 6F 6E 74 65 6E 74 3A  2F 2F 6D 65 64 69 61 2F  content://media/
000318A0  65 78 74 65 72 6E 61 6C  2F 69 6D 61 67 65 73 2F  external/images/
000318B0  6D 65 64 69 61 2F 31 30  35 36 39 34 34 69 6D 61  media/1056944ima
000318C0  67 65 03 3C 7A 73 B9 02  99 EE 10 65 67 00 33 7B  ge <zs¹ ™î eg 3{
000318D0  22 66 69 6C 65 49 6E 66  6F 22 3A 7B 22 46 69 6C  "fileInfo":{"Fil
000318E0  65 48 61 73 68 22 3A 22  62 63 4D 33 4F 63 56 49  eHash":"bcM3OcVI
000318F0  47 42 65 44 70 65 35 6A  6E 51 51 35 45 51 5C 75  GBeDpe5jnQQ5EQ\u
00031900  30 30 33 64 5C 75 30 30  33 64 22 2C 22 44 75 72  003d\u003d","Dur
00031910  61 74 69 6F 6E 22 3A 30  2E 30 7D 2C 22 54 68 75  ation":0.0},"Thu
00031920  6D 62 6E 61 69 6C 49 6E  66 6F 22 3A 7B 22 54 68  mbnailInfo":{"Th
00031930  75 6D 62 6E 61 69 6C 45  50 22 3A 22 41 51 41 41  umbnailEP":"AQAA
00031940  41 49 54 59 66 48 50 53  30 4D 55 4A 71 49 56 7A  AITYfHPS0MUJqIVz
00031950  42 41 6F 64 44 6A 6D 2F  59 4D 6D 51 39 5A 49 44  BAodDjm/YMmQ9ZID
00031960  2F 6C 33 36 6C 78 35 33  6B 45 7A 7A 22 7D 7D 03  /l36lx53kEzz"}}
00031970  06 00 82 01 6E 6F 5F 73  70 41 51 41 41 41 4D 43    , no_spAQAAAMC
00031980  32 4A 68 33 50 5A 79 42  57 68 56 4C 44 45 55 4E  2Jh3PZyBWhVLDEUN
00031990  30 37 74 68 4E 59 76 62  49 76 43 34 72 2F 2F 78  07thNYvbIvC4r//x
000319A0  6C 53 4C 4E 43 2F 64 43  6E 04 C7 A4 85 75 0F 28  lSLNC/dCn Ç¤…u (
000319B0  00 25 05 08 08 01 08 31  08 02 06 06 08 08 00 15   %      1
000319C0  01 04 08 08 00 00 08 0D  11 08 09 08 01 08 01 08
```

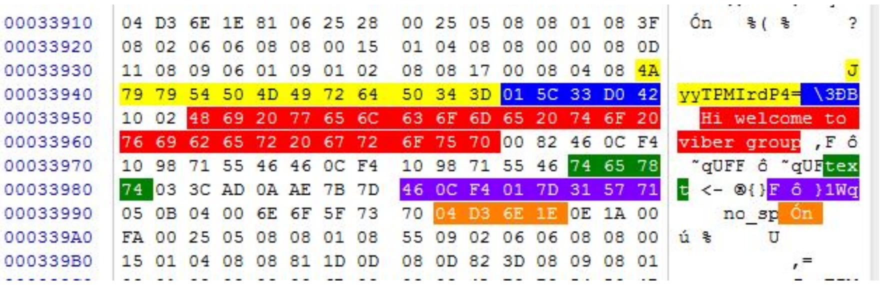
Figure 4.28: Viber media message format - Received

In Figure 4.28,

- Message participant address is highlighted in yellow.
- Message received timestamp is highlighted in blue.
- Message body is highlighted in red.
- Extra uri is highlighted in purple.
- Extra mime information is underlined in green.
- Message info is highlighted in grey.
- Encryption parameter is highlighted in cyan.
- Message read timestamp is highlighted in orange.

#### 4.3.1.5 Viber Group Message Anatomy - Sent

Format of message sent to a group is shown in Figure 4.29.



```
00033910  04 D3 6E 1E 81 06 25 28  00 25 05 08 08 01 08 3F   Ón   %( %     ?
00033920  08 02 06 06 08 08 00 15  01 04 08 08 00 00 08 0D
00033930  11 08 09 06 01 09 01 02  08 08 17 00 08 04 08 4A                  J
00033940  79 79 54 50 4D 49 72 64  50 34 3D 01 5C 33 D0 42  yyTPMIrdP4= \3ÐB
00033950  10 02 48 69 20 77 65 6C  63 6F 6D 65 20 74 6F 20    Hi welcome to
00033960  76 69 62 65 72 20 67 72  6F 75 70 00 82 46 0C F4  viber group ,F ô
00033970  10 98 71 55 46 46 0C F4  10 98 71 55 46 74 65 78   ˜qUFF ô ˜qUFtex
00033980  74 03 3C AD 0A AE 7B 7D  46 0C F4 01 7D 31 57 71  t <- ®{}F ô }1Wq
00033990  05 0B 04 00 6E 6F 5F 73  70 04 D3 6E 1E 0E 1A 00      no_sp Ón
000339A0  FA 00 25 05 08 08 01 08  55 09 02 06 06 08 08 00  ú %      U
000339B0  15 01 04 08 08 81 1D 0D  08 0D 82 3D 08 09 08 01            ,=
```

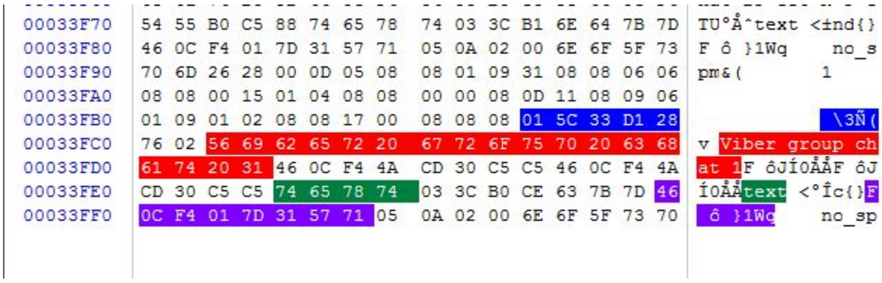
**Figure 4.29:** Viber group message format - sent

In Figure 4.29,

- Message participant address is highlighted in yellow.
- Message received timestamp is highlighted in blue.
- Message body is highlighted in red.
- Extra mime information is underlined in green.
- Group id is highlighted in purple.
- Message read timestamp is highlighted in orange.

#### 4.3.1.6 Viber Group Message Anatomy - Received

Messages that are received on group can be manually carved from memory. Format of text message received on Viber is shown in Figure 4.30



```
00033F70  54 55 B0 C5 88 74 65 78  74 03 3C B1 6E 64 7B 7D  TU°Å^text <±nd{}
00033F80  46 0C F4 01 7D 31 57 71  05 0A 02 00 6E 6F 5F 73  F ô }1Wq    no_s
00033F90  70 6D 26 28 00 0D 05 08  08 01 09 31 08 08 06 06  pm&(        1
00033FA0  08 08 00 15 01 04 08 08  00 00 08 0D 11 08 09 06
00033FB0  01 09 01 02 08 08 17 00  08 08 08 01 5C 33 D1 28              \3Ñ(
00033FC0  76 02 56 69 62 65 72 20  67 72 6F 75 70 20 63 68  v Viber group ch
00033FD0  61 74 20 31 46 0C F4 4A  CD 30 C5 C5 46 0C F4 4A  at 1F ôJÍ0ÅÅF ôJ
00033FE0  CD 30 C5 C5 74 65 78 74  03 3C B0 CE 63 7B 7D 46  Í0ÅÅtext <°Îc{}F
00033FF0  0C F4 01 7D 31 57 71 05  0A 02 00 6E 6F 5F 73 70   ô }1Wq     no_sp
```

**Figure 4.30:** Viber group message format - Received

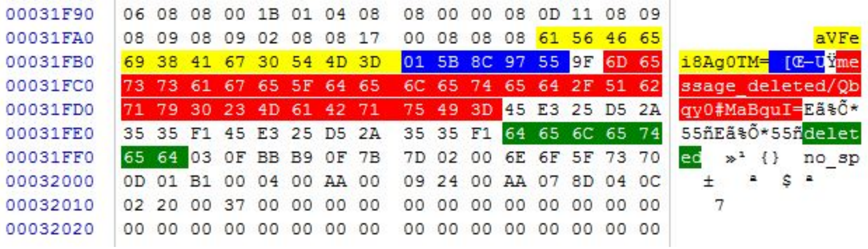
In Figure 4.30,

- Message received timestamp is highlighted in blue.

- Actual Message is highlighted in red.
- Extra mime information is underlined in green.
- Group id is highlighted in purple.

#### 4.3.1.7 Viber Secret Message Anatomy

Viber provides feature of secret messages that destroy itself after set timer. After deleting secret messages, Viber keeps some of its data in database file (viber_messages) but overwrite message body with 'Deleted message' text. We can carve secret message information from database file as shown in Figure 4.31



```
00031F90  06 08 08 00 1B 01 04 08  08 00 00 08 0D 11 08 09
00031FA0  08 09 08 09 02 08 08 17  00 08 08 08 61 56 46 65              aVFe
00031FB0  69 38 41 67 30 54 4D 3D  01 5B 8C 97 55 9F 6D 65  i8Ag0TM= [Œ—UŸme
00031FC0  73 73 61 67 65 5F 64 65  6C 65 74 65 64 2F 51 62  ssage_deleted/Qb
00031FD0  71 79 30 23 4D 61 42 71  75 49 3D 45 E3 25 D5 2A  qy0#MaBquI=Eã%Õ*
00031FE0  35 35 F1 45 E3 25 D5 2A  35 35 F1 64 65 6C 65 74  55ñEã%Õ*55ñdelet
00031FF0  65 64 03 0F BB B9 0F 7B  7D 02 00 6E 6F 5F 73 70  ed  »¹ {}  no_sp
00032000  0D 01 B1 00 04 00 AA 00  09 24 00 AA 07 8D 04 0C    ±   ª  $ ª
00032010  02 20 00 37 00 00 00 00  00 00 00 00 00 00 00 00     7
00032020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

**Figure 4.31:** Viber secret message format

In Figure 4.31,

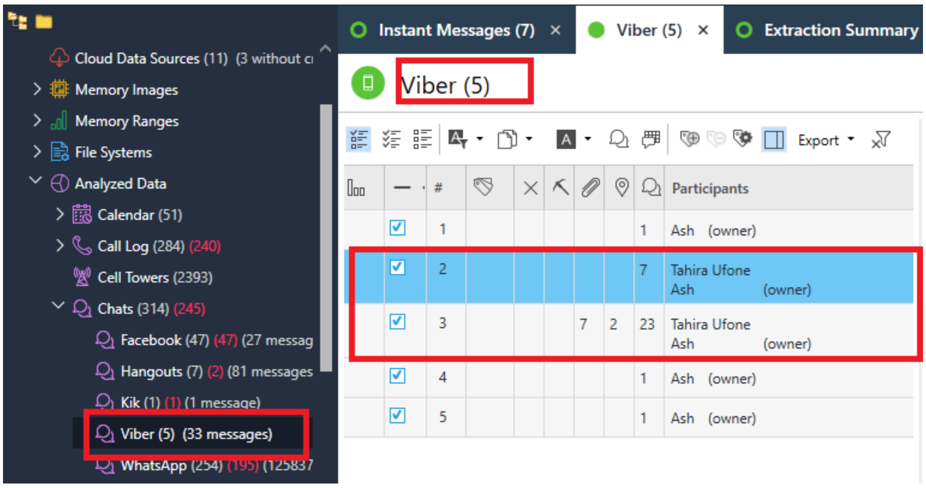
- Message participant address is highlighted in yellow.
- Message received timestamp is highlighted in blue.
- Message body that is overwritten with 'Message Deleted' text is highlighted in red.
- Extra mime information is underlined in green.

### 4.3.2 Analysis of Viber Chat & media shared between participants on device - Data not Deleted

Viber contacts, calls, and messages can be extracted using many tools or it can be carved by manually analyzing of database files or by reading memory byte-by-byte as discussed in previous section. In this section, I have used Cellebrite UFED Physical analyzer to retrieve information from Viber application.

**Viber Conversations detail** In Figure 4.32, we can see that total of 33 messages are shared in 5 conversations.

In conversation number 3 that was happened between 'ash' and 'Tahira Ufone', total of 23 messages were shared, and among these 23 messages 7 messages were media files and 2 were location information.



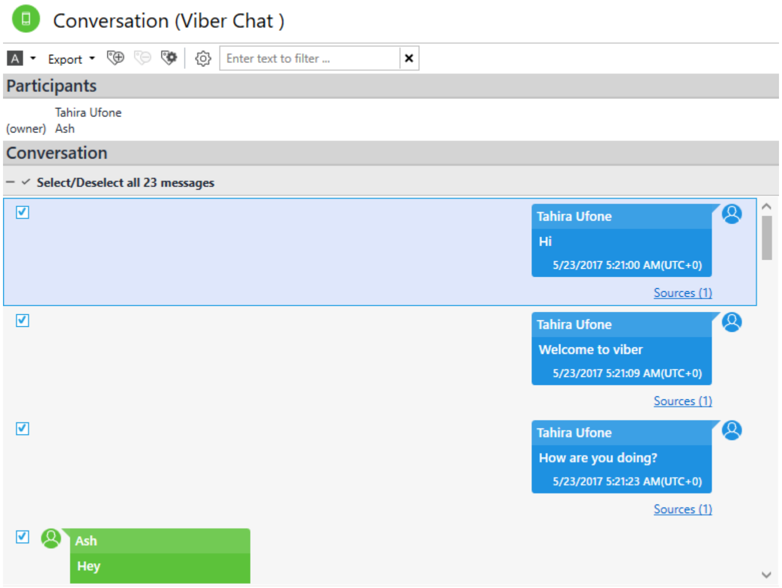
Cloud Data Sources (11) (3 without c
Memory Images
Memory Ranges
File Systems
Analyzed Data
Calendar (51)
Call Log (284) (240)
Cell Towers (2393)
Chats (314) (245)
Facebook (47) (47) (27 messag
Hangouts (7) (2) (81 messages
Kik (1) (1) (1 message)
Viber (5) (33 messages)
WhatsApp (254) (195) (125837

Viber (5)

| | — | # | | ✕ | | | | | Participants |
|---|---|---|---|---|---|---|---|---|---|
| | ☑ | 1 | | | | | | 1 | Ash (owner) |
| | ☑ | 2 | | | | | | 7 | Tahira Ufone<br>Ash (owner) |
| | ☑ | 3 | | | | 7 | 2 | 23 | Tahira Ufone<br>Ash (owner) |
| | ☑ | 4 | | | | | | 1 | Ash (owner) |
| | ☑ | 5 | | | | | | 1 | Ash (owner) |

**Figure 4.32:** Viber Conversation Details

**Text Messages** One-to-one text messages that are extracted using Cellebrite UFED Physical Analyzer are shown in Figure 4.33. In this figure, we can see all 23 messages shared between two participants are recovered.


Conversation (Viber Chat )

Participants

Tahira Ufone
(owner) Ash

Conversation

– ✓ Select/Deselect all 23 messages

☑ Tahira Ufone
Hi
5/23/2017 5:21:00 AM(UTC+0)
Sources (1)

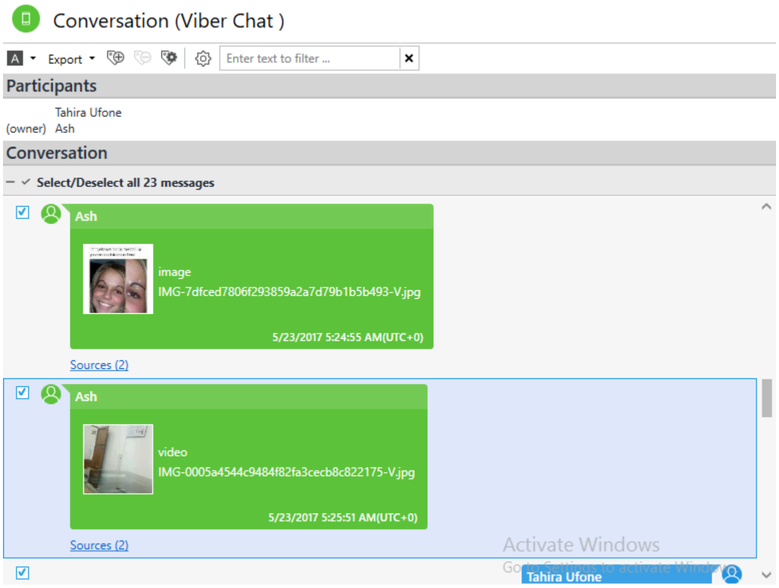
☑ Tahira Ufone
Welcome to viber
5/23/2017 5:21:09 AM(UTC+0)
Sources (1)

☑ Tahira Ufone
How are you doing?
5/23/2017 5:21:23 AM(UTC+0)
Sources (1)

☑ Ash
Hey

**Figure 4.33:** Viber one-to-one text message

**Media Messages** Like text messages, media including image, audio, video, document, and sound can be retrieved. In Figure 4.34, we can see audio and video message shared by owner.


Conversation (Viber Chat )

Participants

Tahira Ufone
(owner) Ash

Conversation

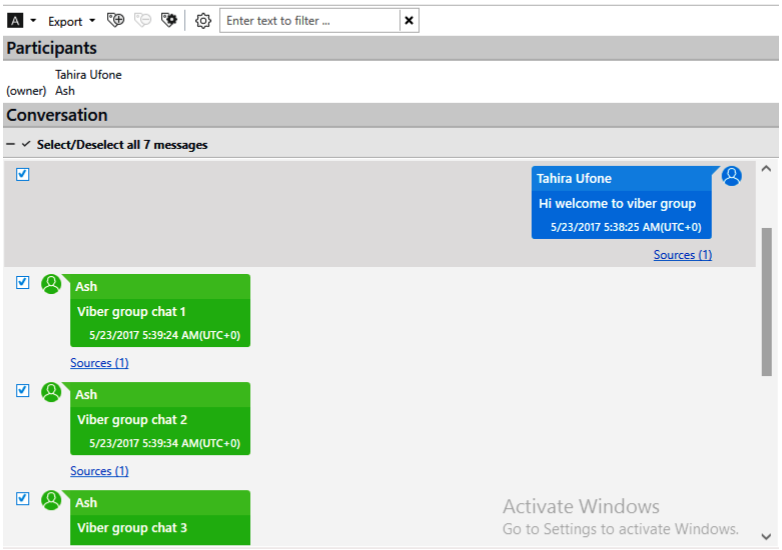
– ✓ Select/Deselect all 23 messages

☑ Ash
image
IMG-7dfced7806f293859a2a7d79b1b5b493-V.jpg
5/23/2017 5:24:55 AM(UTC+0)
Sources (2)

☑ Ash
video
IMG-0005a4544c9484f82fa3cecb8c822175-V.jpg
5/23/2017 5:25:51 AM(UTC+0)
Sources (2)

☑ Tahira Ufone

**Figure 4.34:** Viber one-to-one media message



Participants

Tahira Ufone
(owner) Ash

Conversation

Select/Deselect all 7 messages

☑ Tahira Ufone
Hi welcome to viber group
5/23/2017 5:38:25 AM(UTC+0)
Sources (1)

☑ Ash
Viber group chat 1
5/23/2017 5:39:24 AM(UTC+0)
Sources (1)

☑ Ash
Viber group chat 2
5/23/2017 5:39:34 AM(UTC+0)
Sources (1)

☑ Ash
Viber group chat 3

**Figure 4.35:** Viber group message

**Viber Group Messages** Like one-to-one text and media messages, Viber group messages can also be extracted using Cellebrite UFED Physical Analyzer as shown in Figure 4.35

**Call and Viber Contact Details** All Viber contacts including profile picture and last seen timestamp information can be recovered from Android device using Cellebrite UFED Physical Analyzer.

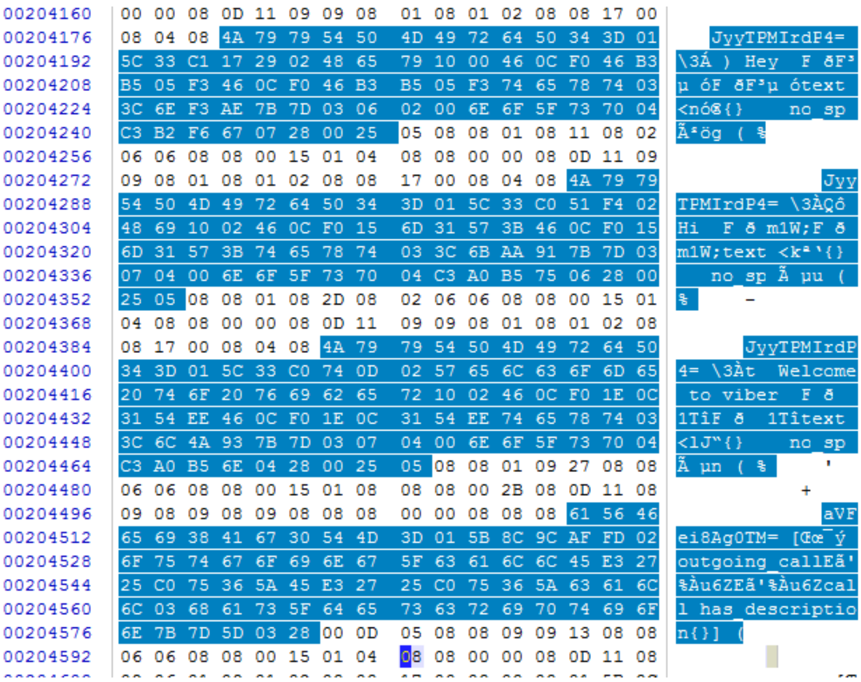
Similarly, Viber incoming/outgoing/missed call information with timestamps and duration of call can be extracted by reading database files or using many forensic tools.

### 4.3.3 Analysis of Viber Chat & media shared between participants on Android device - Data Deleted

One of the challenges, a forensic investigator faces is to carve the data that has been deleted. Most of the time, culprit thinks that if he removes data from phone/device, he can get rid of evidence. But much of this deleted data can be carved from user device; so is the case with Viber messages.

Cellebrite UFED Physical Analyzer does not carve Viber's deleted data and I have to manually analyze databases files to extract deleted data.

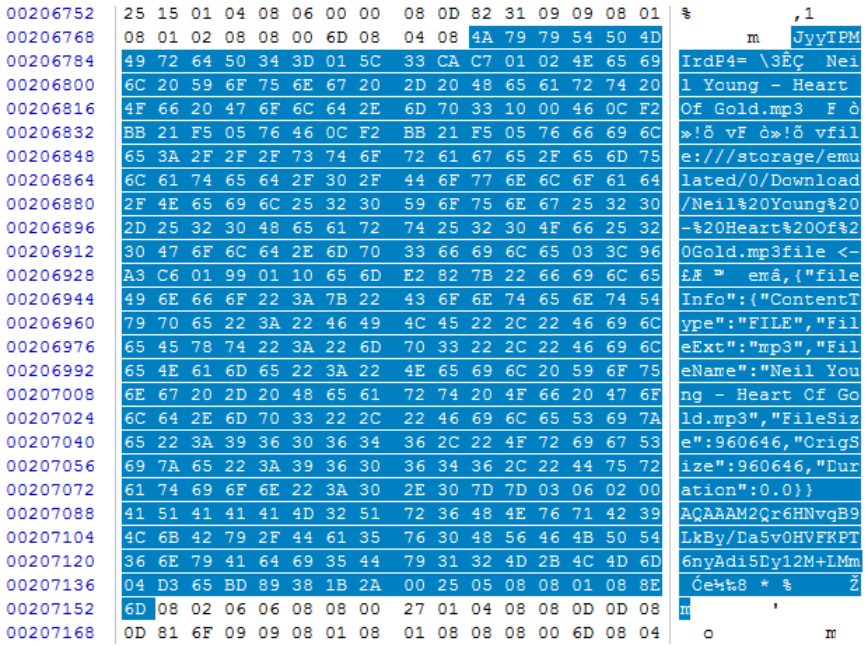
**Viber text messages** When I have analyzed viber_messages database in Winhex, I have seen that all the messages that are marked deleted still lies in database as shown in Figure 4.36



```
00204160  00 00 08 0D 11 09 09 08  01 08 01 02 08 08 17 00
00204176  08 04 08 4A 79 79 54 50  4D 49 72 64 50 34 3D 01     JyyTPMIrdP4=
00204192  5C 33 C1 17 29 02 48 65  79 10 00 46 0C F0 46 B3  \3Á ) Hey  F ðF³
00204208  B5 05 F3 46 0C F0 46 B3  B5 05 F3 74 65 78 74 03  µ óF ðF³µ ótext
00204224  3C 6E F3 AE 7B 7D 03 06  02 00 6E 6F 5F 73 70 04  <nó®{}     no_sp
00204240  C3 B2 F6 67 07 28 00 25  05 08 08 01 08 11 08 02  Ã²ög ( %
00204256  06 06 08 08 00 15 01 04  08 08 00 00 08 0D 11 09
00204272  09 08 01 08 01 02 08 08  17 00 08 04 08 4A 79 79               Jyy
00204288  54 50 4D 49 72 64 50 34  3D 01 5C 33 C0 51 F4 02  TPMIrdP4= \3ÀQô
00204304  48 69 10 02 46 0C F0 15  6D 31 57 3B 46 0C F0 15  Hi  F ð m1W;F ð
00204320  6D 31 57 3B 74 65 78 74  03 3C 6B AA 91 7B 7D 03  m1W;text <kª'{}
00204336  07 04 00 6E 6F 5F 73 70  04 C3 A0 B5 75 06 28 00     no_sp Ã µu (
00204352  25 05 08 08 01 08 2D 08  02 06 06 08 08 00 15 01  %     -
00204368  04 08 08 00 00 08 0D 11  09 09 08 01 08 01 02 08
00204384  08 17 00 08 04 08 4A 79  79 54 50 4D 49 72 64 50        JyyTPMIrdP
00204400  34 3D 01 5C 33 C0 74 0D  02 57 65 6C 63 6F 6D 65  4= \3Àt  Welcome
00204416  20 74 6F 20 76 69 62 65  72 10 02 46 0C F0 1E 0C   to viber  F ð
00204432  31 54 EE 46 0C F0 1E 0C  31 54 EE 74 65 78 74 03  1TîF ð  1Tîtext
00204448  3C 6C 4A 93 7B 7D 03 07  04 00 6E 6F 5F 73 70 04  <lJ"{}     no_sp
00204464  C3 A0 B5 6E 04 28 00 25  05 08 08 01 09 27 08 08  Ã µn ( %      '
00204480  06 06 08 08 00 15 01 08  08 08 00 2B 08 0D 11 08             +
00204496  09 08 09 08 09 08 08 08  00 00 08 08 08 61 56 46               aVF
00204512  65 69 38 41 67 30 54 4D  3D 01 5B 8C 9C AF FD 02  ei8Ag0TM= [Œœ¯ý
00204528  6F 75 74 67 6F 69 6E 67  5F 63 61 6C 6C 45 E3 27  outgoing_callEã'
00204544  25 C0 75 36 5A 45 E3 27  25 C0 75 36 5A 63 61 6C  %Àu6ZEã'%Àu6Zcal
00204560  6C 03 68 61 73 5F 64 65  73 63 72 69 70 74 69 6F  l has_descriptio
00204576  6E 7B 7D 5D 03 28 00 0D  05 08 08 09 09 13 08 08  n{}] (
00204592  06 06 08 08 00 15 01 04  08 08 00 00 08 0D 11 08
```

**Figure 4.36:** Viber deleted text message

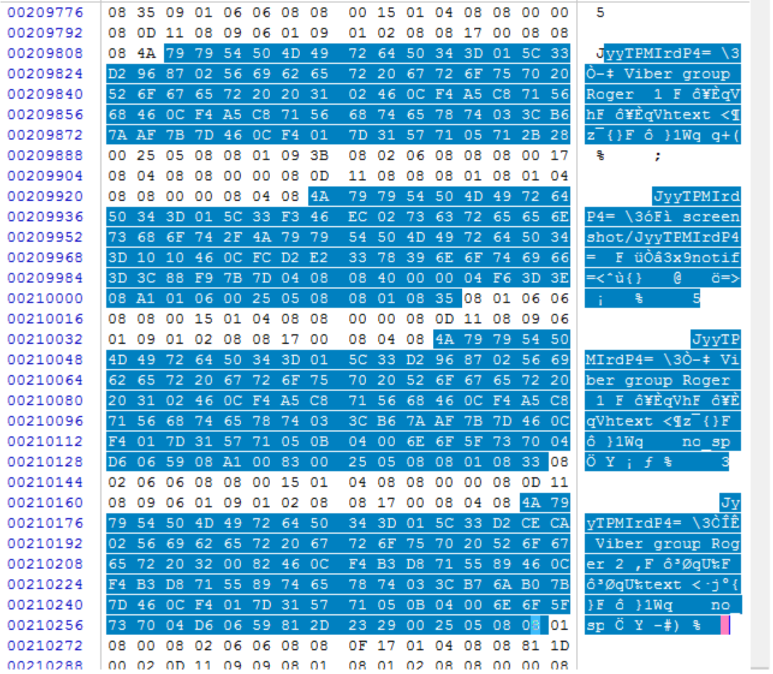
**Viber deleted media messages** Viber media message that was deleted is carved from memory and is shown in Figure 4.37.



```
00206752  25 15 01 04 08 06 00 00  08 0D 82 31 09 09 08 01  %          ,1
00206768  08 01 02 08 08 00 6D 08  04 08 4A 79 79 54 50 4D        m   JyyTPM
00206784  49 72 64 50 34 3D 01 5C  33 CA C7 01 02 4E 65 69  IrdP4= \3ÊÇ  Nei
00206800  6C 20 59 6F 75 6E 67 20  2D 20 48 65 61 72 74 20  l Young - Heart 
00206816  4F 66 20 47 6F 6C 64 2E  6D 70 33 10 00 46 0C F2  Of Gold.mp3  F ò
00206832  BB 21 F5 05 76 46 0C F2  BB 21 F5 05 76 66 69 6C  »!õ vF ò»!õ vfil
00206848  65 3A 2F 2F 2F 73 74 6F  72 61 67 65 2F 65 6D 75  e:///storage/emu
00206864  6C 61 74 65 64 2F 30 2F  44 6F 77 6E 6C 6F 61 64  lated/0/Download
00206880  2F 4E 65 69 6C 25 32 30  59 6F 75 6E 67 25 32 30  /Neil%20Young%20
00206896  2D 25 32 30 48 65 61 72  74 25 32 30 4F 66 25 32  -%20Heart%20Of%2
00206912  30 47 6F 6C 64 2E 6D 70  33 66 69 6C 65 03 3C 96  0Gold.mp3file <–
00206928  A3 C6 01 99 01 10 65 6D  E2 82 7B 22 66 69 6C 65  £Æ ™  emâ‚{"file
00206944  49 6E 66 6F 22 3A 7B 22  43 6F 6E 74 65 6E 74 54  Info":{"ContentT
00206960  79 70 65 22 3A 22 46 49  4C 45 22 2C 22 46 69 6C  ype":"FILE","Fil
00206976  65 45 78 74 22 3A 22 6D  70 33 22 2C 22 46 69 6C  eExt":"mp3","Fil
00206992  65 4E 61 6D 65 22 3A 22  4E 65 69 6C 20 59 6F 75  eName":"Neil You
00207008  6E 67 20 2D 20 48 65 61  72 74 20 4F 66 20 47 6F  ng - Heart Of Go
00207024  6C 64 2E 6D 70 33 22 2C  22 46 69 6C 65 53 69 7A  ld.mp3","FileSiz
00207040  65 22 3A 39 36 30 36 34  36 2C 22 4F 72 69 67 53  e":960646,"OrigS
00207056  69 7A 65 22 3A 39 36 30  36 34 36 2C 22 44 75 72  ize":960646,"Dur
00207072  61 74 69 6F 6E 22 3A 30  2E 30 7D 7D 03 06 02 00  ation":0.0}}
00207088  41 51 41 41 41 4D 32 51  72 36 48 4E 76 71 42 39  AQAAAM2Qr6HNvqB9
00207104  4C 6B 42 79 2F 44 61 35  76 30 48 56 46 4B 50 54  LkBy/Da5v0HVFKPT
00207120  36 6E 79 41 64 69 35 44  79 31 32 4D 2B 4C 4D 6D  6nyAdi5Dy12M+LMm
00207136  04 D3 65 BD 89 38 1B 2A  00 25 05 08 08 01 08 8E   Óe½‰8 * %     Ž
00207152  6D 08 02 06 06 08 08 00  27 01 04 08 08 0D 0D 08  m       '
00207168  0D 81 6F 09 09 08 01 08  01 08 08 08 00 6D 08 04    o          m
```

**Figure 4.37:** Viber deleted media message

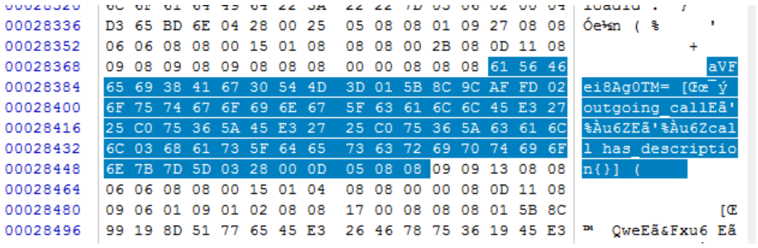
**Viber deleted group messages** Viber deleted group messages that are carved manually are shown in Figure 4.38.



```
00209776  08 35 09 01 06 06 08 08  00 15 01 04 08 08 00 00   5
00209792  08 0D 11 08 09 06 01 09  01 02 08 08 17 00 08 08
00209808  08 4A 79 79 54 50 4D 49  72 64 50 34 3D 01 5C 33   JyyTPMIrdP4= \3
00209824  D2 96 87 02 56 69 62 65  72 20 67 72 6F 75 70 20  Ò–‡ Viber group 
00209840  52 6F 67 65 72 20 20 31  02 46 0C F4 A5 C8 71 56  Roger  1 F ô¥ÈqV
00209856  68 46 0C F4 A5 C8 71 56  68 74 65 78 74 03 3C B6  hF ô¥ÈqVhtext <¶
00209872  7A AF 7B 7D 46 0C F4 01  7D 31 57 71 05 71 2B 28  z¯{}F ô }1Wq q+(
00209888  00 25 05 08 08 01 09 3B  08 02 06 08 08 08 00 17   %     ;
00209904  08 04 08 08 00 00 08 0D  11 08 08 08 01 08 01 04
00209920  08 08 00 00 08 04 08 4A  79 79 54 50 4D 49 72 64         JyyTPMIrd
00209936  50 34 3D 01 5C 33 F3 46  EC 02 73 63 72 65 65 6E  P4= \3óFì screen
00209952  73 68 6F 74 2F 4A 79 79  54 50 4D 49 72 64 50 34  shot/JyyTPMIrdP4
00209968  3D 10 10 46 0C FC D2 E2  33 78 39 6E 6F 74 69 66  =  F üÒâ3x9notif
00209984  3D 3C 88 F9 7B 7D 04 08  08 40 00 00 04 F6 3D 3E  =<ˆù{}   @   ö=>
00210000  08 A1 01 06 00 25 05 08  08 01 08 35 08 01 06 06   ¡   %     5
00210016  08 08 00 15 01 04 08 08  00 00 08 0D 11 08 09 06
00210032  01 09 01 02 08 08 17 00  08 04 08 4A 79 79 54 50             JyyTP
00210048  4D 49 72 64 50 34 3D 01  5C 33 D2 96 87 02 56 69  MIrdP4= \3Ò–‡ Vi
00210064  62 65 72 20 67 72 6F 75  70 20 52 6F 67 65 72 20  ber group Roger 
00210080  20 31 02 46 0C F4 A5 C8  71 56 68 46 0C F4 A5 C8   1 F ô¥ÈqVhF ô¥È
00210096  71 56 68 74 65 78 74 03  3C B6 7A AF 7B 7D 46 0C  qVhtext <¶z¯{}F 
00210112  F4 01 7D 31 57 71 05 0B  04 00 6E 6F 5F 73 70 04  ô }1Wq    no_sp
00210128  D6 06 59 08 A1 00 83 00  25 05 08 08 01 08 33 08  Ö Y ¡ ƒ %     3
00210144  02 06 06 08 08 00 15 01  04 08 08 00 00 08 0D 11
00210160  08 09 06 01 09 01 02 08  08 17 00 08 04 08 4A 79                Jy
00210176  79 54 50 4D 49 72 64 50  34 3D 01 5C 33 D2 CE CA  yTPMIrdP4= \3ÒÎÊ
00210192  02 56 69 62 65 72 20 67  72 6F 75 70 20 52 6F 67   Viber group Rog
00210208  65 72 20 32 00 82 46 0C  F4 B3 D8 71 55 89 46 0C  er 2 ‚F ô³ØqU‰F 
00210224  F4 B3 D8 71 55 89 74 65  78 74 03 3C B7 6A B0 7B  ô³ØqU‰text <·j°{
00210240  7D 46 0C F4 01 7D 31 57  71 05 0B 04 00 6E 6F 5F  }F ô }1Wq    no_
00210256  73 70 04 D6 06 59 81 2D  23 29 00 25 05 08 08 01  sp Ö Y -#) %
00210272  08 00 08 02 06 06 08 08  0F 17 01 04 08 08 81 1D
00210288  00 02 0D 11 09 09 08 01  08 01 02 08 08 00 00 08
```

**Figure 4.38:** Viber deleted group message

**Viber deleted call** Deleted calls records can also be carved from memory as shown in Figure 4.39.



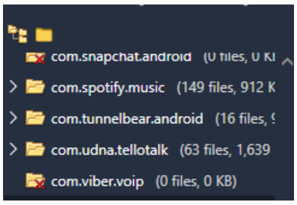
```
00028336  D3 65 BD 6E 04 28 00 25  05 08 08 01 09 27 08 08  Óe½n ( %       '
00028352  06 06 08 08 00 15 01 08  08 08 00 2B 08 0D 11 08          +
00028368  09 08 09 08 09 08 08 08  00 00 08 08 08 61 56 46               aVF
00028384  65 69 38 41 67 30 54 4D  3D 01 5B 8C 9C AF FD 02  ei8Ag0TM= [Œœ¯ý
00028400  6F 75 74 67 6F 69 6E 67  5F 63 61 6C 6C 45 E3 27  outgoing_callEã'
00028416  25 C0 75 36 5A 45 E3 27  25 C0 75 36 5A 63 61 6C  %Àu6ZEã'%Àu6Zcal
00028432  6C 03 68 61 73 5F 64 65  73 63 72 69 70 74 69 6F  l has_descriptio
00028448  6E 7B 7D 5D 03 28 00 0D  05 08 08 09 09 13 08 08  n{}] (
00028464  06 06 08 08 00 15 01 04  08 08 00 00 08 0D 11 08
00028480  09 06 01 09 01 02 08 08  17 00 08 08 08 01 5B 8C                [Œ
00028496  99 19 8D 51 77 65 45 E3  26 46 78 75 36 19 45 E3  ™  QweEã&Fxu6 Eã
```

**Figure 4.39:** Viber deleted call

## 4.3.4 Analysis of Viber Chat & media shared between participants on device - Application Deleted

Once the Viber application is deleted, no data can be carved. After deleting the application, Viber over-writes all its bytes with zero, hence investigator can not retrieve any data.

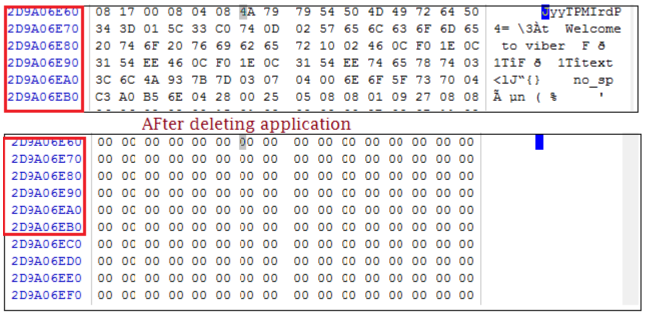
As shown in Figure 4.40, viber application was once installed but it is now deleted and hence marked with red cross. Next thing to note in the figure is that zero files are retrieved.



com.snapchat.android (0 files, 0 KI
com.spotify.music (149 files, 912 K
com.tunnelbear.android (16 files, !
com.udna.tellotalk (63 files, 1,639
com.viber.voip (0 files, 0 KB)

**Figure 4.40:** Viber deleted application marked with red cross

In Figure 4.41, we can see that viber messages are now overwritten with zeros.



```
2D9A06E60  08 17 00 08 04 08 4A 79  79 54 50 4D 49 72 64 50        JyyTPMIrdP
2D9A06E70  34 3D 01 5C 33 C0 74 0D  02 57 65 6C 63 6F 6D 65  4= \3Àt  Welcome
2D9A06E80  20 74 6F 20 76 69 62 65  72 10 02 46 0C F0 1E 0C   to viber  F ð
2D9A06E90  31 54 EE 46 0C F0 1E 0C  31 54 EE 74 65 78 74 03  1TîF ð  1Tîtext
2D9A06EA0  3C 6C 4A 93 7B 7D 03 07  04 00 6E 6F 5F 73 70 04  <lJ"{}    no_sp
2D9A06EB0  C3 A0 B5 6E 04 28 00 25  05 08 08 01 09 27 08 08  Ã µn ( %      '
```

AFter deleting application

```
2D9A06E60  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06E70  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06E80  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06E90  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06EA0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06EB0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06EC0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06ED0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06EE0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
2D9A06EF0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

**Figure 4.41:** Viber messages byte after deleting application

## 4.4 Forensic Investigation of TelloTalk

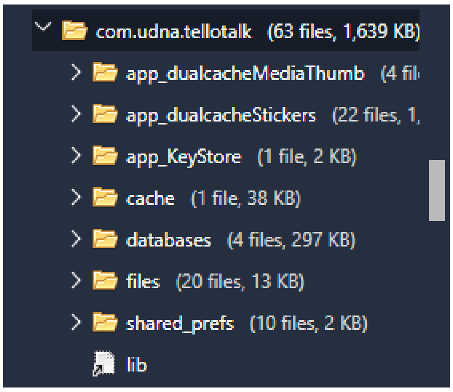
Artifacts that a investigator can gather from TelloTalk application during investigation are discussed in this section.

### 4.4.1 TelloTalk Anatomy

Like other social messaging applications, TelloTalk keeps application and user related data in internal memory of Android device, under a folder named *com.udna.tellotalk*. This folder is inaccessible to user unless Android device is rooted. Path to this folder is /data/data/com.udna.tellotalk and its directory structure is shown in Figure 4.42.

TelloTalk can save media files on internal or external memory depending on user settings. If TelloTalk stores media files on device internal memory, path to its media folder is /data/media/0/TelloTalk.

Both these folder should carefully be examined as they hold important information that might help during investigation.



```
com.udna.tellotalk  (63 files, 1,639 KB)
    app_dualcacheMediaThumb  (4 fil
    app_dualcacheStickers  (22 files, 1,
    app_KeyStore  (1 file, 2 KB)
    cache  (1 file, 38 KB)
    databases  (4 files, 297 KB)
    files  (20 files, 13 KB)
    shared_prefs  (10 files, 2 KB)
    lib
```

**Figure 4.42:** TelloTalk directory structure

#### 4.4.1.1 TelloTalk Databases

TelloTalk has two databases that it uses to store user and application related data. These database files are found in /data/data/com.udna.tellotalk/databases directory. TelloTalk does not encrypt database files like Viber and investigator can easily retrieve all data from these files. These two database files are:

- history
- awss3transfertable

Both these files are discussed in next section.

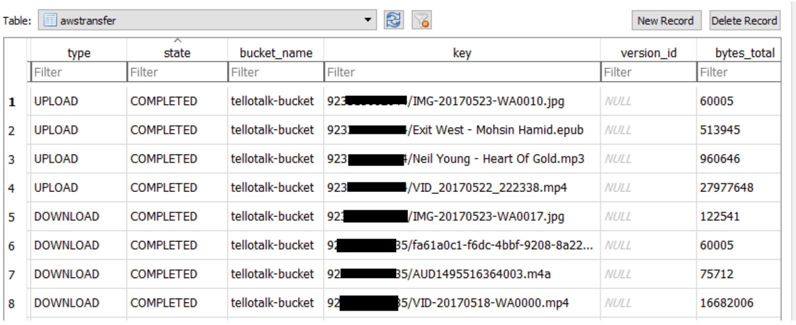
**4.4.1.1.1 history** This is the most important database file as it is used to store all application and user related data including account, contacts, conversations, and messages data. Important tables in this schema are discussed below:

1. accounts - This table stores client's account information. It stores user's unique id, phone number, display name, keys, and port.
2. contacts - It stores details of all TelloTalk contacts including user's account unique id, name, phone number, last seen timestamp and groups id.
3. conversation - Details of TelloTalk one-to-one and group conversation are stored in this table. Following are details of this table:
    - uuid - unique id assigned to each conversation
    - name - name of conversation
    - accountUuid - user own unique id
    - contactJid - TelloTalk unique contact id of other member of conversation. It is usually phone number concatenated with @tellotalk.
    - created - timestamp of when conversation was created/first message was sent.
    - hidden - flag; is this chat hidden?
4. messages - this table stores all messages shared on group or on one-to-one chat. Details of this table is:
    - uuid - unique id of each message
    - conversationUuid - unique id of conversation; to link each message with its conversation
    - timeSent - Epoch time-stamp of when message was sent

- counterpart - phone number + @tellotalk of message writer.
- body - actual message;
    - if it is text message, value in this field will be actual message
    - if it is media message, value in this field will be phone/name of Media File or user unique id | size of file e.g. 923315082044/Exit West - Mohsin Hamid.epub|513945
    - if it is location message, value will be latitude and longitude of location e.g. geo:33.5780967,73.0621473
- type - message type
    - its value is 0, if it is text or location message
    - its value is 1, if it is image message
    - its value is 2, if message type is audio, video, sound and document.
- relativeFilePath - if message type is media and it is sent to other party, this field holds the complete path of media file lies on user device.
- read - flag set to 1 if message is read otherwise 0.
- edited - if orignal message is edited, this field will store the edited text
- deleted - if user deletes a message, value of this field is 1. Otherwise default value is 0.
- fileTag - Special tag assigned to media or location message.
    - if message is of location type, this field has location value in text. E.g Khadim Hussain Road, Rawalpindi, Pakistan
    - if it is media message then value can be Image message, Audio message, Video message.
- remoteMsgId - This value corresponds to unique id of message sent by other participant.

5. non_tello_contacts - This table stores all phone book contacts that are not registered on TelloTalk. This information includes non-TelloTalk user name and phone number.
6. prekeys - When TelloTalk is first installed on system, this table is populated with 100 public keys that is used to encrypt messages. These keys changes when required.
7. signed_prekeys - It stores public key that is first generated at the time of installation but changes periodic also.

**4.4.1.1.2 awss3transfertable** TelloTalk has created a separate database schema to store data of all media transactions. This database has table named *awstranfer*



| | type | state | bucket_name | key | version_id | bytes_total |
|---|---|---|---|---|---|---|
| 1 | UPLOAD | COMPLETED | tellotalk-bucket | 923[redacted]/IMG-20170523-WA0010.jpg | NULL | 60005 |
| 2 | UPLOAD | COMPLETED | tellotalk-bucket | 923[redacted]/Exit West - Mohsin Hamid.epub | NULL | 513945 |
| 3 | UPLOAD | COMPLETED | tellotalk-bucket | 923[redacted]/Neil Young - Heart Of Gold.mp3 | NULL | 960646 |
| 4 | UPLOAD | COMPLETED | tellotalk-bucket | 923[redacted]/VID_20170522_222338.mp4 | NULL | 27977648 |
| 5 | DOWNLOAD | COMPLETED | tellotalk-bucket | 92[redacted]/IMG-20170523-WA0017.jpg | NULL | 122541 |
| 6 | DOWNLOAD | COMPLETED | tellotalk-bucket | 9[redacted]5/fa61a0c1-f6dc-4bbf-9208-8a22... | NULL | 60005 |
| 7 | DOWNLOAD | COMPLETED | tellotalk-bucket | 92[redacted]5/AUD1495516364003.m4a | NULL | 75712 |
| 8 | DOWNLOAD | COMPLETED | tellotalk-bucket | 92[redacted]5/VID-20170518-WA0000.mp4 | NULL | 16682006 |

**Figure 4.43:** TelloTalk - content of table ’awstranfer’

that keeps data related to media that was uploaded or downloaded as shown in Figure 4.43. TelloTalk transfers 5242880 bytes in one message so if file is greater than specified number of bytes, it divides the files in chunk of 5242880. Information related to these chunks are also available in this table. Other details of this table are:

1. _id - unique id assigned to each transfer.
2. main_upload_id - If file size is greater than 5242880 bytes, it is divided into chunks. Value in this field corresponds to the _id of main transfer.
3. type - type of transfer; Upload or download
4. state - state of each transfer; either Completed or Part_Completed
5. key - name of file; corresponds to *body* field of message table in history database
6. bytes_total - size of file in bytes
7. is_encrypted - encryption flag; 1 if file is encrypted otherwise 0
8. file - relative path/location where file is stored
9. is_multipart - multi-part flag; 1 if file is transferred in chunks otherwise 0

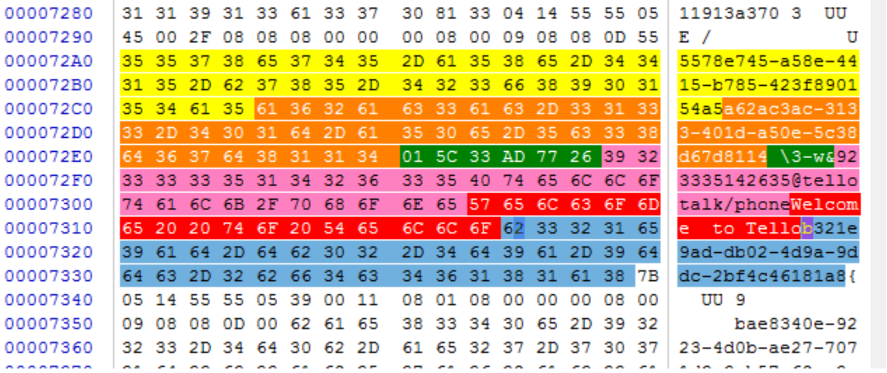
#### 4.4.1.2 Cryptography elements in TelloTalk

**Database Encryption** Unfortunately, TelloTalk does not encrypt any of its database files and investigator can extract all application and user related data from these files easily.

**Message Encryption** TelloTalk sends all messages over network in plain text. Messages are not encrypted.

#### 4.4.1.3 TelloTalk Text Message Anatomy

Format of TelloTalk text messages is shown in Figure 4.44.



```
00007280  31 31 39 31 33 61 33 37  30 81 33 04 14 55 55 05  11913a370 3  UU
00007290  45 00 2F 08 08 08 00 00  00 08 00 09 08 08 0D 55  E /            U
000072A0  35 35 37 38 65 37 34 35  2D 61 35 38 65 2D 34 34  5578e745-a58e-44
000072B0  31 35 2D 62 37 38 35 2D  34 32 33 66 38 39 30 31  15-b785-423f8901
000072C0  35 34 61 35 61 36 32 61  63 33 61 63 2D 33 31 33  54a5a62ac3ac-313
000072D0  33 2D 34 30 31 64 2D 61  35 30 65 2D 35 63 33 38  3-401d-a50e-5c38
000072E0  64 36 37 64 38 31 31 34  01 5C 33 AD 77 26 39 32  d67d8114 \3-w&92
000072F0  33 33 33 35 31 34 32 36  33 35 40 74 65 6C 6C 6F  3335142635@tello
00007300  74 61 6C 6B 2F 70 68 6F  6E 65 57 65 6C 63 6F 6D  talk/phoneWelcom
00007310  65 20 20 74 6F 20 54 65  6C 6C 6F 62 33 32 31 65  e  to Tellob321e
00007320  39 61 64 2D 64 62 30 32  2D 34 64 39 61 2D 39 64  9ad-db02-4d9a-9d
00007330  64 63 2D 32 62 66 34 63  34 36 31 38 31 61 38 7B  dc-2bf4c46181a8{
00007340  05 14 55 55 05 39 00 11  08 01 08 00 00 00 08 00    UU 9
00007350  09 08 08 0D 00 62 61 65  38 33 34 30 65 2D 39 32       bae8340e-92
00007360  32 33 2D 34 64 30 62 2D  61 65 32 37 2D 37 30 37  23-4d0b-ae27-707
```

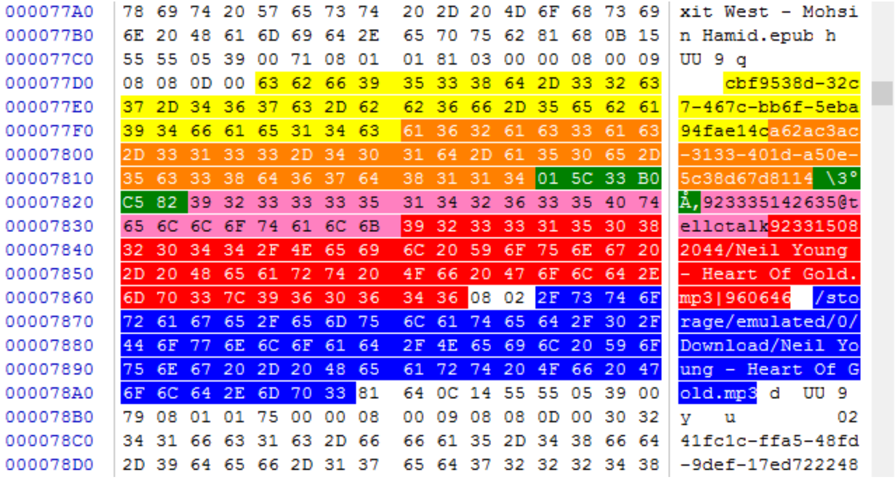
**Figure 4.44:** TelloTalk text message anatomy

In Figure 4.44,

1. 24 bytes unique id assigned to each message is highlighted in yellow.
2. 24 bytes conversation id is highlighted in orange.
3. 6 bytes Epoch timestamp is highlighed in green.
4. Counterpart is highlighted in pink.
5. Message body is highlighted in red.
6. 24 bytes remote message id is highlighted in blue.

#### 4.4.1.4 TelloTalk Media Message Anatomy

Format of TelloTalk media messages is shown in Figure 4.45.



```
000077A0  78 69 74 20 57 65 73 74  20 2D 20 4D 6F 68 73 69  xit West - Mohsi
000077B0  6E 20 48 61 6D 69 64 2E  65 70 75 62 81 68 0B 15  n Hamid.epub h
000077C0  55 55 05 39 00 71 08 01  01 81 03 00 00 08 00 09  UU 9 q
000077D0  08 08 0D 00 63 62 66 39  35 33 38 64 2D 33 32 63      cbf9538d-32c
000077E0  37 2D 34 36 37 63 2D 62  62 36 66 2D 35 65 62 61  7-467c-bb6f-5eba
000077F0  39 34 66 61 65 31 34 63  61 36 32 61 63 33 61 63  94fae14ca62ac3ac
00007800  2D 33 31 33 33 2D 34 30  31 64 2D 61 35 30 65 2D  -3133-401d-a50e-
00007810  35 63 33 38 64 36 37 64  38 31 31 34 01 5C 33 B0  5c38d67d8114 \3°
00007820  C5 82 39 32 33 33 33 35  31 34 32 36 33 35 40 74  Å,923335142635@t
00007830  65 6C 6C 6F 74 61 6C 6B  39 32 33 33 31 35 30 38  ellotalk92331508
00007840  32 30 34 34 2F 4E 65 69  6C 20 59 6F 75 6E 67 20  2044/Neil Young
00007850  2D 20 48 65 61 72 74 20  4F 66 20 47 6F 6C 64 2E  - Heart Of Gold.
00007860  6D 70 33 7C 39 36 30 36  34 36 08 02 2F 73 74 6F  mp3|960646  /sto
00007870  72 61 67 65 2F 65 6D 75  6C 61 74 65 64 2F 30 2F  rage/emulated/0/
00007880  44 6F 77 6E 6C 6F 61 64  2F 4E 65 69 6C 20 59 6F  Download/Neil Yo
00007890  75 6E 67 20 2D 20 48 65  61 72 74 20 4F 66 20 47  ung - Heart Of G
000078A0  6F 6C 64 2E 6D 70 33 81  64 0C 14 55 55 05 39 00  old.mp3 d  UU 9
000078B0  79 08 01 01 75 00 00 08  00 09 08 08 0D 00 30 32  y   u          02
000078C0  34 31 66 63 31 63 2D 66  66 61 35 2D 34 38 66 64  41fc1c-ffa5-48fd
000078D0  2D 39 64 65 66 2D 31 37  65 64 37 32 32 32 34 38  -9def-17ed722248
```

**Figure 4.45:** TelloTalk text message anatomy

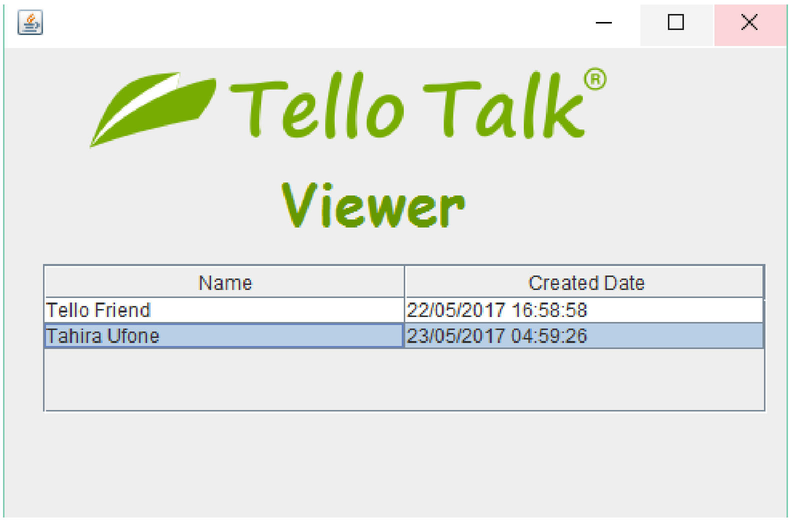
In Figure 4.45,

1. 24 bytes unique id assigned to each message is highlighted in yellow.
2. 24 bytes conversation id is highlighted in orange.
3. 6 bytes Epoch timestamp is highlighed in green.
4. Counterpart is highlighted in pink.
5. Message body is highlighted in red.
6. Relative path of file is highlighted in blue.

### 4.4.2 Analysis of TelloTalk Chat & media shared between participants on device - Data not Deleted

TelloTalk stores its conversations and messages information in database. Using Cellebrite UFED Physical Analyzer or other tools available in market, we can read database file and extract messages in each conversation manually. But there is no application in market that can be used to view TelloTalk chats. So I created a basic Java application like WhatsApp Viewer and named it **TelloTalk Viewer**. This application will display all conversations and messages in each conversation.

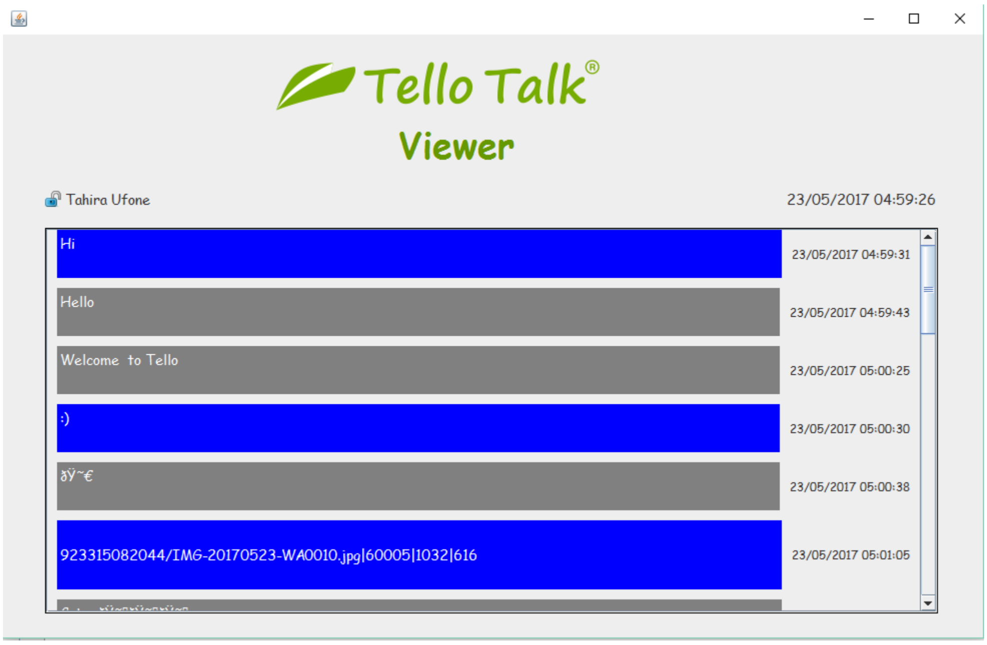
**Conversations** TelloTalk Viewer displays all TelloTalk conversations along with its name and created date as shown in Figure 4.46.



| Name | Created Date |
|---|---|
| Tello Friend | 22/05/2017 16:58:58 |
| Tahira Ufone | 23/05/2017 04:59:26 |

**Figure 4.46:** TelloTalk conversation - TelloTalk Viewer

**Text Messages** Text messages can be viewed in TelloTalk Viewer as shown in Figure 4.47.



Tahira Ufone 23/05/2017 04:59:26

Hi 23/05/2017 04:59:31

Hello 23/05/2017 04:59:43

Welcome to Tello 23/05/2017 05:00:25

:) 23/05/2017 05:00:30

ðŸ˜€ 23/05/2017 05:00:38

923315082044/IMG-20170523-WA0010.jpg|60005|1032|616 23/05/2017 05:01:05

**Figure 4.47:** TelloTalk text messages - TelloTalk Viewer

**Media Messages** Media messages that have image, audio, video, sound, and document can be extracted but manually. In messages table of history database, there is field named *relativeFilePath*, value in this field holds the exact path of media file and can be used retrieve it.

**TelloTalk Contacts** TelloTalk contacts can be viewed in DB browser for SQLite as shown in Figure 4.48.

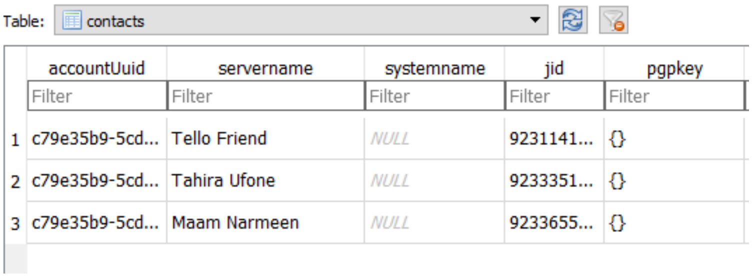


Table: contacts

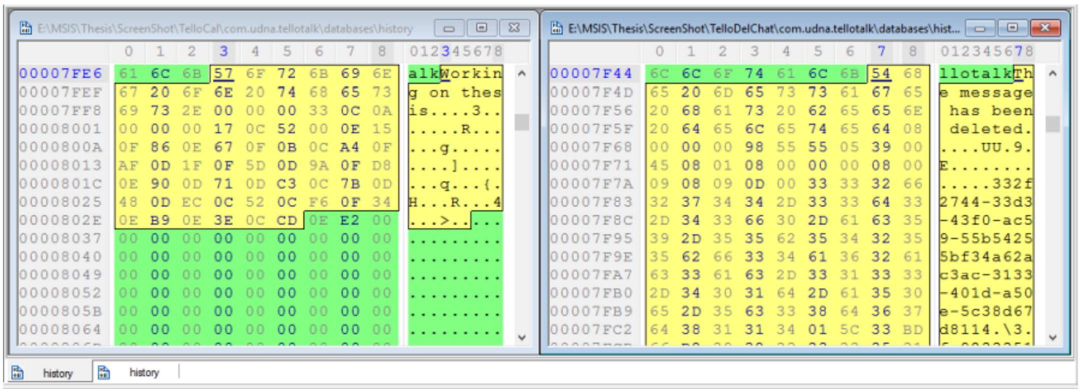
| | accountUuid | servername | systemname | jid | pgpkey |
|---|---|---|---|---|---|
| | Filter | Filter | Filter | Filter | Filter |
| 1 | c79e35b9-5cd... | Tello Friend | NULL | 9231141... | {} |
| 2 | c79e35b9-5cd... | Tahira Ufone | NULL | 9233351... | {} |
| 3 | c79e35b9-5cd... | Maam Narmeen | NULL | 9233655... | {} |

**Figure 4.48:** TelloTalk contacts

### 4.4.3 Analysis of TelloTalk Chat & media shared between participants on Android device - Data Deleted

In this subsection, text and media messages that can be recovered after deleting of single chat or complete conversation will be discussed.

**Recovering Deleted Message in Conversation** When we remove a text message from a conversation, TelloTalk over write its body content with 'This message has been deleted'. So once a message within a conversation is deleted, we can not retrieved it. As shown in Figure 4.49, actual message 'Working on thesis' (left window) is replaced with 'This message has been deleted' (right window). Still we can get some of data related to deleted messages like sent timestamp and contact information of other participant.



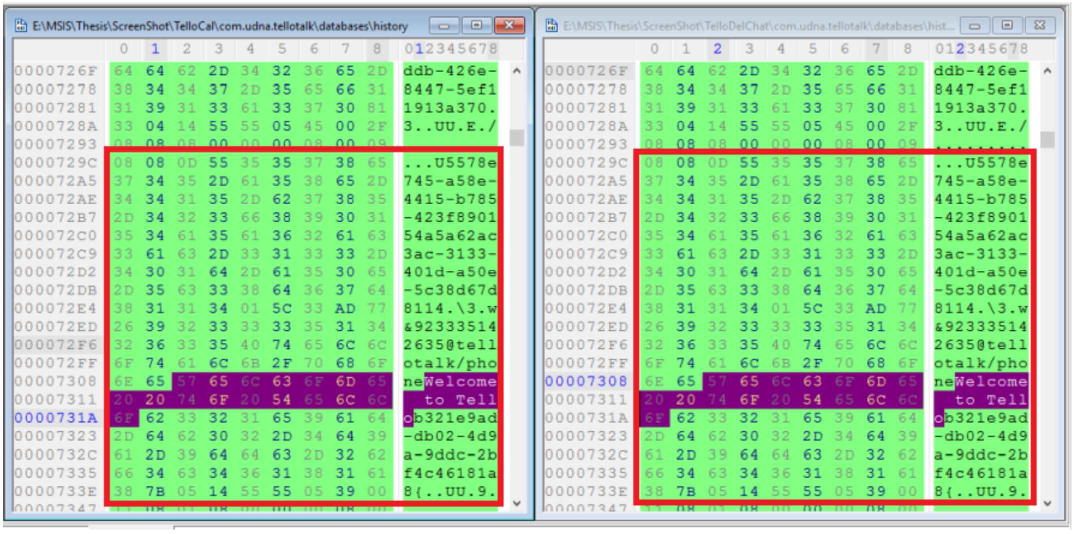
```
E:\MSIS\Thesis\ScreenShot\TelloCal\com.udna.tellotalk\databases\history
          0  1  2  3  4  5  6  7  8   012345678
00007FE6  61 6C 6B 57 6F 72 6B 69 6E  alkWorkin
00007FEF  67 20 6F 6E 20 74 68 65 73  g on thes
00007FF8  69 73 2E 00 00 00 33 0C 0A  is....3..
00008001  00 00 00 17 0C 52 00 0E 15  ......R...
0000800A  0F 86 0E 67 0F 0B 0C A4 0F  ...g.....
00008013  AF 0D 1F 0F 5D 0D 9A 0F D8  ....]....
0000801C  0E 90 0D 71 0D C3 0C 7B 0D  ...q...{.
00008025  48 0D EC 0C 52 0C F6 0F 34  H...R...4
0000802E  0E B9 0E 3E 0C CD 0E E2 00  ...>.....
00008037  00 00 00 00 00 00 00 00 00  .........
00008040  00 00 00 00 00 00 00 00 00  .........
00008049  00 00 00 00 00 00 00 00 00  .........
00008052  00 00 00 00 00 00 00 00 00  .........
0000805B  00 00 00 00 00 00 00 00 00  .........
00008064  00 00 00 00 00 00 00 00 00  .........
```

```
E:\MSIS\Thesis\ScreenShot\TelloDelChat\com.udna.tellotalk\databases\hist...
          0  1  2  3  4  5  6  7  8   012345678
00007F44  6C 6C 6F 74 61 6C 6B 54 68  llotalkTh
00007F4D  65 20 6D 65 73 73 61 67 65  e message
00007F56  20 68 61 73 20 62 65 65 6E   has been
00007F5F  20 64 65 6C 65 74 65 64 08   deleted.
00007F68  00 00 00 98 55 55 05 39 00  ....UU.9.
00007F71  45 08 01 08 00 00 00 08 00  E........
00007F7A  09 08 09 0D 00 33 33 32 66  .....332f
00007F83  32 37 34 34 2D 33 33 64 33  2744-33d3
00007F8C  2D 34 33 66 30 2D 61 63 35  -43f0-ac5
00007F95  39 2D 35 35 62 35 34 32 35  9-55b5425
00007F9E  35 62 66 33 34 61 36 32 61  5bf34a62a
00007FA7  63 33 61 63 2D 33 31 33 33  c3ac-3133
00007FB0  2D 34 30 31 64 2D 61 35 30  -401d-a50
00007FB9  65 2D 35 63 33 38 64 36 37  e-5c38d67
00007FC2  64 38 31 31 34 01 5C 33 BD  d8114.\3.
```

**Figure 4.49:** TelloTalk deleted message

**Recovering Messages in Deleted Conversation** If user deletes a complete conversation, all messages with in this conversation are marked deleted but still remain in the memory and investigator can retrieve these messages. Number of messages an investigator can retrieve depends on user activities as this unallocated space will be used to save future messages.

In this research, I have been able to recover most of deleted messages after user deletes a conversation. Database files are recovered at two points, one before and other after deletion of conversation. As shown in Figure 4.50, comparison of these two database files are done and we have reach the conclusion that it is still possible to recover a text message after deleting complete conversation.



E:\MSIS\Thesis\ScreenShot\TelloCal\com.udna.tellotalk\databases\history

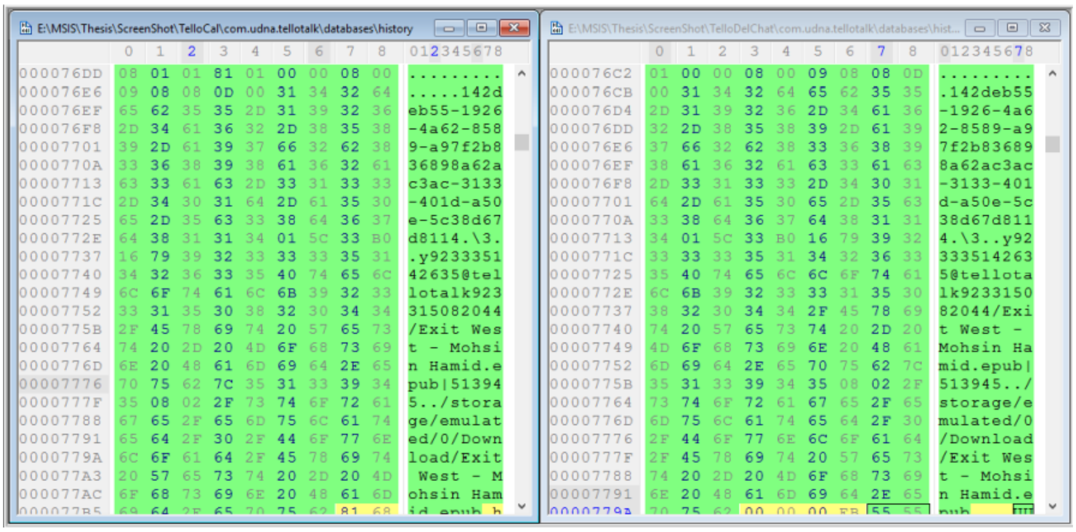
```
          0  1  2  3  4  5  6  7  8   012345678
0000726F  64 64 62 2D 34 32 36 65 2D  ddb-426e-
00007278  38 34 34 37 2D 35 65 66 31  8447-5ef1
00007281  31 39 31 33 61 33 37 30 81  1913a370.
0000728A  33 04 14 55 55 05 45 00 2F  3..UU.E./
00007293  08 08 08 00 00 00 08 00 09  .........
0000729C  08 08 0D 55 35 35 37 38 65  ...U5578e
000072A5  37 34 35 2D 61 35 38 65 2D  745-a58e-
000072AE  34 34 31 35 2D 62 37 38 35  4415-b785
000072B7  2D 34 32 33 66 38 39 30 31  -423f8901
000072C0  35 34 61 35 61 36 32 61 63  54a5a62ac
000072C9  33 61 63 2D 33 31 33 33 2D  3ac-3133-
000072D2  34 30 31 64 2D 61 35 30 65  401d-a50e
000072DB  2D 35 63 33 38 64 36 37 64  -5c38d67d
000072E4  38 31 31 34 01 5C 33 AD 77  8114.\3.w
000072ED  26 39 32 33 33 33 35 31 34  &92333514
000072F6  32 36 33 35 40 74 65 6C 6C  2635@tell
000072FF  6F 74 61 6C 6B 2F 70 68 6F  otalk/pho
00007308  6E 65 57 65 6C 63 6F 6D 65  neWelcome
00007311  20 20 74 6F 20 54 65 6C 6C    to Tell
0000731A  6F 62 33 32 31 65 39 61 64  ob321e9ad
00007323  2D 64 62 30 32 2D 34 64 39  -db02-4d9
0000732C  61 2D 39 64 64 63 2D 32 62  a-9ddc-2b
00007335  66 34 63 34 36 31 38 31 61  f4c46181a
0000733E  38 7B 05 14 55 55 05 39 00  8{..UU.9.
```

E:\MSIS\Thesis\ScreenShot\TelloDelChat\com.udna.tellotalk\databases\hist...

```
          0  1  2  3  4  5  6  7  8   012345678
0000726F  64 64 62 2D 34 32 36 65 2D  ddb-426e-
00007278  38 34 34 37 2D 35 65 66 31  8447-5ef1
00007281  31 39 31 33 61 33 37 30 81  1913a370.
0000728A  33 04 14 55 55 05 45 00 2F  3..UU.E./
00007293  08 08 08 00 00 00 08 00 09  .........
0000729C  08 08 0D 55 35 35 37 38 65  ...U5578e
000072A5  37 34 35 2D 61 35 38 65 2D  745-a58e-
000072AE  34 34 31 35 2D 62 37 38 35  4415-b785
000072B7  2D 34 32 33 66 38 39 30 31  -423f8901
000072C0  35 34 61 35 61 36 32 61 63  54a5a62ac
000072C9  33 61 63 2D 33 31 33 33 2D  3ac-3133-
000072D2  34 30 31 64 2D 61 35 30 65  401d-a50e
000072DB  2D 35 63 33 38 64 36 37 64  -5c38d67d
000072E4  38 31 31 34 01 5C 33 AD 77  8114.\3.w
000072ED  26 39 32 33 33 33 35 31 34  &92333514
000072F6  32 36 33 35 40 74 65 6C 6C  2635@tell
000072FF  6F 74 61 6C 6B 2F 70 68 6F  otalk/pho
00007308  6E 65 57 65 6C 63 6F 6D 65  neWelcome
00007311  20 20 74 6F 20 54 65 6C 6C    to Tell
0000731A  6F 62 33 32 31 65 39 61 64  ob321e9ad
00007323  2D 64 62 30 32 2D 34 64 39  -db02-4d9
0000732C  61 2D 39 64 64 63 2D 32 62  a-9ddc-2b
00007335  66 34 63 34 36 31 38 31 61  f4c46181a
0000733E  38 7B 05 14 55 55 05 39 00  8{..UU.9.
```

**Figure 4.50:** TelloTalk - recovered text message after deleting conversation

Similarly, media message can be recovered after user deletes conversation as shown in Figure 4.51.



E:\MSIS\Thesis\ScreenShot\TelloCal\com.udna.tellotalk\databases\history

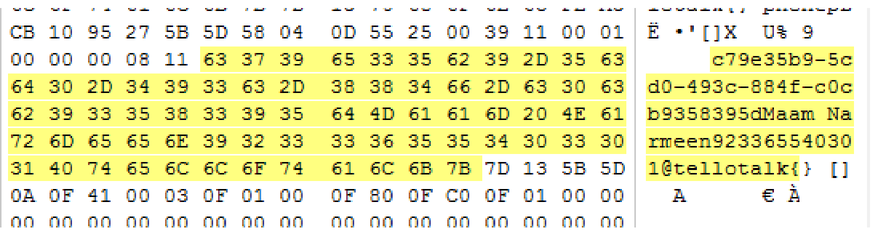
```
          0  1  2  3  4  5  6  7  8   012345678
000076DD  08 01 01 81 01 00 00 08 00  .........
000076E6  09 08 08 0D 00 31 34 32 64  .....142d
000076EF  65 62 35 35 2D 31 39 32 36  eb55-1926
000076F8  2D 34 61 36 32 2D 38 35 38  -4a62-858
00007701  39 2D 61 39 37 66 32 62 38  9-a97f2b8
0000770A  33 36 38 39 38 61 36 32 61  36898a62a
00007713  63 33 61 63 2D 33 31 33 33  c3ac-3133
0000771C  2D 34 30 31 64 2D 61 35 30  -401d-a50
00007725  65 2D 35 63 33 38 64 36 37  e-5c38d67
0000772E  64 38 31 31 34 01 5C 33 B0  d8114.\3.
00007737  16 79 39 32 33 33 33 35 31  .y9233351
00007740  34 32 36 33 35 40 74 65 6C  42635@tel
00007749  6C 6F 74 61 6C 6B 39 32 33  lotalk923
00007752  33 31 35 30 38 32 30 34 34  315082044
0000775B  2F 45 78 69 74 20 57 65 73  /Exit Wes
00007764  74 20 2D 20 4D 6F 68 73 69  t - Mohsi
0000776D  6E 20 48 61 6D 69 64 2E 65  n Hamid.e
00007776  70 75 62 7C 35 31 33 39 34  pub|51394
0000777F  35 08 02 2F 73 74 6F 72 61  5../stora
00007788  67 65 2F 65 6D 75 6C 61 74  ge/emulat
00007791  65 64 2F 30 2F 44 6F 77 6E  ed/0/Down
0000779A  6C 6F 61 64 2F 45 78 69 74  load/Exit
000077A3  20 57 65 73 74 20 2D 20 4D   West - M
000077AC  6F 68 73 69 6E 20 48 61 6D  ohsin Ham
```

E:\MSIS\Thesis\ScreenShot\TelloDelChat\com.udna.tellotalk\databases\hist...

```
          0  1  2  3  4  5  6  7  8   012345678
000076C2  01 00 00 08 00 09 08 08 0D  .........
000076CB  00 31 34 32 64 65 62 35 35  .142deb55
000076D4  2D 31 39 32 36 2D 34 61 36  -1926-4a6
000076DD  32 2D 38 35 38 39 2D 61 39  2-8589-a9
000076E6  37 66 32 62 38 33 36 38 39  7f2b83689
000076EF  38 61 36 32 61 63 33 61 63  8a62ac3ac
000076F8  2D 33 31 33 33 2D 34 30 31  -3133-401
00007701  64 2D 61 35 30 65 2D 35 63  d-a50e-5c
0000770A  33 38 64 36 37 64 38 31 31  38d67d811
00007713  34 01 5C 33 B0 16 79 39 32  4.\3..y92
0000771C  33 33 33 35 31 34 32 36 33  333514263
00007725  35 40 74 65 6C 6C 6F 74 61  5@tellota
0000772E  6C 6B 39 32 33 33 31 35 30  lk9233150
00007737  38 32 30 34 34 2F 45 78 69  82044/Exi
00007740  74 20 57 65 73 74 20 2D 20  t West - 
00007749  4D 6F 68 73 69 6E 20 48 61  Mohsin Ha
00007752  6D 69 64 2E 65 70 75 62 7C  mid.epub|
0000775B  35 31 33 39 34 35 08 02 2F  513945../
00007764  73 74 6F 72 61 67 65 2F 65  storage/e
0000776D  6D 75 6C 61 74 65 64 2F 30  mulated/0
00007776  2F 44 6F 77 6E 6C 6F 61 64  /Download
0000777F  2F 45 78 69 74 20 57 65 73  /Exit Wes
00007788  74 20 2D 20 4D 6F 68 73 69  t - Mohsi
00007791  6E 20 48 61 6D 69 64 2E 65  n Hamid.e
```

**Figure 4.51:** TelloTalk - recovered media message after deleting conversation

**Recovering deleted contact** Investigator can still recover a TelloTalk contact after user deletes it. As shown in Figure 4.52, I have manually recovered a deleted contact.

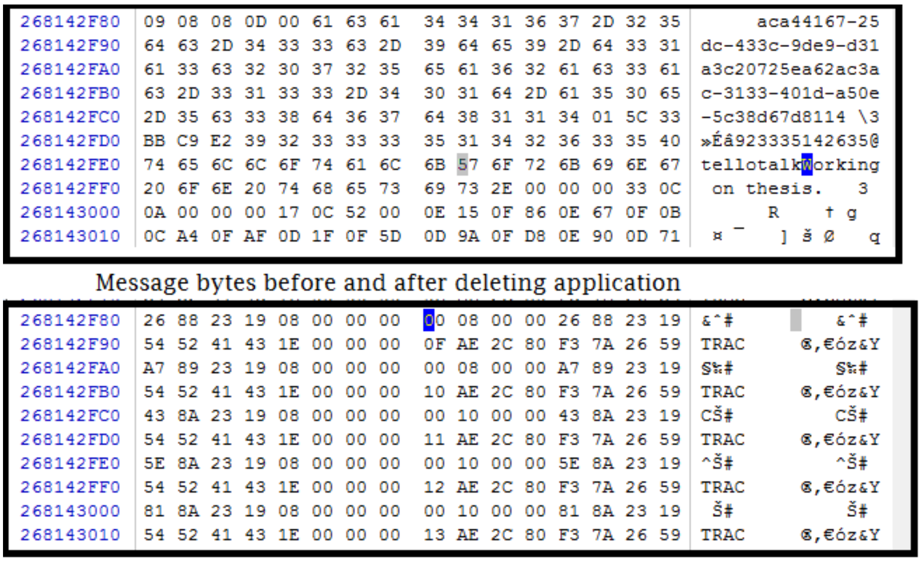


```
CB 10 95 27 5B 5D 58 04   0D 55 25 00 39 11 00 01   Ë •'[]X  U% 9
00 00 00 08 11 63 37 39   65 33 35 62 39 2D 35 63        c79e35b9-5c
64 30 2D 34 39 33 63 2D   38 38 34 66 2D 63 30 63   d0-493c-884f-c0c
62 39 33 35 38 33 39 35   64 4D 61 61 6D 20 4E 61   b9358395dMaam Na
72 6D 65 65 6E 39 32 33   33 36 35 35 34 30 33 30   rmeen92336554030
31 40 74 65 6C 6C 6F 74   61 6C 6B 7B 7D 13 5B 5D   1@tellotalk{} []
0A 0F 41 00 03 0F 01 00   0F 80 0F C0 0F 01 00 00     A      € À
00 00 00 00 00 00 00 00   00 00 00 00 00 00 00 00
```

**Figure 4.52:** TelloTalk - Recovered a deleted contact

### 4.4.4 Analysis of TelloTalk Chat & media shared between participants on device - Application Deleted

Once the TelloTalk application is deleted, TelloTalk over-writes all bytes, but save text messages in different location with user and timestamp information. Hence, investigator can not retrieve most of data. As shown in Figure 4.53, TelloTalk bytes are overwritten with gibberish.



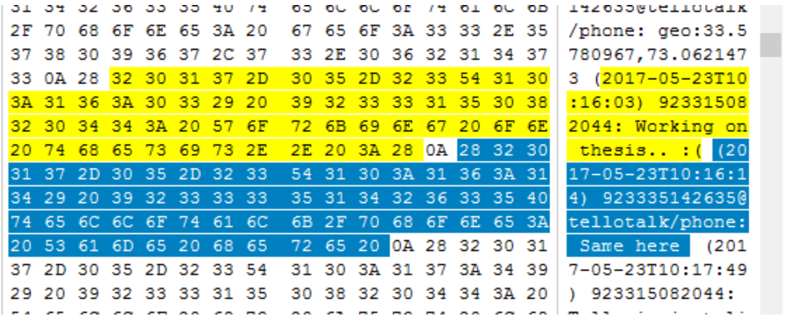
```
268142F80  09 08 08 0D 00 61 63 61  34 34 31 36 37 2D 32 35       aca44167-25
268142F90  64 63 2D 34 33 33 63 2D  39 64 65 39 2D 64 33 31  dc-433c-9de9-d31
268142FA0  61 33 63 32 30 37 32 35  65 61 36 32 61 63 33 61  a3c20725ea62ac3a
268142FB0  63 2D 33 31 33 33 2D 34  30 31 64 2D 61 35 30 65  c-3133-401d-a50e
268142FC0  2D 35 63 33 38 64 36 37  64 38 31 31 34 01 5C 33  -5c38d67d8114 \3
268142FD0  BB C9 E2 39 32 33 33 33  35 31 34 32 36 33 35 40  »Éâ923335142635@
268142FE0  74 65 6C 6C 6F 74 61 6C  6B 57 6F 72 6B 69 6E 67  tellotalkWorking
268142FF0  20 6F 6E 20 74 68 65 73  69 73 2E 00 00 00 33 0C   on thesis.   3
268143000  0A 00 00 00 17 0C 52 00  0E 15 0F 86 0E 67 0F 0B        R    † g
268143010  0C A4 0F AF 0D 1F 0F 5D  0D 9A 0F D8 0E 90 0D 71    ¤ ¯    ] š Ø   q
```

Message bytes before and after deleting application

```
268142F80  26 88 23 19 08 00 00 00  00 08 00 00 26 88 23 19  &^#         &^#
268142F90  54 52 41 43 1E 00 00 00  0F AE 2C 80 F3 7A 26 59  TRAC    ®,€óz&Y
268142FA0  A7 89 23 19 08 00 00 00  00 08 00 00 A7 89 23 19  §‰#         §‰#
268142FB0  54 52 41 43 1E 00 00 00  10 AE 2C 80 F3 7A 26 59  TRAC    ®,€óz&Y
268142FC0  43 8A 23 19 08 00 00 00  00 10 00 00 43 8A 23 19  CŠ#         CŠ#
268142FD0  54 52 41 43 1E 00 00 00  11 AE 2C 80 F3 7A 26 59  TRAC    ®,€óz&Y
268142FE0  5E 8A 23 19 08 00 00 00  00 10 00 00 5E 8A 23 19  ^Š#         ^Š#
268142FF0  54 52 41 43 1E 00 00 00  12 AE 2C 80 F3 7A 26 59  TRAC    ®,€óz&Y
268143000  81 8A 23 19 08 00 00 00  00 10 00 00 81 8A 23 19   Š#          Š#
268143010  54 52 41 43 1E 00 00 00  13 AE 2C 80 F3 7A 26 59  TRAC    ®,€óz&Y
```

**Figure 4.53:** TelloTalk - Messages bytes after deleting application

**Text Messages** Text messages that are recovered are shown in Figure 4.54.



```
31 34 32 36 33 35 40 74  65 6C 6C 6F 74 61 6C 6B  142635@tellotalk
2F 70 68 6F 6E 65 3A 20  67 65 6F 3A 33 33 2E 35  /phone: geo:33.5
37 38 30 39 36 37 2C 37  33 2E 30 36 32 31 34 37  780967,73.062147
33 0A 28 32 30 31 37 2D  30 35 2D 32 33 54 31 30  3 (2017-05-23T10
3A 31 36 3A 30 33 29 20  39 32 33 33 31 35 30 38  :16:03) 92331508
32 30 34 34 3A 20 57 6F  72 6B 69 6E 67 20 6F 6E  2044: Working on
20 74 68 65 73 69 73 2E  2E 20 3A 28 0A 28 32 30   thesis.. :( (20
31 37 2D 30 35 2D 32 33  54 31 30 3A 31 36 3A 31  17-05-23T10:16:1
34 29 20 39 32 33 33 33  35 31 34 32 36 33 35 40  4) 923335142635@
74 65 6C 6C 6F 74 61 6C  6B 2F 70 68 6F 6E 65 3A  tellotalk/phone:
20 53 61 6D 65 20 68 65  72 65 20 0A 28 32 30 31   Same here  (201
37 2D 30 35 2D 32 33 54  31 30 3A 31 37 3A 34 39  7-05-23T10:17:49
29 20 39 32 33 33 31 35  30 38 32 30 34 34 3A 20  ) 923315082044:
```

**Figure 4.54:** TelloTalk - Recovered messages after deleting application

## 4.5 Conclusion

In this chapter, WhatsApp, Viber and TelloTalk are forensically analyzed. Apart from tools mentioned in this section, I have analyzed each application using Access Data FTK Imager [31], Autopsy [32] and Magnet Axiom [33]. Using FTK Imager and Autopsy, all data (as mentioned above) can be extracted but manually. Magnet Axiom is sophisticated tool and can extract WhatsApp and Viber data automatically and present them in meaningful form. However, most of WhatsApp's and Viber's deleted data are extracted manually as well as all data from TelloTalk. In next section, all data gathered during investigation is analyzed.

Chapter 5

# Forensic Investigation Result and Analysis

## 5.1 Introduction

In this chapter, results gathered from each social messaging application is analyzed and lastly comparison of these applications is made.

## 5.2 Analysis of Whatsapp Application

In last chapter, forensic investigation of Whatsapp application was carried out. Results gathered from investigation are analyzed and presented in Table 5.1

| **No.** | **Scenarios** | **Results** | | |
|---|---|---|---|---|
| | | **Application Installed and Chat shared between participants** | **Conversation/ Chat deleted** | **Application Deleted** |
| 1. | Text and location messages sent/received | ✓ | ✓ | $\mathcal{P}$ |
| 2. | Media message sent/received | ✓ | ✓ | $\mathcal{P}$ |
| 3. | Whatsapp contacts | ✓ | ✓ | ✓ |
| 4. | Whatsapp Audio/Video Calls (incoming/outgoing/Missed) | ✓ | ✓ | $\mathcal{P}$ |
| 5. | Group text and media messages | ✓ | ✓ | $\mathcal{P}$ |

**Table 5.1:** Analysis of Whatsapp Application

✓- All data is recovered

✗- No data is recovered

$\mathcal{P}$ - Partial data is recovered, as some of data is overwritten

## 5.3 Analysis of Viber Application

Results gathered from Viber investigation are analyzed and presented in Table 5.2

| No. | Scenarios | Results | | |
|---|---|---|---|---|
| | | **Application Installed and Chat shared between participants** | **Conversation/ Chat deleted** | **Application Deleted** |
| 1. | Text and location messages sent/received | ✓ | $\mathcal{P}$ - Message<br>✓- Conservation | ✗ |
| 2. | Media message sent/received | ✓ | As above | ✗ |
| 3. | Viber Secret Messages | ✗ | ✗ | ✗ |
| 4. | Viber contacts | ✓ | ✓ | ✗ |
| 5. | Viber Audio/Video Calls (incoming/outgoing/Missed) | ✓ | ✓ | ✗ |
| 6. | Group text and media messages | ✓ | $\mathcal{P}$ - Message<br>✓- Conservation | ✗ |

**Table 5.2:** Analysis of Viber Application

✓- All data is recovered

✗- No data is recovered

$\mathcal{P}$ - If particular text/media messages are deleted in a conversation, body of that text is unrecoverable

## 5.4 Analysis of TelloTalk Application

In last chapter, forensic investigation of TelloTalk application was carried out. Results gathered from investigation are analyzed and presented in Table 5.3

| **No.** | **Scenarios** | **Results** | | |
|---|---|---|---|---|
| | | **Application Installed and Chat shared between participants** | **Conversation/ Chat deleted** | **Application Deleted** |
| 1. | Text and location messages sent/received | ✓ | $\mathcal{P}$ - Messaages<br>✓- Conversation | ✓ |
| 2. | Media message sent/received | ✓ | As above | ✗ |
| 3. | Tello contacts | ✓ | ✓ | ✗ |
| 4. | Group text and media messages | ✓ | $\mathcal{P}$ - Message<br>✓- Conversation | ✗ |

**Table 5.3:** Analysis of Tellotalk Application

✓- All data is recovered

✗- No data is recovered

$\mathcal{P}$ - If particular text/media messages are deleted in a conversation, body of that text is unrecoverable.

## 5.5 Comparison of Whatsapp, Viber and TelloTalk

When we look from cryptographic point of view, Whatsapp provides best security as it encrypts user's data stored in database files with .crypt12 algorithm and it also provides end-to-end encryption. Though latest version of Viber provides end-to-end encryption, but it doesn't encrypt its database files so data can be hacked very easily. Worst of all three is Pakistani developed social messaging application 'TelloTalk', as they do not provide any kind of security to its user.

From Forensic point of view, compassion of these three application are presented in Table 5.4

| **No.** | **Scenarios** | **Results** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **Application Installed and Chat shared between participants** | | | **Chat/conversation deleted** | | | **Application Deleted** | | |
| | | **W** | **V** | **T** | **W** | **V** | **T** | **W** | **V** | **T** |
| 1. | Text and location messages sent/received | ✓ | ✓ | ✓ | ✓ | Σ | Σ | 𝒫 | ✗ | ✓ |
| 2. | Media message sent/received | ✓ | ✓ | ✓ | ✓ | Σ | Σ | 𝒫 | ✗ | ✗ |
| 3. | Media Files | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| 4. | Contacts | ✓ | ✓ | ✓ | ✓ | ✓ | | ✓ | ✗ | ✗ |
| 5. | Audio/Video Calls (incoming/ outgoing/Missed) | ✓ | ✓ | ⋇ | ✓ | ✓ | ⋇ | 𝒫 | ✗ | ⋇ |
| 6. | Group text and media messages | ✓ | ✓ | ⋇ | ✓ | Σ | ⋇ | 𝒫 | ✗ | ⋇ |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 7. | Secret Messages | ⋇ | ✗ | ⋇ | | ✗ | ⋇ | ⋇ | ✗ | ⋇ |

**Table 5.4:** Comparison of Whatsapp, Viber and TelloTalk

W - WhatsApp

V - Viber

T - TelloTalk

✓- All data is recovered

✗- No data is recovered

$\mathcal{P}$ - Partial data is recovered

⋇ - Feature not available

$\Sigma$ - If particular text and media message is deleted in a conversation, all is recovered but body of that text and if all conversation is deleted, all data is recovered.

## 5.6 Guideline for Safe Deletion

After analysis of social messaging application on Android device, it is concluded that most of data can not be recovered if we follow this basic guideline:

- When user deletes an application, all data related to that application must be deleted including backup files, media files, database files and/or complete data directory from Android system.
- Not only these file must be deleted, but uninstall process of these application must overwrite all data with random or zero bytes.
- It has been seen that when user deletes a conversation or particular message rather than complete application, this conversation/message still lies in system memory. So when user deletes a conversation or a single text, there bytes must be overwritten.

- Since some applications do not provide safe deletion feature, it is responsibility of user to delete its data securely. There are many tools available in the market that overwrites empty/unallocated space with random bytes multiple times to ensure complete deletion.

Chapter 6

# Conclusion and Future Work

## 6.1 Introduction

In this chapter, we will draw conclusion and identify future work related to forensic investigation of social messaging applications. Beside this, we will suggest some guideline to secure Android applications.

## 6.2 Guidelines to Secure Android Application

All applications should provide highest level of security to its clients as these clients put their trust in the hands of application provider to keep their personal, official and even unimportant data secure.

Some applications tested during this research proved to be very secure. Here is list of important guidelines that must be followed to achieve highest level of security.

**Data Isolation -** Application must store files in internal memory as by default these files are only accessible to that application only.

**Scoped Directory Access -** Application should provide scoped directory access. Application's media folders that are stored either on internal storage or external are accessible to all other applications; folder access should not be user based but app based. If another application requires access to this folder, it should create a request.

**Encrypted files -** Application and user's data files should be encrypted in order to keep them secure.

**Data integrity -** All application should maintain user's data integrity, no one should be allowed to tamper user's data. Applications should implement integrity using the CBC or CTR mode with one of the following functions:

- HMAC-SHA1
- HMAC-SHA-256
- HMAC-SHA-512
- GCM mode

**Use of Android Keystore System -** Application should use Android keystore system to store keys as it made difficult to extract keys from store. It limits use of keys like it allows user authentication to utilize key or it allows key to be used in certain modes.

**End-to-end network encryption -** Social application should implement end-to-end encryption; no one should be able to read these messages except intended sender and receiver.

Do not root your device unless necessary because device loses root level security and anyone can access data that was only supposed to be accessed by root user.

Only enable USB debugging when required so no one can access device shell.

## 6.3 Future Work

Field of forensic is very vast, as new changes/versions are introduced very frequently. If one has performed research on Android 6.0.1, other researchers can work on latest version of Android devices. Other areas where this research can extend are:

- In this research, only three most used applications were forensically tested. Other social messaging application like Snapchat, Facebook messenger, Line can be tested too.
- We can examine social messaging application for all security checks mentioned in above section.
- Focus of this research was only on data stored in internal memory; we did not check the data/messages packets while transaction. One can check the security of messages sent/received over network.
- If phone is made *Factory Reset*, can data still be recovered?

## 6.4 Conclusion

In this research, three most used social messaging applications Whatsapp, Viber, and TelloTalk are forensically analyzed. TelloTalk is first Pakistani developed social messaging application and it was not forensically analyzed before. Though Whatsapp and Viber were analyzed before, no one has done extensive study that includes retrieving deleted data.

It is concluded that TelloTalk is worst in providing user security whereas Whatsapp proved to be best. Though latest version of Viber provides end-to-end encryption, its application and user files are not encrypted and data can easily be extracted. Whatsapp provides an extra layer of security by encrypting its database files, but forensic investigator can still extract data from it by decrypting these files. When Viber and TelloTalk application is deleted from mobile, most of data bytes are overwritten and difficult to retrieve. However in case of Whatsapp, investigator can extract most of data after deleting application.

As use of social messaging application in constantly on the rise, this research will prove to be very helpful in field of digital forensic.

# References

[1] Security enhancements in android 6.0 | android open source project. https://source.android.com/security/enhancements/enhancements60. (Accessed on 03/02/2018).

[2] Yaffs 2 specification | yaffs - a flash file system for embedded use. https://yaffs.net/yaffs-2-specification. (Accessed on 03/02/2018).

[3] Martin W Mckee, Paul T Schultz, and Robert A Sartini. Weighting social network relationships based on communications history, July 26 2016. US Patent 9,400,972.

[4] Alex S Taylor and Jane Vincent. An sms history. In *Mobile world*, pages 75–91. Springer, 2005.

[5] Instagram direct | instagram help centre. https://help.instagram.com/400205900081854. (Accessed on 03/28/2018).

[6] Whatsapp. https://www.whatsapp.com/. (Accessed on 03/02/2018).

[7] Most popular messaging apps 2018 | statista. https://www.statista.com/statistics/258749/most-popular-global-mobile-messenger-apps/. (Accessed on 03/28/2018).

[8] Messages between oscar pistorius and girlfriend read in court - cnn. https://edition.cnn.com/2014/03/24/world/oscar-pistorius-trial-whatsapp-messages/index.html. (Accessed on 03/28/2018).

[9] Italian investigation reveals whatsapp messages from isis - business insider. http://www.businessinsider.com/

italian-investigation-reveals-whatsapp-messages-from-isis-2016-7. (Accessed on 03/28/2018).

[10] The haunting last whatsapp message molly mclaren sent to a friend moments before she was stabbed to death - kent live. https://www.kentlive.news/news/kent-news/haunting-last-whatsapp-message-molly-1108297. (Accessed on 03/28/2018).

[11] Viber - free calls and messages. https://www.viber.com/. (Accessed on 03/03/2018).

[12] Tellotalk pakistan's first instant messager. http://www.tellotalk.com/. (Accessed on 03/03/2018).

[13] Kingoroot for android, the best one click root tool/apk for free. https://www.kingoapp.com/. (Accessed on 03/03/2018).

[14] Free download kingoroot android for windows. https://www.kingoapp.com/android-root/download.htm. (Accessed on 03/28/2018).

[15] Android debug bridge (adb) | android studio. https://developer.android.com/studio/command-line/adb.html. (Accessed on 03/03/2018).

[16] Timothy Vidas, Chengye Zhang, and Nicolas Christin. Toward a general collection methodology for android devices. *digital investigation*, 8:S14–S24, 2011.

[17] Linux dd command. http://www.linuxnix.com/what-you-should-know-about-linux-dd-command/. (Accessed on 03/28/2018).

[18] Cellebrite - ufed physical analyzer. http://ec2-107-23-31-70.compute-1.amazonaws.com/mobile-forensics/products/applications/ufed-physical-analyzer. (Accessed on 03/28/2018).

[19] Ufed physical analyzer - sales inquiry. https://www.cellebrite.com/en/sales-inquiry/?leadcat=Website. (Accessed on 03/28/2018).

[20] github.com. https://github.com/andreas-mausch/whatsapp-viewer. (Accessed on 03/28/2018).

[21] Db browser for sqlite. http://sqlitebrowser.org/. (Accessed on 03/03/2018).

[22] Winhex: Hex editor & disk editor, computer forensics & data recovery software. `https://www.x-ways.net/winhex/`. (Accessed on 03/03/2018).

[23] Noora Al Mutawa, Ibrahim Baggili, and Andrew Marrington. Forensic analysis of social networking applications on mobile devices. *Digital Investigation*, 9:S24–S33, 2012.

[24] Mohammad Iftekhar Husain and Ramalingam Sridhar. iforensics: forensic analysis of instant messaging on smart phones. In *International Conference on Digital Forensics and Cyber Crime*, pages 9–18. Springer, 2009.

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

[26] Cosimo Anglano, Massimo Canonico, and Marco Guazzone. Forensic analysis of the chatsecure instant messaging application on android smartphones. *Digital investigation*, 19:44–59, 2016.

[27] GB Satrya, PT Daely, and SY Shin. Android forensics analysis: Private chat on social messenger. In *Ubiquitous and Future Networks (ICUFN), 2016 Eighth International Conference on*, pages 430–435. IEEE, 2016.

[28] Aditya Mahajan, MS Dahiya, and HP Sanghvi. Forensic analysis of instant messenger applications on android devices. *arXiv preprint arXiv:1304.4915*, 2013.

[29] Reformatting lock - emui. `https://emui.huawei.com/en/unlock_detail`. (Accessed on 03/27/2018).

[30] Namheun Son, Yunho Lee, Dohyun Kim, Joshua I James, Sangjin Lee, and Kyungho Lee. A study of user data integrity during acquisition of android devices. *Digital Investigation*, 10:S3–S11, 2013.

[31] Accessdata - ftk imager. `https://accessdata.com/product-download/ftk-imager-version-4.2.0`. (Accessed on 04/02/2018).

[32] Autopsy. `https://www.sleuthkit.org/autopsy/`. (Accessed on 04/02/2018).

[33] Magnet axiom. `https://www.magnetforensics.com/magnet-axiom/`. (Accessed on 04/02/2018).