Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Forensic Analysis of Social Messaging Platforms on an Android Device

by

Habibur Rahman

Supervisor: Dr Suresh Shanmugasundaram

Project submitted as part of the requirements for the award of MSc Cyber Security and Digital Forensics

September 2026

Declaration Form and Form of Consent

DECLARATION

I, Habibur Rahman, declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed: Habibur Rahman

Date: 25/09/2026

FORM OF CONSENT

I, Habibur Rahman, hereby consent that my Project, submitted in candidature for the M.Sc. Cybersecurity and Digital Forensics, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed: Habibur Rahman

Date: 25/09/2026

Abstract

This project explores what happens to digital evidence when messages and media files are deleted from Google Chat, WhatsApp and Telegram. Although a message may disappear from an application, traces of it or its associated files may remain on the device. The study aims to identify these traces and compare how the three applications store and retain evidence after deletion.

The research was carried out using BlueStacks which is an Android emulator, with test accounts and specially created messages and media files. Two experiments were completed. The first focused on deleting text messages, while the second examined the deletion of images, documents, voice messages and text in separate stages. Evidence was collected before and after deletion using Android Debug Bridge (ADB). SHA-256 hashes were used to check file integrity and identify changes, while Autopsy and DB Browser for SQLite were used to examine the collected evidence.

The results showed that the three applications behaved differently. In Google Chat, a database record provided evidence of message deletion, although the original deleted text was not recovered. In WhatsApp, the voice file disappeared from the collected shared storage after deletion, but the image and document remained. In Telegram, the image, document and voice file were still present after local deletion.

One of the main challenges was accessing the applications' private databases. The available WhatsApp database was encrypted, and a readable Telegram message database could not be obtained. This limited the investigation to the evidence accessible through the selected tools and methods.

Overall, the project shows that deleting content from a messaging application does not always remove its associated files from accessible storage. It also provides a practical method for collecting, preserving and comparing forensic evidence from Android messaging applications.

Table of Contents

Declaration Form and Form of Consent ii
Abstract iii
List of Tables vi
List of Figures vi
Chapter 1: Introduction 1
1.1 Statement of the Problem 2
1.2 Project Aim and Objectives 4
1.2.1 Aim 4
1.2.2.1 Objective 1 4
1.2.2.2 Objective 2 4
1.2.2.3 Objective 3 4
1.3 Significance of the Study / Proposed Solution 5
Chapter 2: Literature Review 7
Chapter 3: Research Methodology 12
3.1 Research Philosophy and Approach 12
3.2 Experimental environment 13
3.3 Controlled Evidence Generation 19
3.4 Acquisition, Preservation and Analysis 24
3.5 Ethics, Validity, Reliability, and limitations 25
Chapter 4: Design Process 26
4.1 Design Objectives 26
4.2 Platform and tool selection 27
4.3 Evidence naming and folder structure 27
4.4 Experiment 1 design 31
4.5 Experiment 2 design 32
4.6 Hashing and Comparison Design 32
4.7 Autopsy and database Analysis Design 33
4.8 Design evaluation and transition to implementation 36
Chapter 5: Implementation 38
5.1 Environment configuration and evidence handling 38

5.2 Google Chat Experiment 1 implementation 38
5.3 WhatsApp and Telegram Experiment 1 implementation 41
5.4 Experiment 2 implementation 46
5.5 Google Chat implementation 47
5.6 WhatsApp implementation 49
5.7 Telegram implementation 54
5.8 Implementation summary 56
Chapter 6: Evaluation of Results 57
6.1 Evaluation framework 57
6.2 Google Chat Evaluation 57
6.3 WhatsApp Evaluation 58
6.4 Telegram Evaluation 59
6.5 Cross-Platform comparison 60
6.6 Research objectives and interpretation 61
6.7 Limitations 62
Chapter 7: Conclusion and Recommendations 63
7.1 Conclusion for Google Chat 63
7.2 Conclusion for WhatsApp 64
7.3 Conclusion for Telegram 64
7.4 Overall Contribution and Cross-Platform Conclusion 65
7.5 Recommendations 66
7.6 Final Conclusion 67
Chapter 8: Reflection 69
8.1 Managing My Time 69
8.2 How My Study Skills Improved 70
8.3 What I Learned About the Way I Study 70
8.4 Connecting Theory with Practice 71
8.5 How I Could Have Improved My Work 71
8.6 Conclusion 72
Chapter 9: References 73

List of Tables

Table 1: Research methodology used in the project 13
Table 2: Tools used in the experiments 14
Table 3: Experimental Design 19
Table 4: Design requirements and controls 26
Table 5: Experiment 2 staged evidence states 32
Table 6: Experiment 1 database-level implementation outcome 1 38
Table 7: Experiment 1 database-level implementation outcome 2 41
Table 8: Main Experiment 2 implementation observations 56
Table 9: Cross-platform comparison of the main findings 60
Table 10: Main limitations of the study 62
Table 11: Main Contributions of the Project 65

List of Figures

Figure 1: BlueStacks App Version 14
Figure 2: Autopsy App Version 15
Figure 3: DB Browser for SQLite Version 16
Figure 4: Google Chat App Version 17
Figure 5: Telegram App Version 17
Figure 6: WhatsApp App Version 18
Figure 7: Experimental environment and ADB connection 18
Figure 8: Experiment 1 text-message deletion workflow 19
Figure 9: Controlled Google Chat App Environment in Experiment 1 20
Figure 10: Controlled WhatsApp App Environment in Experiment 1 20
Figure 11: Controlled Telegram App Environment in Experiment 1 21
Figure 12: Experiment 2 staged deletion workflow 21
Figure 13: Controlled Google Chat App Environment in Experiment 2 22
Figure 14: Controlled WhatsApp App Environment in Experiment 2 23

Figure 15: Controlled Telegram App Environment in Experiment 2 23
Figure 16: SHA-256 hash verification of the WhatsApp database before deletion 24
Figure 17: Android_Messaging_Forensics project folder 28
Figure 18: Google Chat Folder Environment for Experiment 1 29
Figure 19: Google Chat Folder Environment for Experiment 2 29
Figure 20: Telegram Folder Environment for Experiment 1 30
Figure 21: Telegram Folder Environment for Experiment 2 30
Figure 22: WhatsApp Folder Environment for Experiment 1 31
Figure 23: WhatsApp Folder Environment for Experiment 2 31
Figure 24: Hash (SHA-256) View 33
Figure 25: Autopsy Selection of Data Source Type 34
Figure 26: Autopsy Configuration Ingest 1 34
Figure 27: Autopsy Configuration Ingest 2 35
Figure 28: DB Browser for SQLite View 36
Figure 29: Experiment 1 Google Chat View 39
Figure 30: Experiment 1 Google Chat csv details 39
Figure 31: Running run-as command 40
Figure 32: Google Chat tombstone record in DB Browser for SQLite 40
Figure 33: Google Chat SQL query 41
Figure 34: Acquisition of the encrypted WhatsApp database during Experiment 1 42
Figure 35: Experiment 1 of WhatsApp Before text Deletion 42
Figure 36: Experiment 1 of WhatsApp After text Deletion 43
Figure 37: Log file of WhatsApp Experiment 1 43
Figure 38: Hash Verification of WhatsApp Experiment 1 44
Figure 39: Telegram Log of Experiment 1 44
Figure 40: Telegram before text deletion Experiment 1 45
Figure 41: Telegram After text deletion Experiment 1 45
Figure 42: Autopsy Result of Telegram Experiment 1 46
Figure 43: Google Chat Log File Analysis for Experiment 2 47
Figure 44: Google Chat Before Autopsy Evidence Experiment 2 48
Figure 45: Google Chat After Autopsy Evidence Experiment 2 48
Figure 46: WhatsApp Log File Analysis for Experiment 2 49
Figure 47: Before Deletion WhatsApp in Experiment 2 50
Figure 48: After Deletion of Document in WhatsApp in Experiment 2 50
Figure 49: After Deletion of Image in WhatsApp in Experiment 2 51
Figure 50: After Deletion of Text in WhatsApp in Experiment 2 51
Figure 51: After Deletion of Voice in WhatsApp in Experiment 2 52
Figure 52: WhatsApp voice evidence before deletion for Experiment 2 53

Figure 53: WhatsApp final search after voice deletion for Experiment 2 53
Figure 54: Telegram Log File Analysis of Experiment 2 54
Figure 55: Telegram voice evidence Before deletion for Experiment 2 55
Figure 56: Telegram voice evidence After deletion for Experiment 2 55
Figure 57: Cross-platform Experiment 2 comparison 61

Chapter 1: Introduction

Instant messaging has become a normal part of everyday communication. Applications such as WhatsApp, Telegram, Snapchat, Facebook Messenger, Signal, Google Chat, Viber, WeChat and similar platforms are now used for conversations that, in the past, may have taken place face-to-face, over a phone call, or through SMS. People use these applications to share a wide range of information, from casual personal conversations to financial discussions, private matters, and, in some cases, activities linked to criminal behavior. Because of this, when a mobile phone is involved in an investigation, messaging application data can become one of the most important sources of digital evidence.

Android is the most widely used mobile operating system globally, which makes it highly relevant in digital forensic investigations [1]. According to StatCounter Global Stats, Android accounted for 68.36% of the worldwide mobile operating-system market in July 2026, making it the most widely used mobile operating system globally [2]. Consequently, Android devices are frequently relevant to digital forensic investigations, requiring specialised methods and tools for the acquisition and analysis of digital evidence [3]. However, examining messaging applications on Android devices is not straightforward. These applications are updated regularly, and each one stores its data differently. Some applications also use strong security, and encryption features to protect user privacy, which can make forensic access more difficult. As a result, investigators may find that forensic tools do not always recover the same data, especially if the tools have not been tested against the latest versions of the applications.

This issue becomes even more complicated when a single Android device contains multiple messaging applications. For example, WhatsApp, Telegram, and Google Chat may all be installed on the same device, but each application may use different storage structures, encryption methods, and security controls. The same forensic acquisition method may therefore produce different results depending on the application being examined. In practice, this creates uncertainty for investigators, particularly when they need to decide which method is most reliable or how to interpret differences between forensic tools.

At present, there is limited clear guidance that compares how different Android messaging applications respond to forensic acquisition and analysis methods. This

creates a gap in practice, as investigators may not have a consistent framework for selecting the most suitable method for each application. This project aims to address that gap by examining the forensic challenges associated with Android instant messaging applications and by comparing how different tools and methods perform when applied to commonly used platforms such as WhatsApp, Telegram, and Google Chat.

1.1 Statement of the Problem

Research into the forensic analysis of mobile messaging applications has increased in recent years. However, much of the existing work tends to focus on a single application, or on testing one forensic tool against a fixed dataset. While this is understandable because of the technical complexity involved, it does not fully reflect the reality of investigative work. In real cases, examiners often have to deal with several messaging applications on the same device, running on newer versions of Android, and using forensic tools that may not perform equally well across all applications.

One of the main difficulties is that WhatsApp, Telegram, and Google Chat do not store their data in the same way. WhatsApp stores message data in an SQLite database, with encrypted backups commonly found on the device's storage. Telegram uses a different approach, as some data may be stored locally while other content is linked to cloud-based storage. Its secret chats also use separate security controls.

Google Chat uses a more cloud-based approach to data storage. Messages and attachments are mainly stored in Google's data centres and synchronised through the user's Google Account. This content is encrypted both during transmission and while stored. However, some information may also be kept locally when it is available for offline access. On Android devices, recoverable evidence may include messages, timestamps, account details, conversation identifiers, attachments, databases, and cached images. The amount of evidence available will depend on factors such as account synchronisation, the application version, offline access, the acquisition method, and access to the app's protected storage. Therefore, a forensic examination of Google Chat may need to consider evidence stored both on the device and in the cloud.

Another issue is that the internal structure of application databases can change after updates. This is often referred to as schema drift. When an application changes the way it organises its tables, fields, or stored records, a forensic tool may no longer interpret

the data correctly. The problem is that this failure is not always obvious. A tool might still process the database and produce a report that looks complete, even though some evidence has been missed because the tool's parsing rules are based on an older version of the application. For investigators, this is a serious concern, especially when the recovered data is being used to build a timeline or support conclusions in a case.

Disappearing messages create another challenge. WhatsApp and Telegram provide features that allow messages to delete themselves after a set period. In normal use, these messages may no longer be visible to the user once they expire. However, deletion from a database does not always mean that the data is immediately removed from the physical storage. In SQLite databases, deleted records may remain for a time in unused areas of the database until that space is overwritten or cleaned through processes such as vacuuming. This means that, depending on the database activity and the time since deletion, fragments of deleted messages may sometimes still be recoverable through detailed examination of the database file.

The Android operating system itself also adds further complications. Newer versions of Android have introduced stronger storage restrictions, including file-based encryption and scoped storage. These changes limit how applications store data and how forensic tools can access application directories. As a result, acquisition methods that may have worked on older Android versions may return incomplete results, or fail altogether, on newer devices. This shows that forensic procedures need to be reviewed regularly, not only when messaging applications change, but also when Android itself is updated.

Overall, these issues show why the absence of recovered messages should not automatically be treated as proof that no relevant communication took place. Missing evidence may be the result of encryption, tool limitations, changes in database structure, or restrictions introduced by the Android operating system. This is important because digital evidence is often used to support legal decisions, and incorrect interpretation can affect the fairness and accuracy of an investigation.

The central problem addressed by this project is therefore the lack of clear, structured, and comparative guidance for examining multiple Android messaging applications under realistic conditions. In particular, the project focuses on how differences in storage, encryption, synchronisation, database structures, and disappearing-message features affect the forensic analysis of WhatsApp, Telegram, and Google Chat.

1.2 Project Aim and Objectives

1.2.1 Aim

The aim of this project is to examine the forensic traces left by WhatsApp, Telegram, and Google Chat on Android devices or Android emulator, and to assess how these traces can be identified, extracted, and interpreted in a consistent way. By comparing the applications under a structured methodology, the project seeks to provide clearer guidance on how messaging evidence can be handled during Android forensic investigations.

1.2.2.1 Objective 1

The first objective is to review existing research on the forensic analysis of Android messaging applications. This will help establish what is already known about the recovery of evidence from platforms such as WhatsApp, Telegram, and Google Chat. The review will also consider how encryption, application updates, and changes in database structure can affect the performance of forensic tools. In addition, it will examine previous findings on the recovery of deleted and disappearing messages. This objective is important because it places the present study within the wider body of research and helps identify the gaps that this project aims to address.

1.2.2.2 Objective 2

The second objective is to carry out a controlled forensic examination of WhatsApp, Telegram, and Google Chat using open-source tools such as ADB, DB Browser for SQLite, and Autopsy. The recovered evidence will be checked against time-stamped records of the test activities, and SHA-256 hashes will be used to confirm that the data remains unchanged. The tools will then be compared based on the type, amount, completeness, and accuracy of the evidence they recover.

1.2.2.3 Objective 3

The third objective is to validate the artefacts recovered during the investigation by comparing them with a time-stamped record of the test activities carried out during the controlled scenarios. This will help confirm whether the messages, files, timestamps, and other recovered data accurately reflect the actions performed on the device. SHA-

256 hashing will also be used at each acquisition stage to verify the integrity of the evidence and to ensure that the acquisition process has not changed the data being examined.

It is evaluate to whether the results support the practical use of these open-source tools in Android messaging-app investigations. This will involve considering their reliability, limitations, and overall suitability for real-world forensic work.

1.3 Significance of the Study / Proposed Solution

The practical value of this research lies mainly in the fact that it compares several applications and tools in the same study. Much of the existing research in mobile messaging forensics focuses on one application or one forensic tool at a time. While this is useful, it does not fully help investigators who may have to examine WhatsApp, Telegram, and Google Chat on the same Android device or Android Emulator. A study that only looks at WhatsApp, for example, does not necessarily show how the same forensic method would perform on Telegram or Signal. This project therefore aims to provide a clearer comparison by applying the same methodology across these applications and assessing the results produced by different forensic tools.

For forensic practitioners, the most immediate benefit of this research is that it can help inform the choice of acquisition method. In real investigations, examiners often have limited time and must decide which tool or method is most suitable for a seized device. By comparing the results from ADB-based acquisition, SQLite database inspection, and Autopsy analysis, this study can support more informed decision-making during Android forensic examinations.

The study is also useful because it considers the issue of application updates and database changes. Messaging applications are updated regularly, and these updates may change the structure of their databases. If forensic tools are not updated at the same pace, they may fail to recover certain artefacts or may interpret the data incorrectly. By identifying any differences between what is actually present on the device and what each tool is able to recover, this project can highlight the importance of regularly validating forensic tools against current application versions.

Another important contribution is the way the research may help with the interpretation of missing evidence. In digital investigations, the absence of recovered messages does

not always mean that no communication took place. It may be the result of encryption, limited access to application data, tool failure, or changes in the database structure. This can be difficult to explain in legal proceedings. By clearly documenting what each method recovers and what it does not recover, the project can support a more transparent and defensible explanation of forensic findings.

More broadly, messaging evidence now plays an important role in many criminal investigations. The ability to recover, validate, and explain this evidence accurately can have a direct impact on the quality of an investigation and the fairness of any legal process that follows. For this reason, improving the reliability of Android messaging application forensics is not only a technical issue, but also an important part of supporting accurate and trustworthy criminal justice outcomes.

Chapter 2: Literature Review

A literature review was conducted to understand what previous researchers have found about forensic analysis of social messaging applications on Android devices. Using a systematic approach helped to make sure that the review considered a broad range of relevant studies, rather than focusing only on research that supports one argument. It also made the process more organised and easier for others to follow.

A structured search was carried out across IEEE Xplore, ScienceDirect, and ACM Digital Library, with Google Scholar used as a supplementary source to capture relevant conference proceedings and grey literature not indexed in the primary databases. The search terms used included “Android messaging forensics”, “WhatsApp forensics”, “Telegram forensics”, “mobile messaging digital forensics”, “SQLite forensics Android”, “schema drift forensics”, “disappearing messages forensics”, “SQLCipher forensic analysis”, and “mobile forensics NIST framework”, applied individually and in combination. Publication dates were restricted to 2014 onwards, a lower bound that corresponds to the earliest directly relevant peer-reviewed study in this domain, and searches were concluded in April 2026.

Papers were kept for inclusion if they focused on the forensic analysis of a mobile messaging application, involved Android devices or Android-compatible environments, appeared in peer-reviewed journals or indexed conference proceedings, and were available in full text in English. Studies concerned exclusively with iOS devices, those limited to network-level interception without any device-side component, and those examining platforms no longer in active use were excluded. Systematic and narrative review articles were included where they offered relevant findings. Following this screening process, twelve studies were selected for inclusion. Each was assessed against four criteria: firstly whether the study described its acquisition tools and methods clearly, whether it used controlled or verifiable test conditions, whether findings were presented with sufficient detail to be independently evaluated, and whether limitations were acknowledged. All twelve studies met at least three of the four criteria and were retained without reservation.

The body of research that meets these criteria begins, historically, forensic analysis of WhatsApp on Android established the methodological template that most subsequent work in this area has followed [4]. At the time of that study, WhatsApp did not encrypt

its local database, which allowed to recover substantial forensic material, including chat history, contact records, and media content, directly from the SQLite databases, cache files, and shared preferences stored on the device [4]. The importance of this work is not simply the volume of evidence recovered but the demonstration that the SQLite database is the primary container of messaging evidence on Android, a finding that has anchored the field ever since. That said, the security conditions under which researcher worked show little resemblance to what investigators face today, and the study is best understood as a baseline against which subsequent developments can be measured [4].

Another research conducted of the scope of single-application research was expanded considerably, who examined twenty popular Android messaging applications in a single comparative study and found that message content could be reconstructed from sixteen of them [5] . This result immediately established that recoverability is not a uniform property across platforms and that the tool or method appropriate for one application may return nothing useful on another. Equally significant was the observation that device forensics and network forensics can function as complementary rather than competing strategies, with network traffic sometimes revealing information that device extraction missed, and vice versa. That insight directly informed the later work of which [6] applied from [7] to a structured comparative analysis of Android messaging applications and incorporated network traffic capture alongside device-level acquisition as a planned component of the methodology rather than a contingency. The NIST framework, which organises the forensic process into four stages of preservation, acquisition, examination, and reporting, provides the procedural baseline that makes findings from this kind of research defensible in a legal context, and both studies benefit from its structure. The adoption of that same framework in the present study reflects its established standing in the field and the fact that it enables a level of methodological consistency that ad hoc approaches cannot provide [7].

Another paper discussed the challenges of disappearing messages [8] , that analysed Signal, Wickr, and Threema, three platforms that all offer timer-based ephemeral messaging as a standard feature. Their research found that deleted messages do not always disappear immediately from a forensic perspective. When a message is removed from a SQLite database, the space it occupied becomes unallocated but is not immediately overwritten. Fragments of deleted content can persist in these unallocated pages until SQLite's vacuuming process reclaims the space, at which point recovery

becomes impossible. The window for recovery is therefore directly tied to how much subsequent database activity has occurred, which in practice means that early acquisition significantly improves the likelihood of recovering ephemeral content. This finding is relevant not only to Signal but to Telegram, whose secret chat feature stores content in a separately encrypted portion of the local database [8], and to WhatsApp, which has introduced its own disappearing message functionality. Extended the understanding of deleted data recovery further by examining WhatsApp, Telegram, and Instagram in a virtual Android environment and found that deleted content can leave residual traces in local cache files even after removal from the primary database [9]. They also found that some content remains accessible through server-side storage depending on the platform, which points to the investigative relevance of data requests to platform operators in cases where device-level extraction is incomplete.

A problem that cuts across all these application-specific findings is the reliability of the forensic tools used to recover evidence. Addressing this directly through an analysis of 320 versions of several messaging applications, finding that changes in database structure across those versions, referred to as schema drift, create persistent problems for forensic tools [1]. The concern is not that tools fail visibly but that they can process a database and produce output that appears complete while having silently missed evidence, because their parsing definitions are based on an earlier version of the application's schema. An investigator reviewing that output would see nothing to indicate that any evidence had been missing. An investigation of this problem under controlled conditions and confirmed that well-established commercial tools are affected by schema drift in exactly this way [10]. Because messaging applications are updated frequently and forensic tool certification cycles operate on a much slower timeline, the gap between what a tool expects and what it encounters in a current case can be substantial. The combined implication of these two studies is that the absence of errors in a forensic report is not sufficient evidence of a complete extraction, and that tools need to be validated against current application versions on a regular basis rather than assumed to be accurate because they have been certified against older ones.

A related but distinct practical problem is the identification of SQLite database files in the first place. Another researcher noted that messaging applications do not always assign recognisable file extensions to their SQLite databases, with the result that automated forensic tools relying on extension-based detection can miss them entirely

[11]. In response, they developed FORC, a tool that identifies SQLite files by inspecting binary file headers rather than relying on extensions [11]. This approach improves complete detection and is particularly relevant in the context of manual verification, where a researcher needs to be confident that all candidate database files have been located before analysis begins. Taken in combination with the schema drift findings of [1] and [10], the work of [11] paints a picture of forensic tool reliability that should give investigators pause: files can be missed at the detection stage, and evidence can be missed at the parsing stage, in both cases without any visible indication that something has gone wrong.

The most comprehensive perspective on the field is provided by researcher [12], whose systematic review published in ACM Computing Surveys examined a wide body of research and concluded that the widespread adoption of end-to-end encryption has fundamentally changed what forensic investigators can achieve through communication interception. The focus has shifted from content capture in transit to the recovery of device-level artefacts, and the literature has adapted accordingly. Their review also identified what they characterised as a significant structural gap: the shortage of comparative, cross-platform studies that apply a consistent methodology across several applications and present results in a form that other researchers can replicate. Most published work, they found, is too narrow in scope or too tied to a specific application version to serve as a reliable basis for broader guidance. They called explicitly for a unified and regularly updated evaluation framework for secure messaging forensics, and that call remains largely unanswered in the literature.

It is that gap which the present study is designed to address. The reviewed literature has established that SQLite databases are the primary source of device-level messaging evidence on Android [4], that recoverability varies significantly across platforms and is not reliably predictable from device-level extraction alone [5] [13], that schema drift creates an ongoing risk of silent evidence loss even in well-validated tools [1][10], that deleted and ephemeral content can sometimes be recovered but within a narrow and time-sensitive window [8] [9], and that standard file detection methods may miss relevant databases unless header-based identification is used [11]. What is absent from the existing body of research is a important study that tests multiple forensic tools across WhatsApp, Telegram, and Google Chat simultaneously on current Android

hardware, under a single reproducible methodology grounded in the NIST framework [7][6], producing results that can be compared directly across platforms.

Chapter 3: Research Methodology

This chapter describes the methodology used in this research. The study was designed as controlled digital forensics experiment rather than a study involving human participants. Google Chat, WhatsApp and Telegram were tested in the same Android environment-based emulator so that the behaviour of the three applications could be compared under similar conditions. The focus was not simply whether a message disappeared from the screen, but what evidence could still be acquired and examined after deletion.

3.1 Research Philosophy and Approach

A pragmatic research philosophy was used because the project was driven by a practical forensic problem: identifying useful artefacts after controlled messaging activity and deletion. The work was primarily deductive. Expected artefacts such as message records, media files, metadata and deletion indicators were examined using repeatable procedures. Quantitative measurements, including file counts, file sizes, hash values and Autopsy search results, were supported by qualitative interpretation of database fields, storage behaviour and tool limitations.

Element	**The Choice which is used in this project**	**Reason for use**
Philosophy	Pragmatism	Focuses on practical evidence and what can be demonstrated.
Approach	Primarily deductive	Tests expected forensic artefacts through controlled actions.
Strategy	Controlled experiment	Keeps app activity and deletion events known to the researcher.
Method	Mainly quantitative with qualitative interpretation	Combines measurable file changes with forensic interpretation.
Analysis	Before/after comparison	Shows what persisted, changed or disappeared after deletion.

Table 1: Research methodology used in the project

In *Table 1* summarizes the research approach used in this study. Pragmatism was selected because the research focuses on the practical examination of forensic evidence. A controlled experimental approach was used to compare the three applications under similar conditions. File counts, sizes and SHA-256 hashes provided measurable results, while database records and tool limitations required further interpretation.

The study is an applied controlled experiment rather than a population survey. Its purpose is not to claim that every Android device, application version or forensic tool will produce the same result. It investigates Google Chat, WhatsApp and Telegram in one reproducible emulator, which is android based environment. This scope allows application-specific behavior to be compared while preserving the same broad acquisition and analysis logic. The methodology distinguishes acquisition, identification and recovery. Acquisition means obtaining an evidence object on the other hand, Identification means associating an acquired artifact with the controlled experiment, and lastly Recovery means obtaining the original deleted content or another explicitly defined target. This distinction prevents a successful backup command or a deletion-related record from being described automatically as recovered message content.

Two experiments were used. Experiment 1 focused on controlled text-message deletion. Experiment 2 extended the design to text, image, document and voice/audio evidence. The second experiment was added to improve the study by testing whether different content types produced different observable shared-storage outcomes and by isolating deletion events sequentially.

3.2 Experimental environment

In following Table 2, it explains what kind of tools are used in this experiment.

Software	**Recorded Version**
Host OS	Windows 11
BlueStacks	App Player 5.22.233.1003
Android	Android 9 previously reported; exact build not independently reconfirmed
ADB	1.0.41; Platform Tools 37.0.1-15733141
ADB connection	127.0.0.1:5555
ADB executable	D:\UWTSD\adb\platform-tools\adb.exe

Autopsy	4.23.1 RELEASE; Sleuth Kit 4.15.0
DB Browser for SQLite	3.13.1; SQLite 3.46.1
Google Chat	2026.06.15.932290237.release | com.google.android.apps.dynamite
WhatsApp	2.26.23.74 | com.whatsapp
Telegram	12.8.3 | org.telegram.messenger

Table 2: Tools used in the experiments

The experiments were performed on a Windows 11 host using BlueStacks App Player. ADB was selected because it provides a transparent command-line method for checking packages, testing permissions and copying accessible shared storage. PowerShell was used to execute commands, preserve transcripts and command histories, calculate SHA-256 values and compare manifests. Autopsy was used to create separate logical-file cases and perform keyword and file-oriented examination. DB Browser for SQLite was used only when a readable SQLite database existed. The acquisitions were logical copies of accessible shared storage. They were not physical device images. Direct access attempts to private application directories were documented, as were run-as failures and Backup Manager outcomes. The study therefore evaluates the evidence exposed by these specific methods rather than assuming full device access.

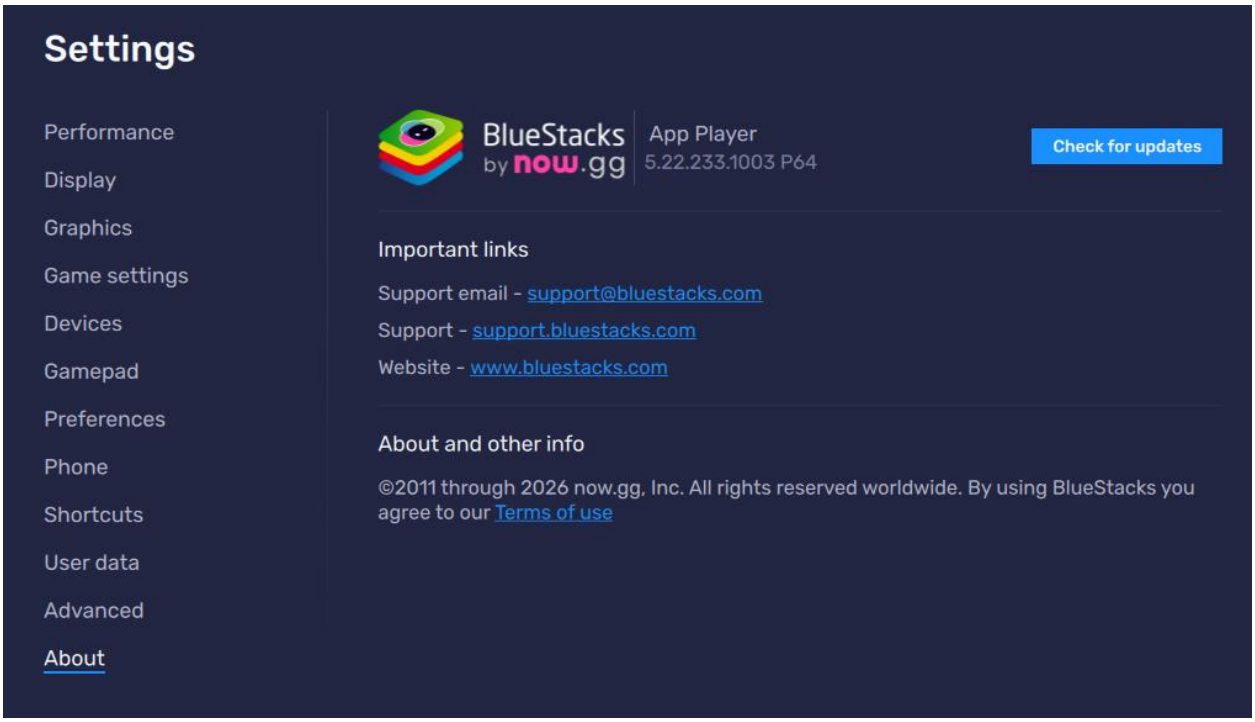


Figure 1: BlueStacks App Version

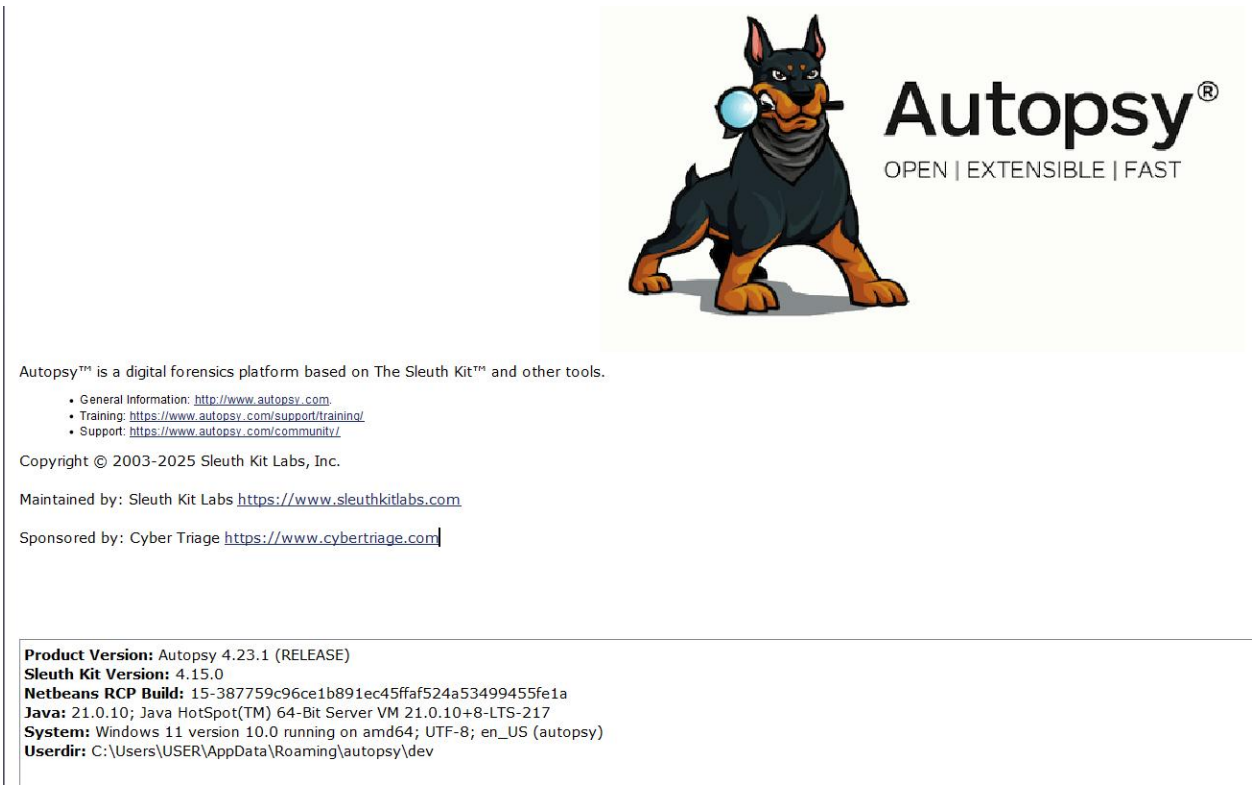


Figure 2: Autopsy App Version

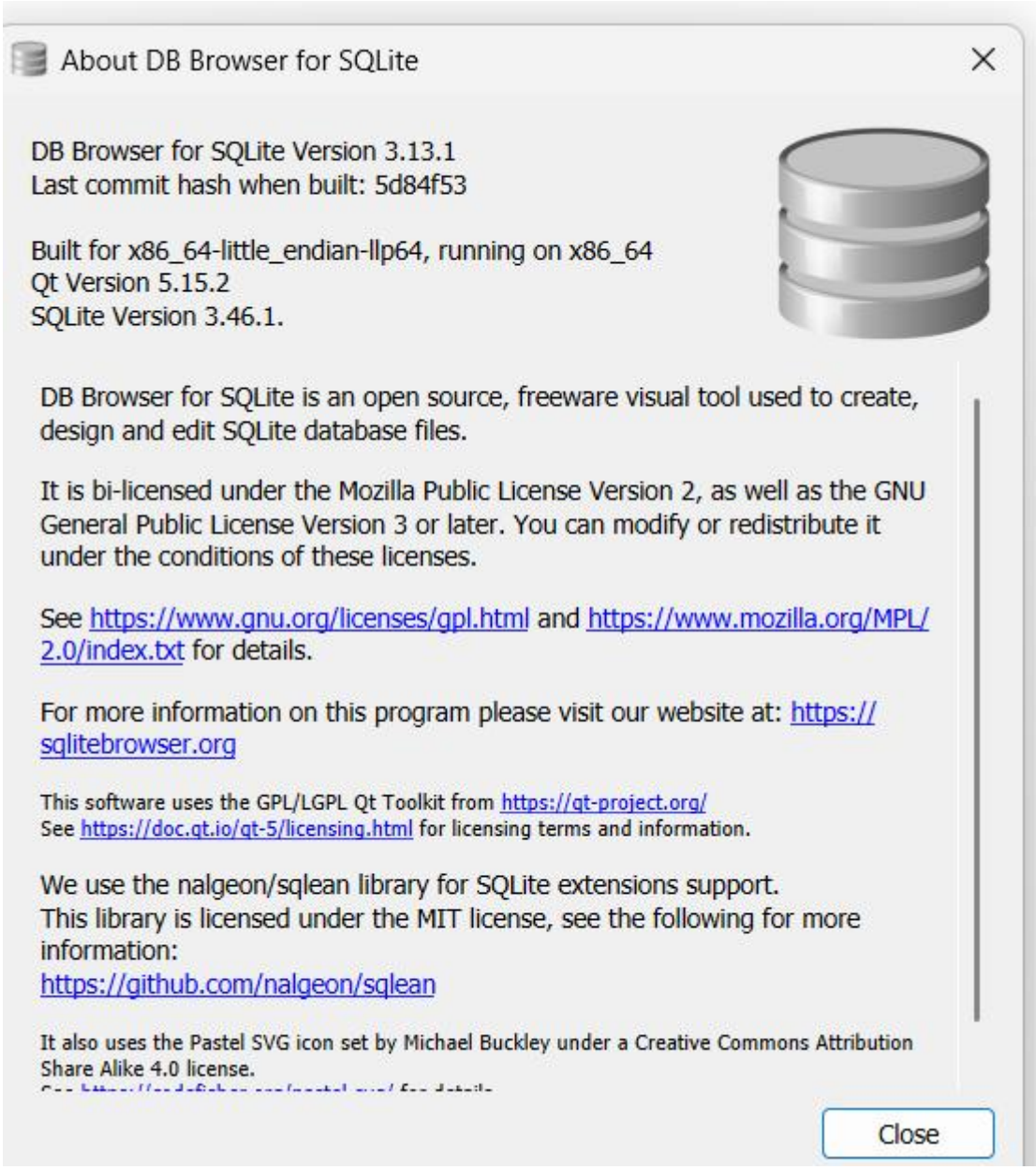


Figure 3: DB Browser for SQLite Version

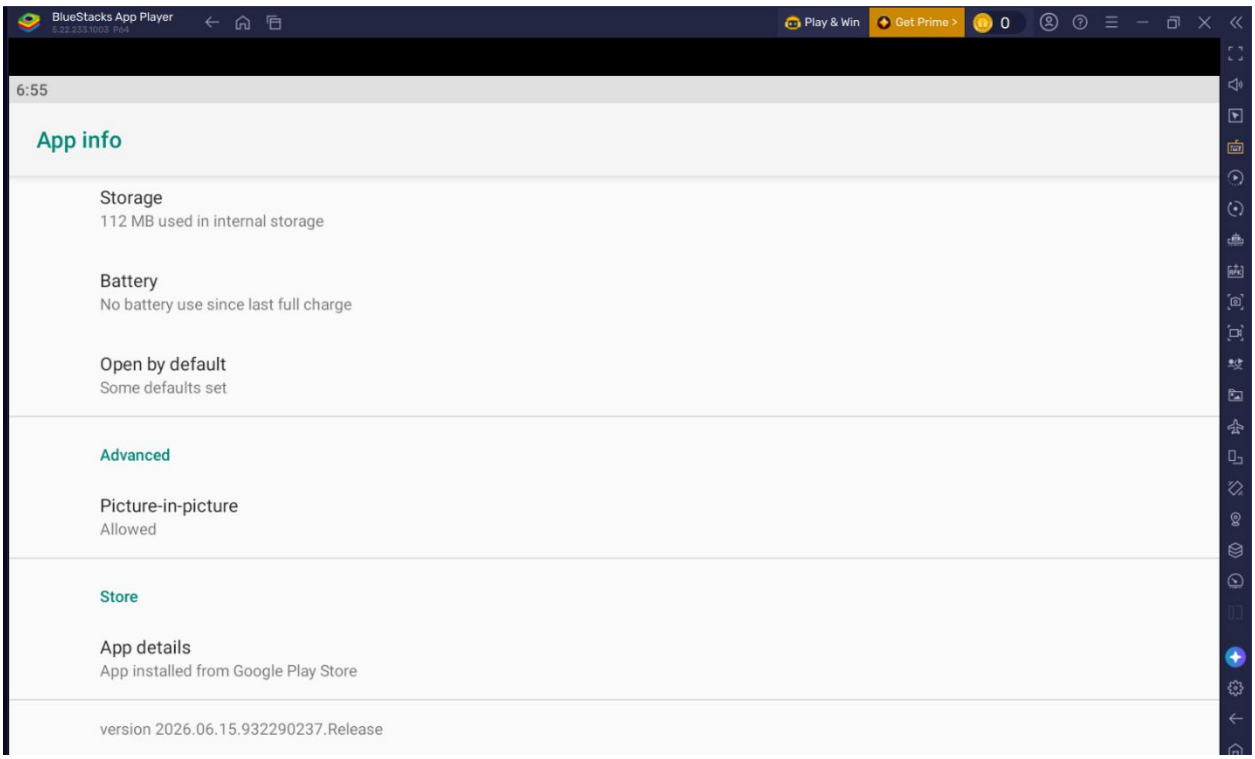


Figure 4: Google Chat App Version

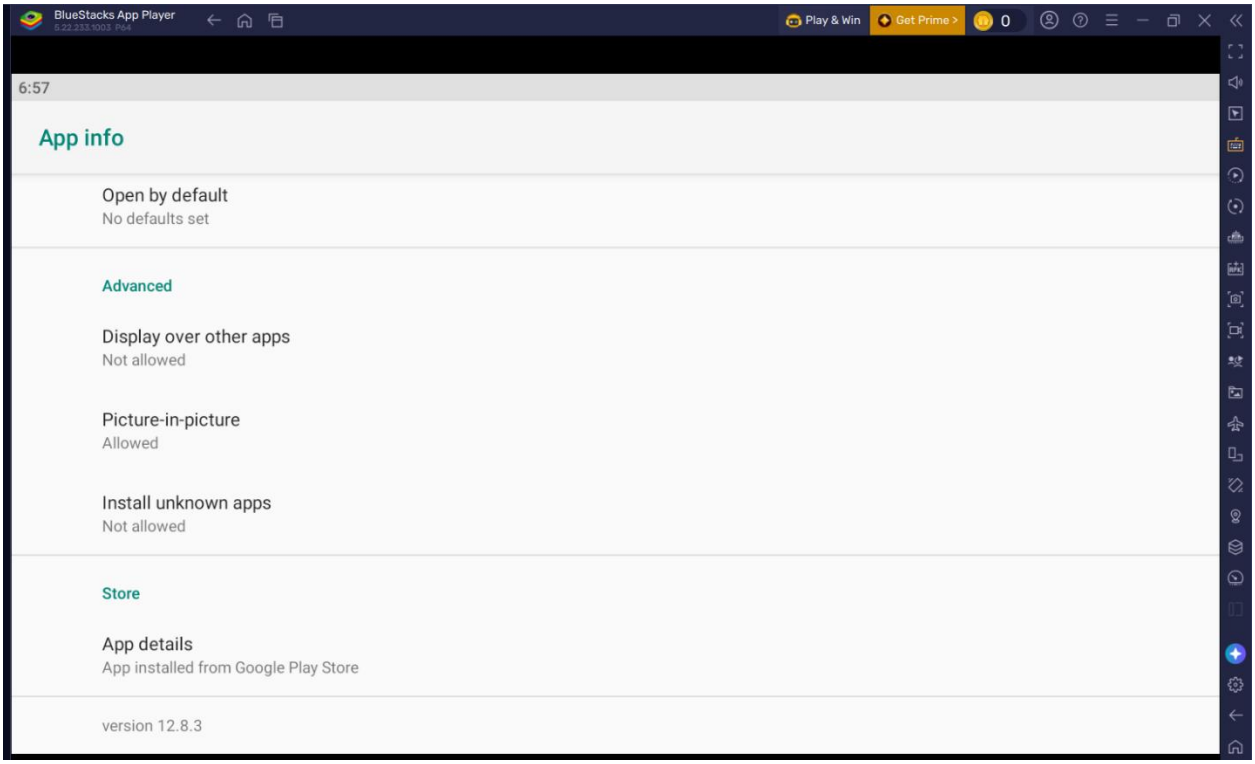


Figure 5: Telegram App Version

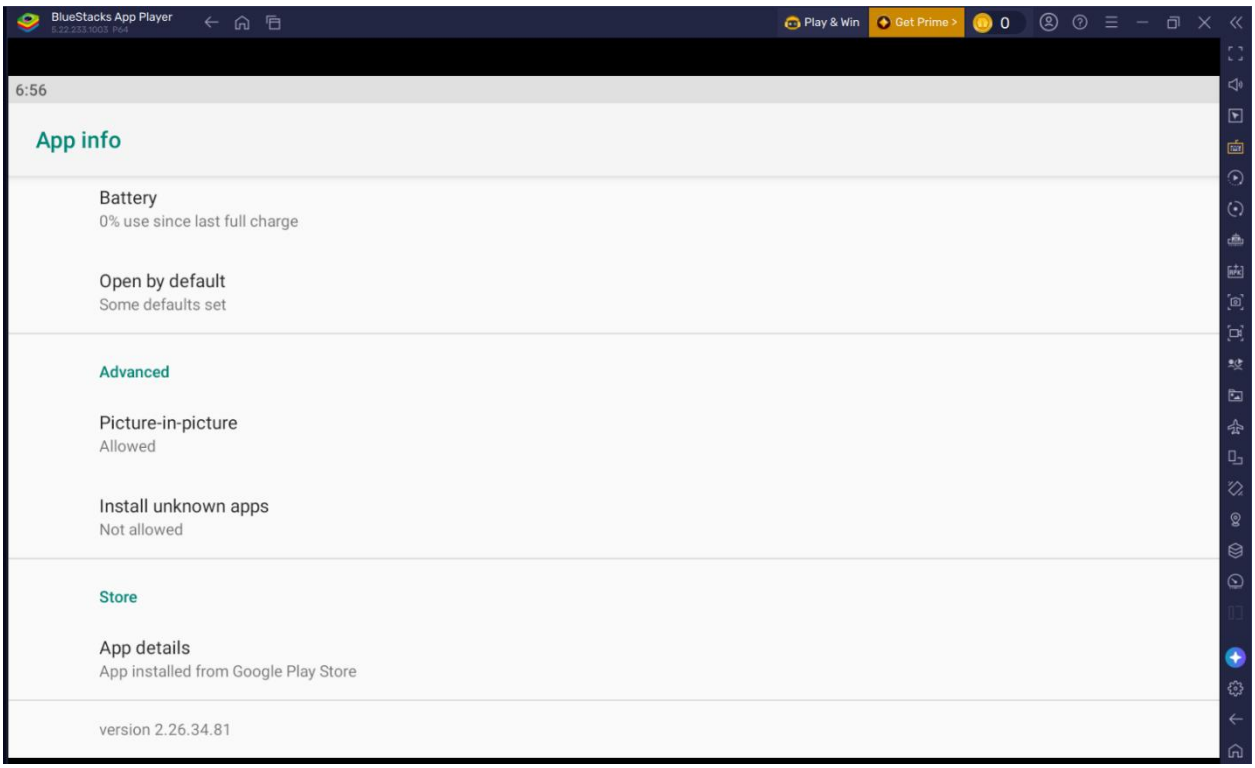


Figure 6: WhatsApp App Version

From Figure 1 to Figure 6 shows the versions of the applications and forensic tools used in the experiments.

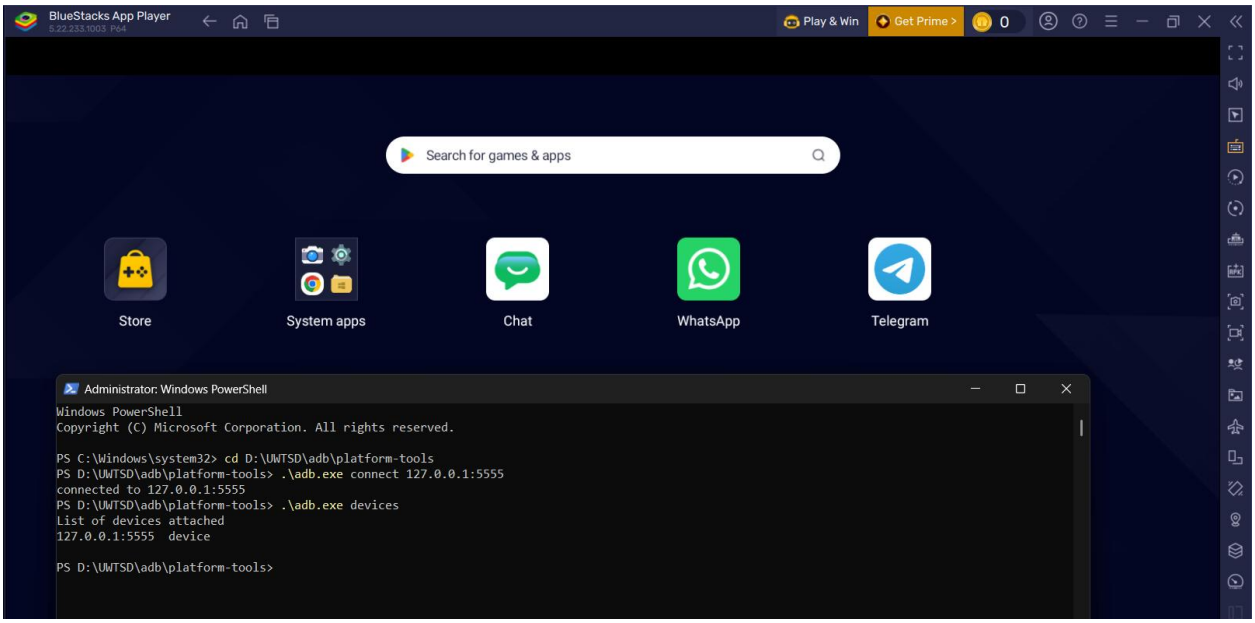


Figure 7: Experimental environment and ADB connection

Figure 7 demonstrates the experimental environment and confirms that ADB recognised the BlueStacks emulator at 127.0.0.1:5555. This connection allowed the researcher to collect accessible files from the emulator during the forensic experiments.

3.3 Controlled Evidence Generation

Controlled test data was created specifically for the research. No real user conversations were collected. Experiment 1 concentrated on controlled text-message deletion. Experiment 2 extended the design to text, image, document and voice evidence. The second experiment used staged deletion. All evidence was present for the BEFORE acquisition, then the image, document, voice and text items were deleted one at a time, with a new acquisition after each deletion. This design made the timing of filesystem changes easier to interpret.

Experiment	**Controlled content**	**Main comparison**
Experiment 1	Text messages and supporting media	Before deletion vs after deletion
Experiment 2	Text, image, document and voice	Before deletion → After image deletion, document deletion, voice deletion, text deletion

Table 3: Experimental Design

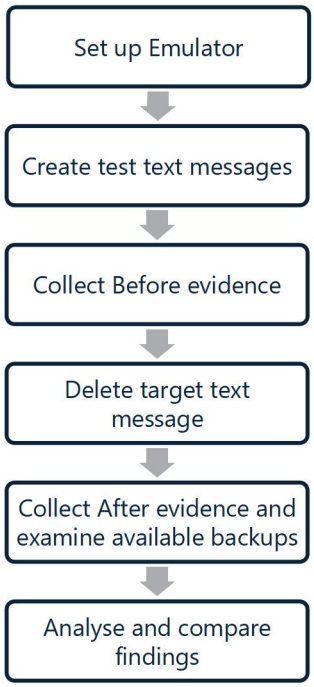


Figure 8: Experiment 1 text-message deletion workflow

Figure 8 illustrate the workflow used in Experiment 1. Five controlled text messages and one image were created in each application. The first text message was selected for deletion, while the remaining items were kept for comparison. Evidence was collected before and after deletion and examined using the available forensic methods. The following Figure 9, Figure 10, and Figure 11 demonstrate the controlled environment created for experiment 1 for Google Chat, Telegram, WhatsApp.

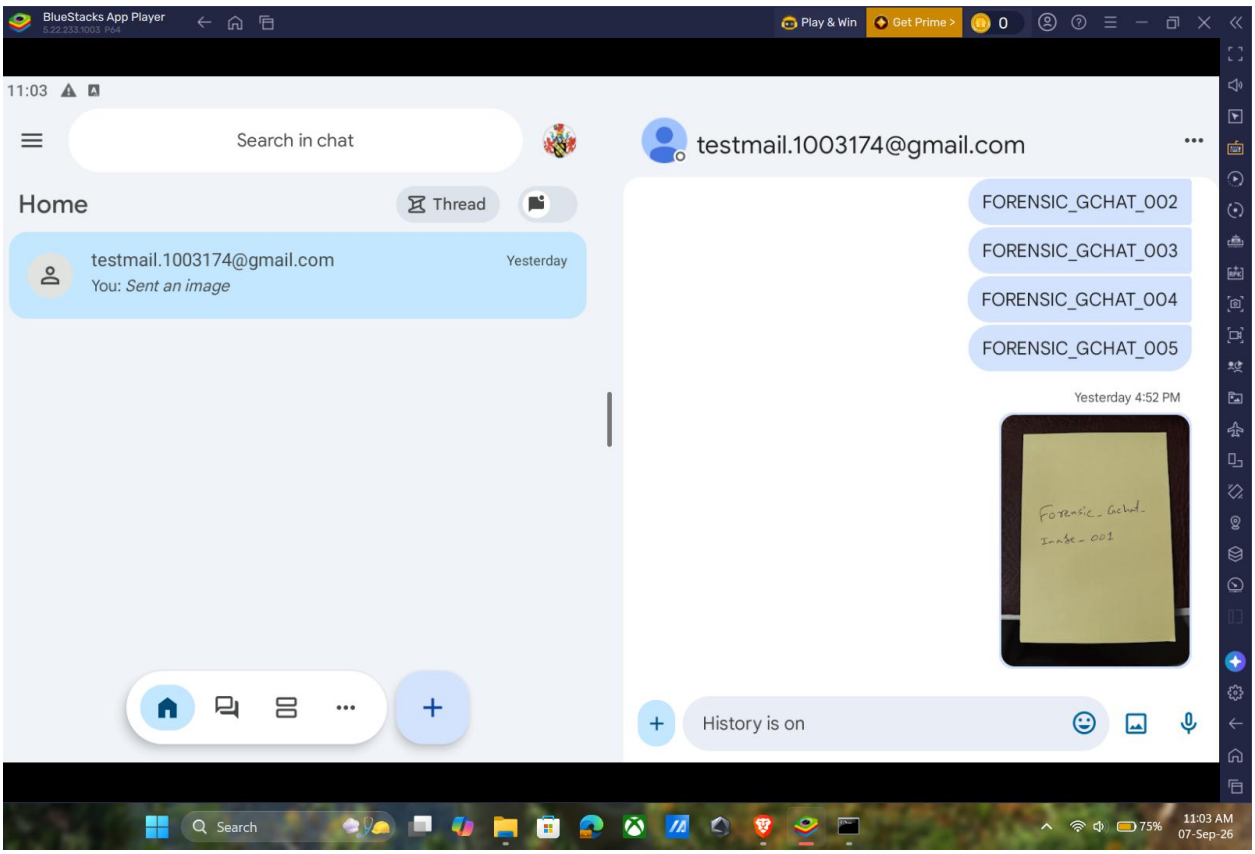


Figure 9: Controlled Google Chat App Environment in Experiment 1

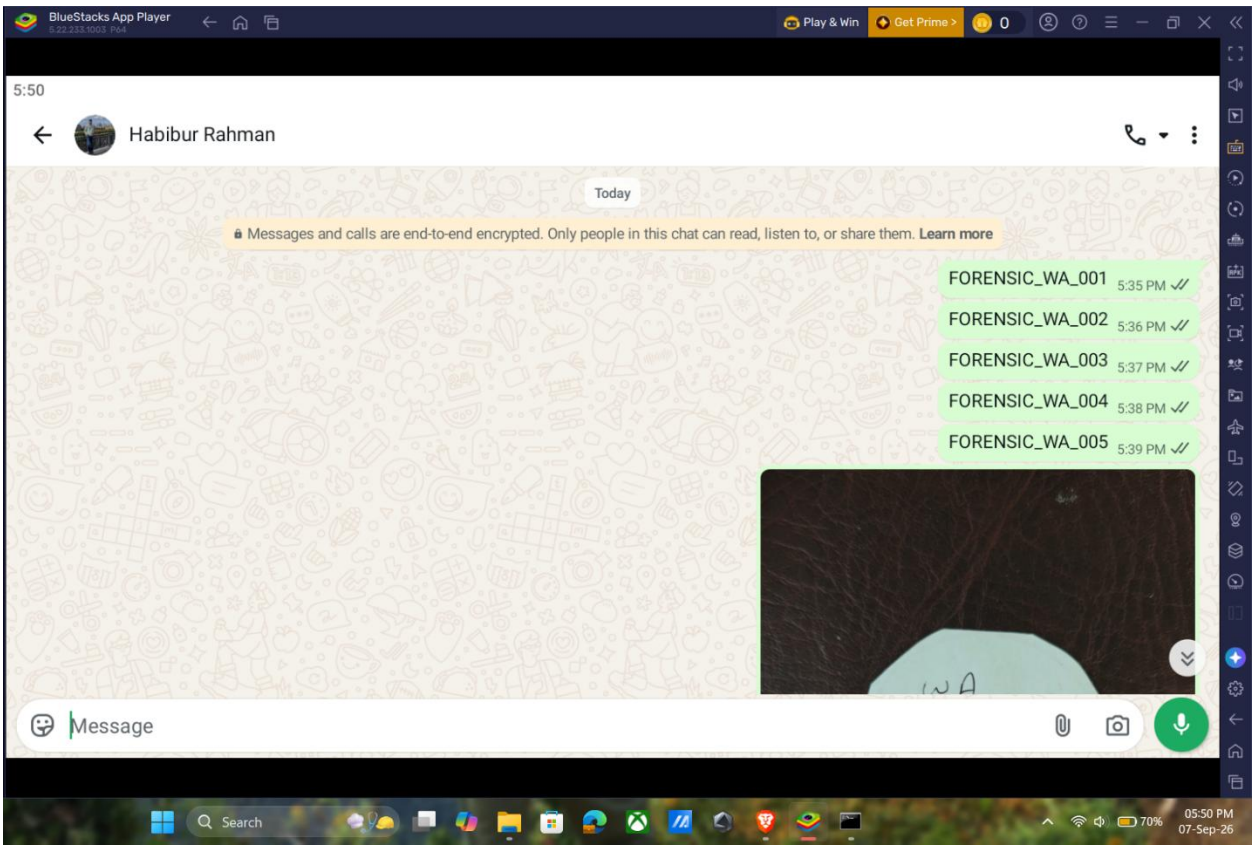


Figure 10: Controlled WhatsApp App Environment in Experiment 1

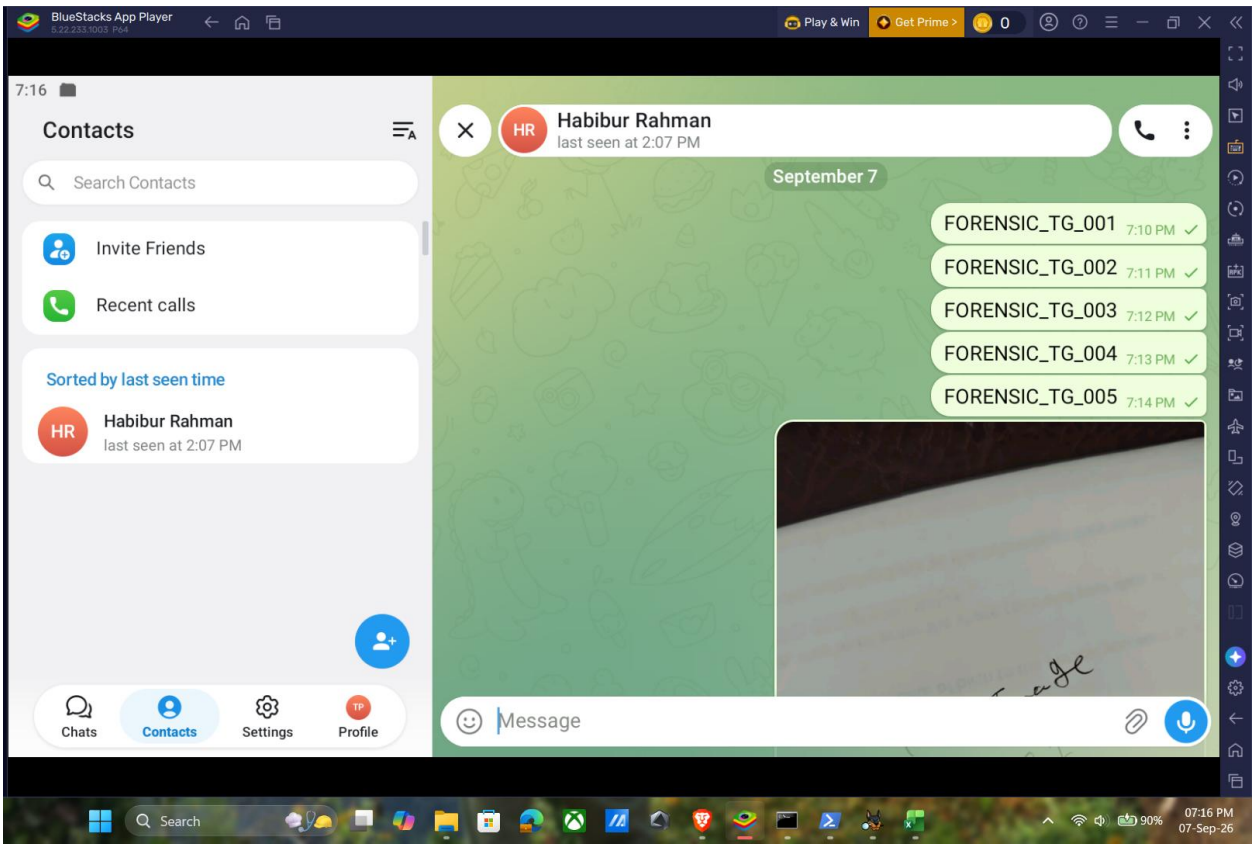


Figure 11: Controlled Telegram App Environment in Experiment 1

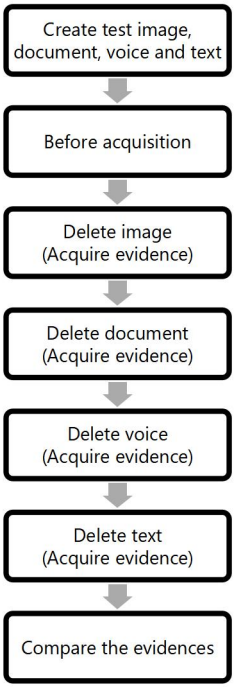


Figure 12: Experiment 2 staged deletion workflow

Figure 12 demonstrate how the staged deletion workflow was used in Experiment 2. A test image, document, voice message and text message were prepared for each application. Evidence was collected before deletion and again after each item was

deleted in the order of image, document, voice message and text message. This produced five acquisition stages for comparison. Following Figure 13, Figure 14, and Figure 15 visualize the controlled environment created for experiment 2 for Google Chat, Telegram, WhatsApp.

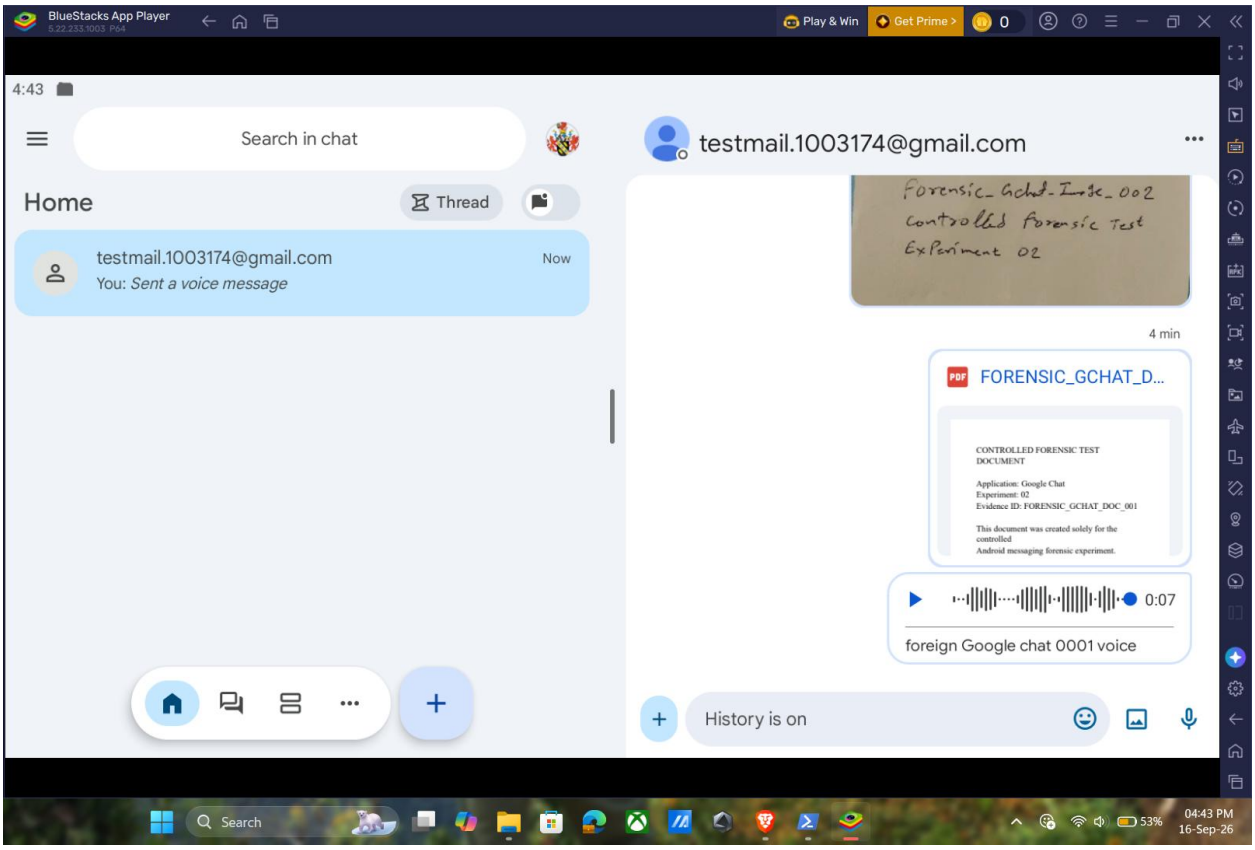


Figure 13: Controlled Google Chat App Environment in Experiment 2

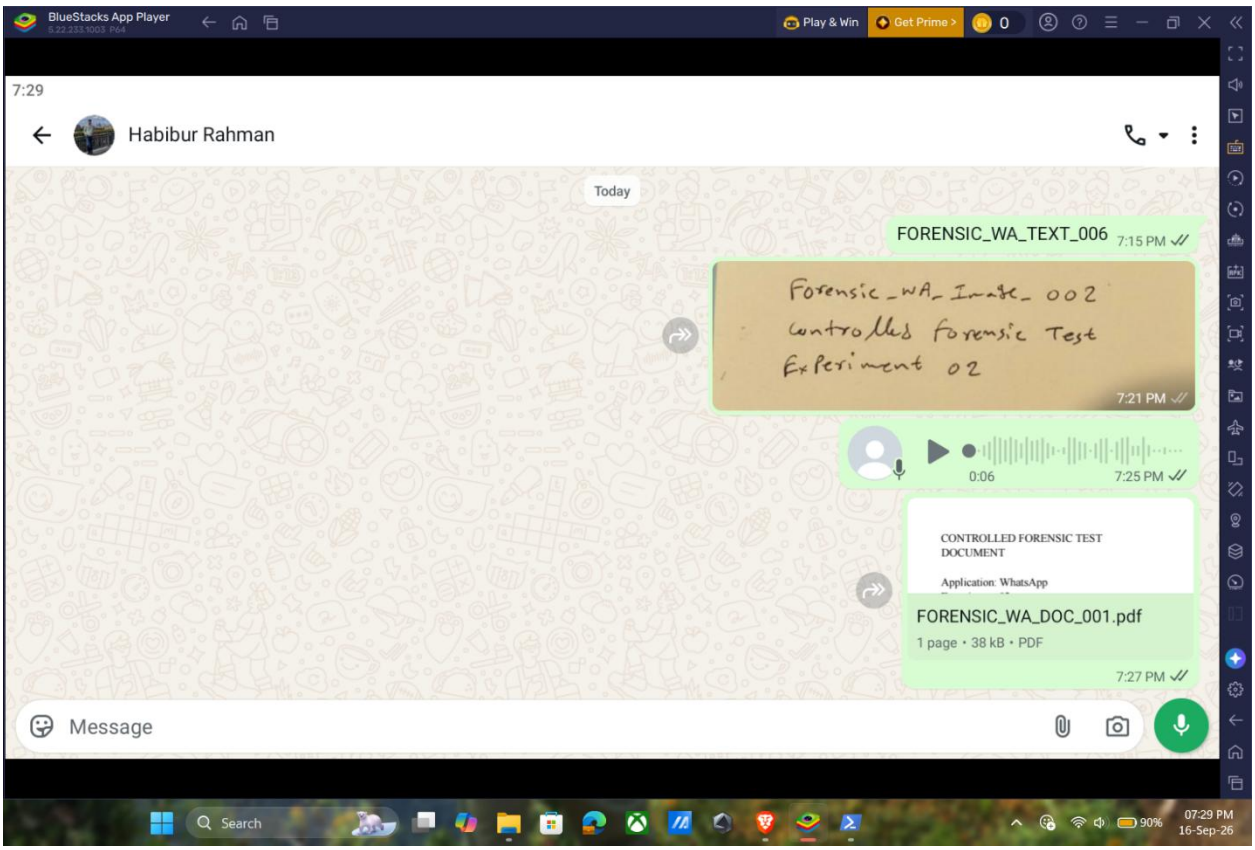


Figure 14: Controlled WhatsApp App Environment in Experiment 2

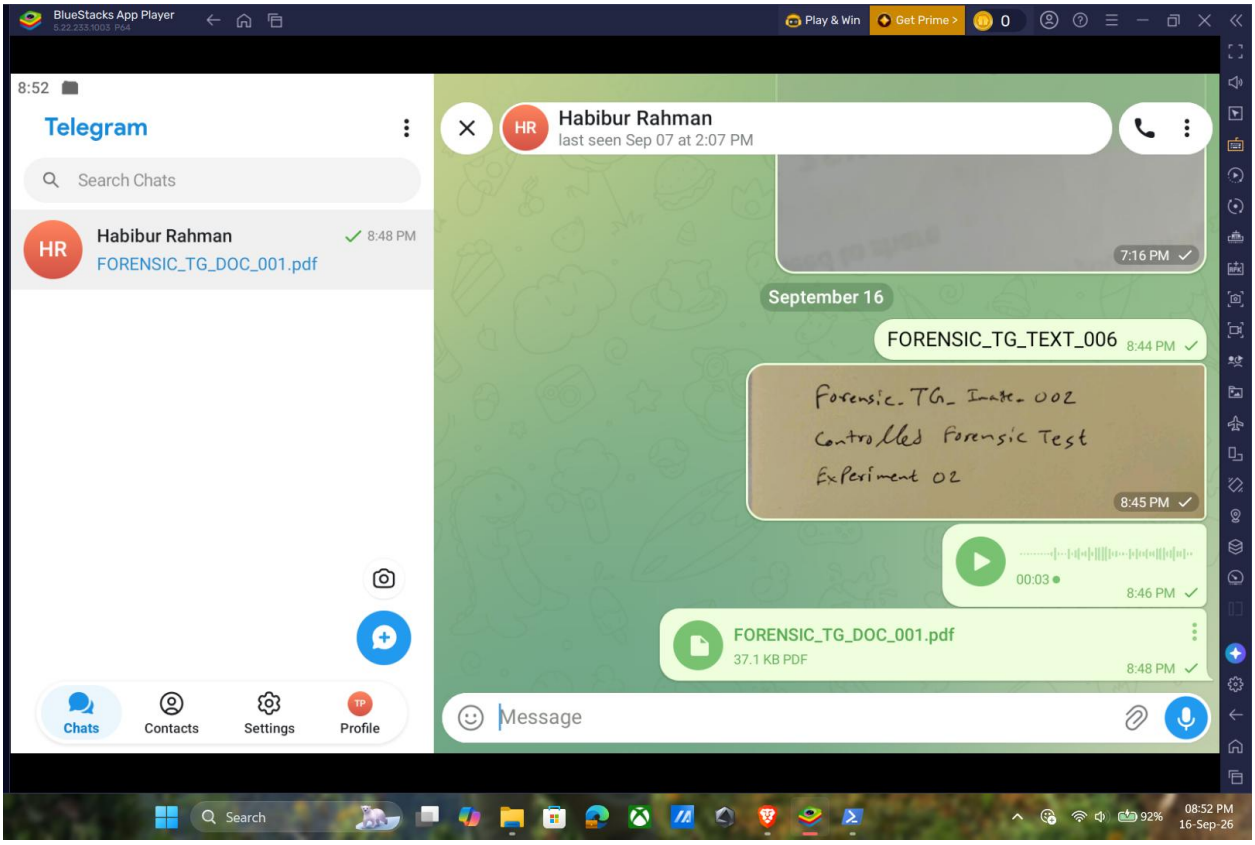


Figure 15: Controlled Telegram App Environment in Experiment 2

Collecting evidence after each deletion made it possible to identify when a file was added, removed or changed. However, a change observed after deletion was not automatically treated as being caused by that action, as normal application activity could also affect stored files.

3.4 Acquisition, Preservation and Analysis

At each required stage, accessible /sdcard/ data was acquired into a separate evidence folder. The evidence was not described as a physical image because the acquisition was a logical/shared-storage collection. SHA-256 manifests were generated so that corresponding files could be compared. Autopsy cases were created for the relevant Before and After acquisitions. Searches used the controlled evidence identifiers and, where necessary, the actual stored filename. Database analysis was performed only when the acquired material supported it.

```
PS C:\Windows\system32> Get-FileHash "D:\UWTSD\Android_Messaging_Forensics\WhatsApp\Database\Before_Deletion\msgstore.db.crypt14" -Algorithm SHA256

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          A43A986AABAE4ECD63C5204761740E05EE5D76E044B0EF19C32E0460ABEBFE43       D:\UWTSD\Android_Messaging_Forensics\WhatsApp\Database\Before_Deletion\msgstore.db.crypt14

PS C:\Windows\system32>
```

Figure 16: SHA-256 hash verification of the WhatsApp database before deletion

Figure 16 shows matching SHA-256 hash values for the preserved WhatsApp database files before deletion confirming that the file were identical at the time of hashing.

SHA-256 was used as the primary integrity and comparison mechanism. For each acquisition, relative file paths, sizes and hashes were recorded in manifests. Comparisons classified paths as removed, added or changed. A file was considered changed only when the corresponding relative path existed in both states but its recorded hash differed. This method is stronger than comparing file counts alone because identical counts can conceal substitutions, while increased counts do not identify which artifacts appeared.

Experiment 2 additionally preserved PowerShell transcripts, command-history exports, result CSVs and record-file hashes. The transcript was stopped before its final SHA-256 was calculated, ensuring that the hash represented the finalized file. PSReadLine history was treated as supplementary because it can contain commands from unrelated sessions. These controls improve reproducibility while acknowledging that a hash establishes the identity of a saved file at a point in time rather than proving that an acquisition was complete.

3.5 Ethics, Validity, Reliability, and limitations

The experiments used controlled test accounts and synthetic messages and media files. No real user conversations or human participants were involved. Evidence was collected only for the research and stored in separate folders for each application and acquisition stage.

Internal validity was strengthened through unique markers, preserved Before evidence, sequential Experiment 2 acquisitions and consistent hashing. Positive multimedia controls were particularly useful because image, document and verified voice files could be observed directly in shared storage. However, negative text searches in Before evidence limit any claim that deletion caused text disappearance. The correct interpretation is that the marker was not discoverable through the selected shared-storage search.

Reliability was supported by repeated use of the same broad workflow and by recording exact command outputs, hashes and Autopsy results. Experiment 2 improved reliability further by isolating each deletion stage. Nevertheless, the study remains a controlled case study using one emulator environment and specific application versions. Application updates, Android versions, account settings and acquisition privileges may produce different outcomes.

The principal limitation is incomplete private-storage access. Direct private database access was denied for the applications under the tested ADB permissions and run-as failed. Google Chat Experiment 1 obtained a readable backup-derived database, but comparable readable Before and After private databases were not available across all applications. WhatsApp encryption and Telegram's lack of a readable exported database further constrained database-level comparison.

The experiment used controlled accounts and synthetic forensic markers rather than third-party private conversations. Evidence was collected only for the research task and preserved in structured directories. Ethical reporting also requires avoiding exaggerated claims. The study therefore states that original deleted text was not recovered through the selected methods rather than claiming permanent deletion, and it separates deletion indicators from recovered content.

Chapter 4: Design Process

This chapter explains how the two experiments were planned and designed to examine digital evidence from Google Chat, WhatsApp and Telegram. It covers the selection of forensic tools, the organisation of evidence, and the steps planned for collecting and comparing data before and after deletion.

4.1 Design Objectives

The experiment was designed around three practical requirements: the activity had to be controlled, each acquisition had to remain separate, and the same general process had to be applied to Google Chat, WhatsApp and Telegram. The design also needed to preserve evidence integrity while remaining realistic for a non-rooted Android emulation environment.

Design requirement	**Implementation**
Known evidence	Unique Forensic identifiers were used for controlled items.
Known deletion point	Deletion actions and times were recorded.
Evidence separation	Before and later acquisitions were stored in different folders.
Integrity	SHA-256 manifests were generated.
Cross-app comparison	Equivalent evidence types were tested across the three apps.

Table 4: Design requirements and controls

Experiment 1 was designed as a simple controlled text-deletion test. Five text messages and one image were created for each application, with the first text marker selected for deletion. This provided retained messages as internal controls while limiting the deletion event to one target. The design exposed an important limitation: shared-storage keyword searches could be negative even before deletion. Consequently, Experiment 2 was designed to provide stronger observable controls through multimedia files and to examine content types that are more likely to leave shared-storage artifacts.

Experiment 2 therefore required four evidence classes which are text, image, document and voice/audio with a sequential deletion schedule. All evidence had to be present in the Before state. Each deletion then had to be followed immediately by a new acquisition and manifest comparison. This made the experiment more diagnostic because it could identify the interval in which a file-level change first appeared.

4.2 Platform and tool selection

BlueStacks was retained as the Android platform because it provided a controlled and repeatable environment in which the three applications could be installed and accessed through ADB. Using one emulator reduced hardware variability and allowed the same host-side directory structure and acquisition commands to be reused. The trade-off is that emulator behavior is not automatically representative of every physical Android handset, so the design explicitly limits generalization.

ADB was selected as the principal acquisition interface because it exposes both successful and unsuccessful access attempts clearly. Shared storage could be copied with adb pull, while private directory checks documented permission boundaries. PowerShell provided reproducible command execution, hashing and logging. Autopsy was selected for logical-file examination and keyword searching. DB Browser for SQLite was included as a conditional tool rather than a mandatory stage: it would be used only if a readable SQLite database was acquired.

Database analysis was included only where readable data was available. This allowed each application to be examined using the evidence that could be acquired without assuming that the same level of database access would be possible across all three applications.

4.3 Evidence naming and folder structure

Unique marker naming was used to minimize ambiguity. Experiment 1 markers followed application prefixes such as GCHAT, WA and TG with numeric suffixes. Google Chat used FORENSIC_GCHAT_001 through _005, WhatsApp used FORENSIC_WA_001 through _005, and Telegram used FORENSIC_TG_001 through _005. Experiment 2 added content-type identifiers, for example FORENSIC_TG_IMAGE_002.jpg and FORENSIC_WA_DOC_001.pdf. This scheme allowed a filename or keyword hit to be associated immediately with the correct application and experiment. As already shown in Figure 9, Figure 10, Figure 11 for Experiment 1 and in Figure 13, Figure 14, Figure 15 for experiment 2

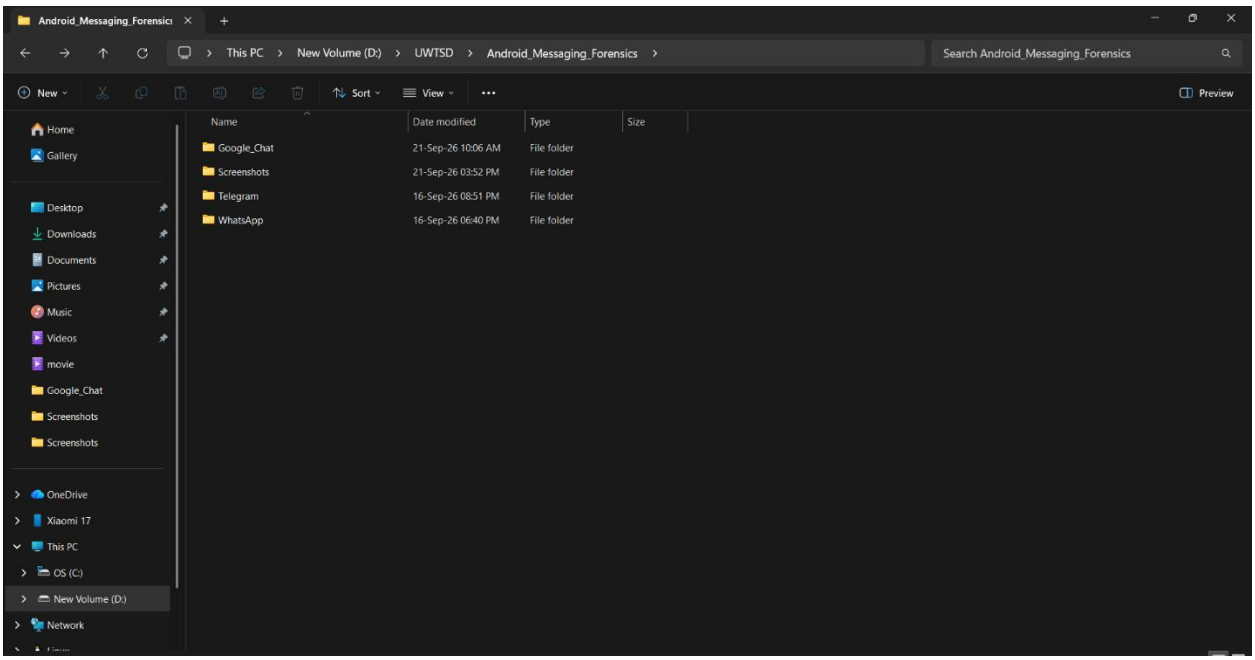


Figure 17: Android_Messaging_Forensics project folder

The following Figure 17 shows that the location and the name of the folder is. Controlled identifiers made it possible to search for specific test items without relying on ordinary conversation text. Experiment 1 remained in the original application working folders because it was completed first. Experiment 2 was later placed inside a dedicated Experiment_02_Multimedia_Deletion folder. The original working structure was preserved because logs, commands and analysis records refer to those paths.

The evidence directory design used application-specific roots and separate Experiment 2 folders. Within Experiment 2, Before, After_Image_Deletion, After_Document_Deletion, After_Voice_Deletion and After_Text_Deletion were kept separate. Additional folders were reserved for hashes, screenshots, test files, logs, database-related material and Autopsy cases.

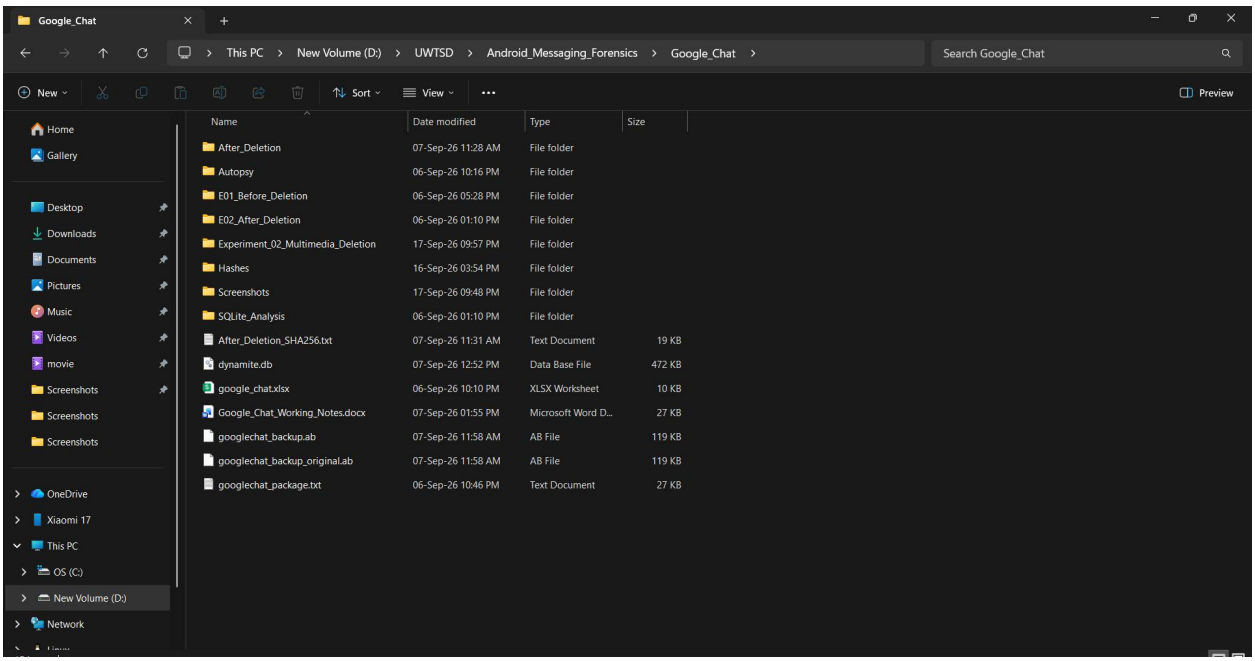


Figure 18: Google Chat Folder Environment for Experiment 1

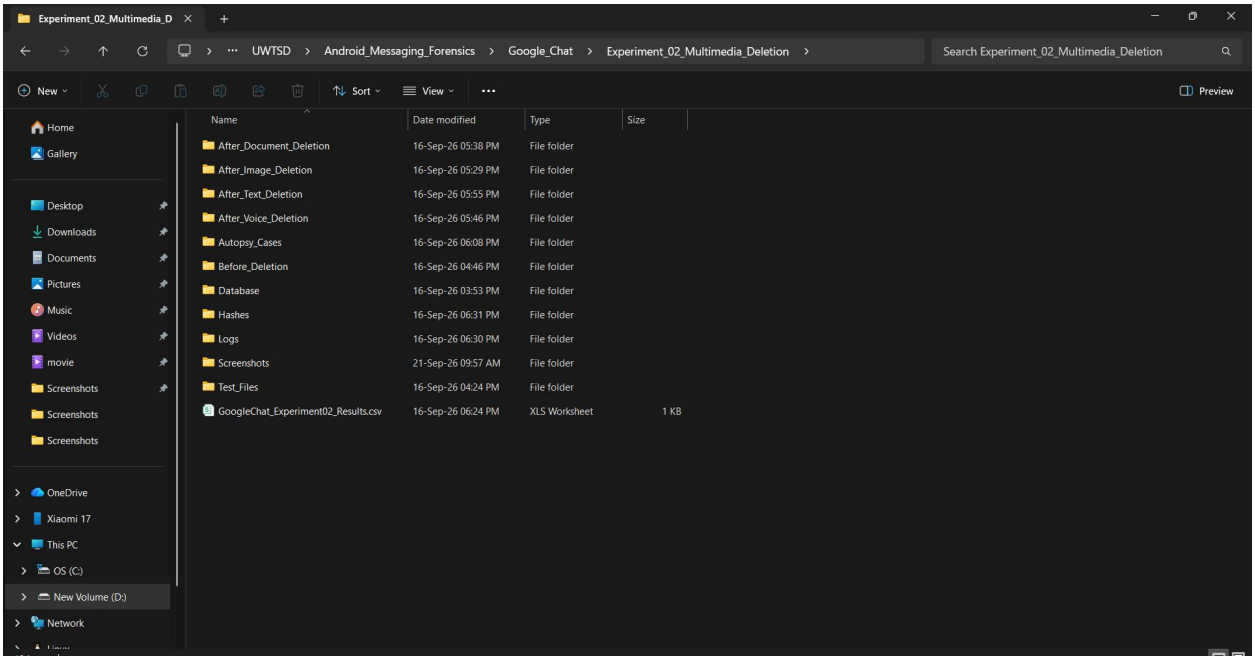


Figure 19: Google Chat Folder Environment for Experiment 2

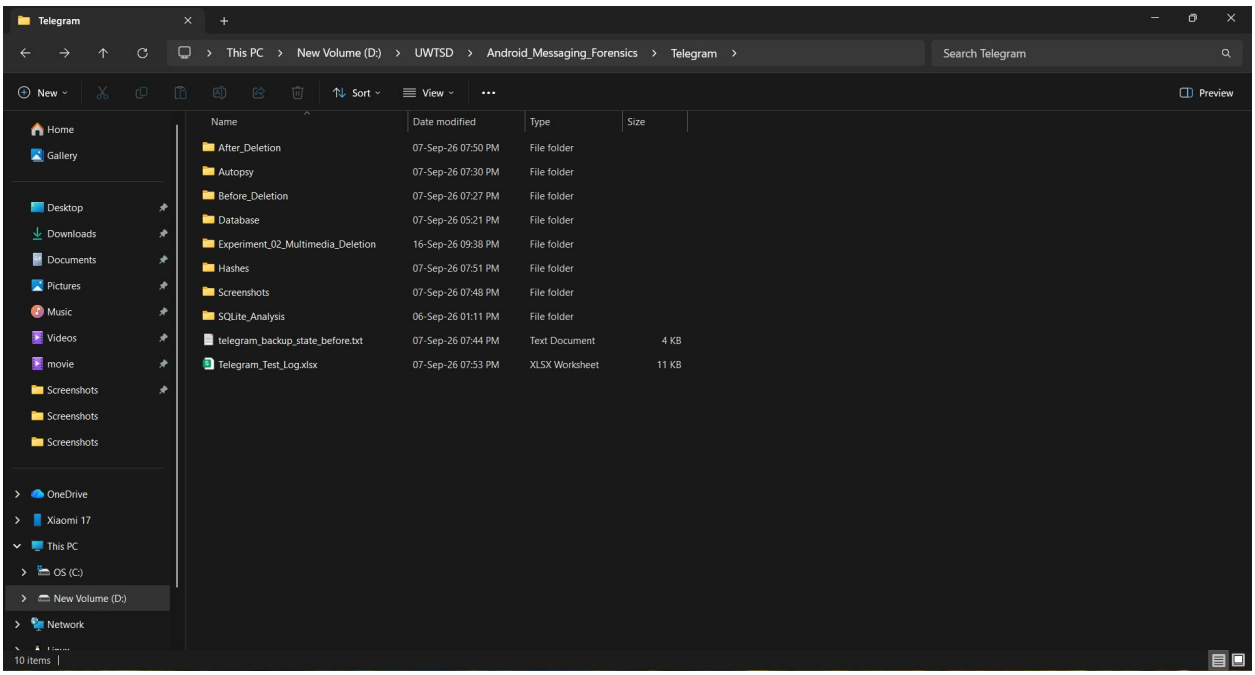


Figure 20: Telegram Folder Environment for Experiment 1

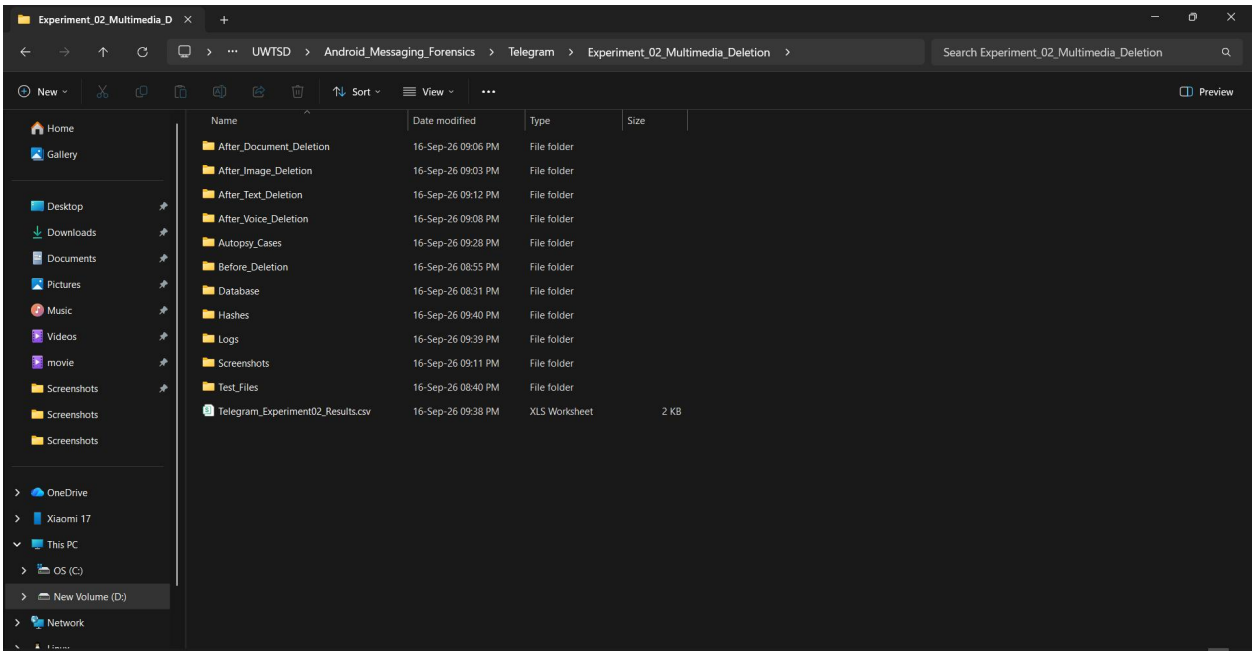


Figure 21: Telegram Folder Environment for Experiment 2

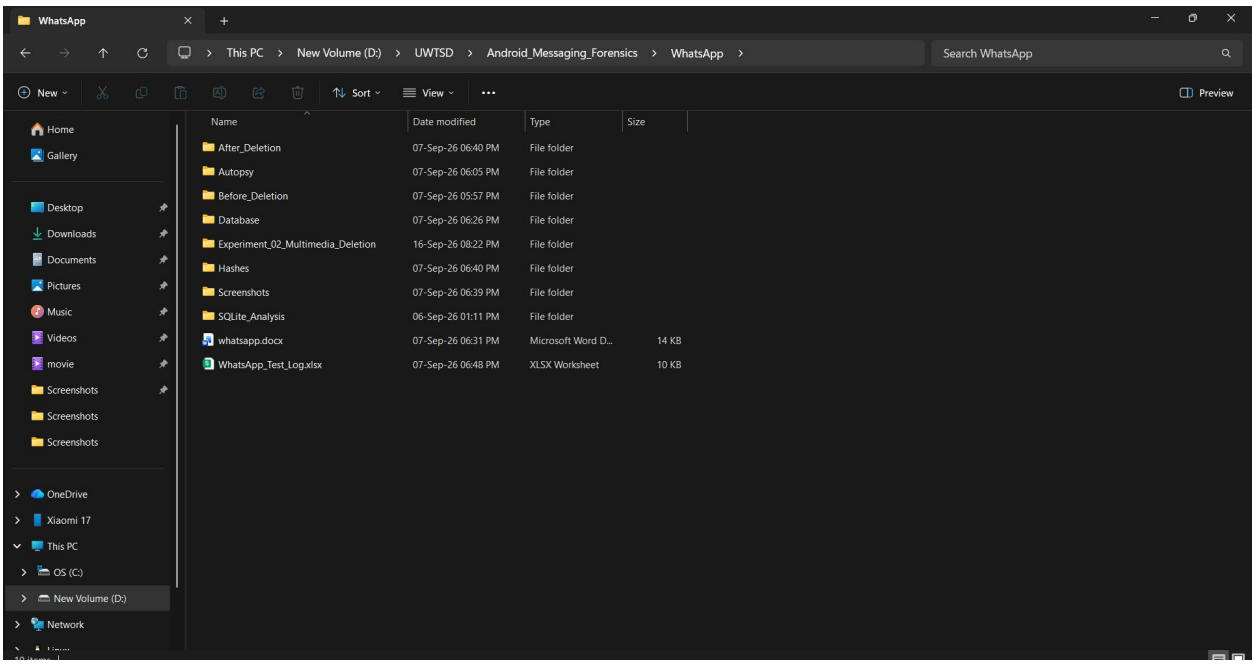


Figure 22: WhatsApp Folder Environment for Experiment 1

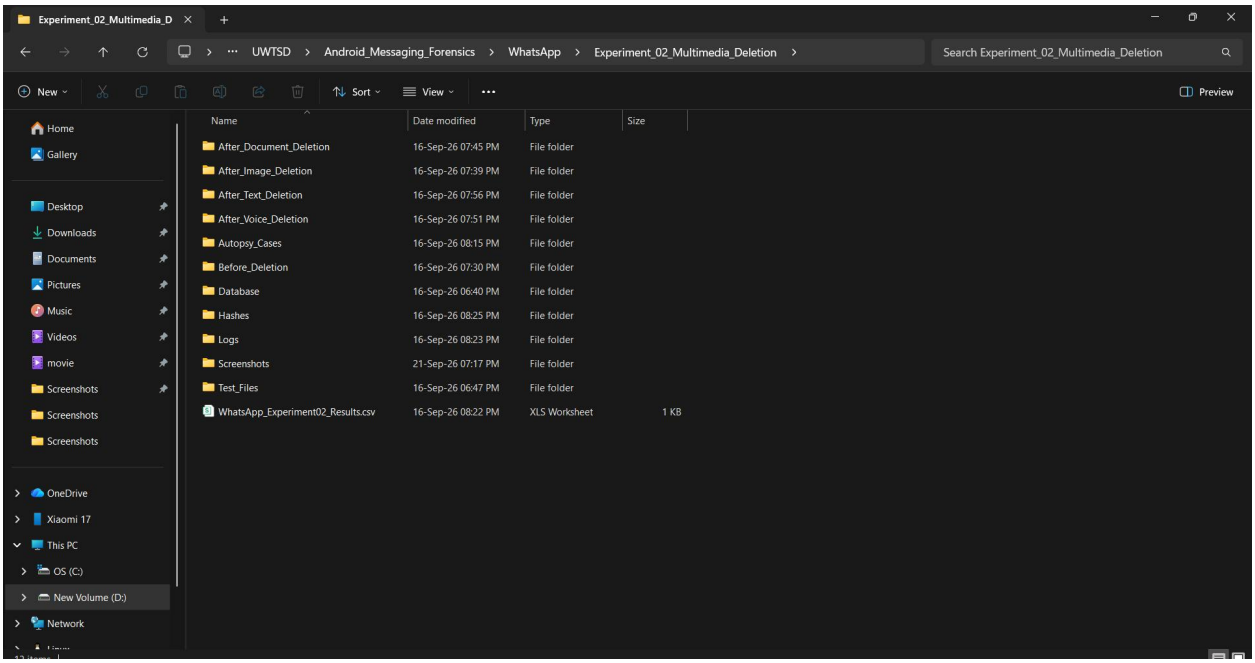


Figure 23: WhatsApp Folder Environment for Experiment 2

From Figure 18 to Figure 23, illustrates how the evidence folders were organised for each application and experiment.

4.4 Experiment 1 design

Experiment 1 established the initial text-deletion workflow. Controlled messages were created, a Before acquisition was preserved, one selected text message was deleted, and an After acquisition was collected. The intention was to look for message content,

deletion indicators and file-level differences. The same broad idea was used for all three applications, although the evidence available from each application differed.

4.5 Experiment 2 design

Experiment 2 was designed after the first experiment to give a clearer view of multimedia deletion. Instead of deleting several evidence types together, one item was deleted at a time. A fresh shared-storage acquisition followed each deletion. This provided a temporal sequence and reduced the risk of wrongly linking a file-system change to the wrong deletion event.

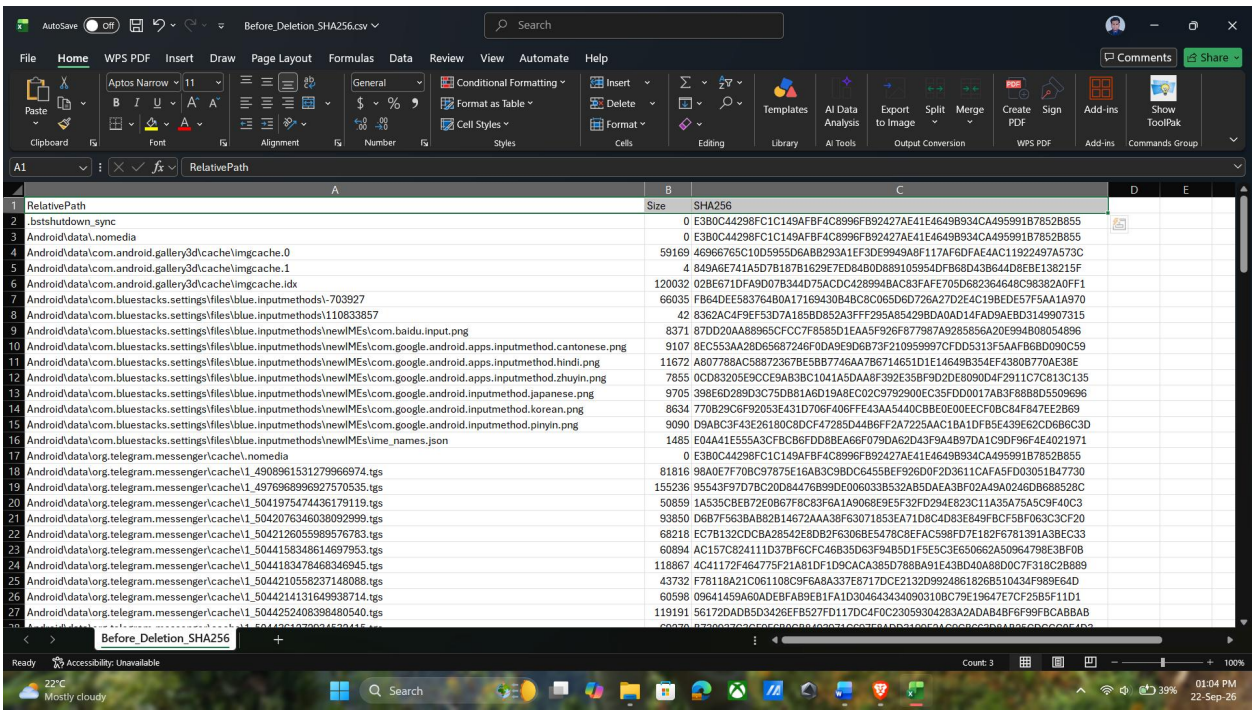
Stage	**State of controlled evidence**
Before	Text, image, document and voice present
After Image Deletion	Image deleted in app, other controlled items still present
After Document Deletion	Image and document deleted
After Voice Deletion	Image, document and voice deleted
After Text Deletion (Final)	All four controlled items deleted in app

Table 5: Experiment 2 staged evidence states

Table 5 shows the evidence state at each stage of Experiment 2.

4.6 Hashing and Comparison Design

SHA-256 was selected for integrity documentation and file comparison. Each acquisition was converted into a manifest of relative paths and hashes. Comparisons produced three categories: removed paths, added paths and changed paths. A direct Before then After comparison was retained in addition to sequential comparisons. The design explicitly avoids equating additions with deletion artifacts. Exact file checks were added where a controlled artifact could be identified confidently. Targeted hash checks were also planned for controlled files that could be identified in the acquired evidence. These checks supported the wider manifest comparison by confirming whether specific files remained unchanged between acquisition stages.



RelativePath	Size	SHA256
.bstshutdown_sync	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
Android\data\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
Android\data\com.android.gallery3d\cache\imgcache.0	59169	46966765C10D5955D6ABB293A1EF3DE9949A8F117AF6DFAE4AC11922497A573C
Android\data\com.android.gallery3d\cache\imgcache.1	4	849A6E741A5D7B187B1629E7ED84B0D889105954DFB68D43B644D8EBE138215F
Android\data\com.android.gallery3d\cache\imgcache.idx	120032	02BE671DFA9D07B344D75ACDC428994BAC83FAFE705D682364648C98382A0FF1
Android\data\com.bluestacks.settings\files\blue.inputmethods\-703927	66035	FB64DEE583764B0A17169430B4BC8C065D6D726A27D2E4C19BEDE57F5AA1A970
Android\data\com.bluestacks.settings\files\blue.inputmethods\110833857	42	8362AC4F9EF53D7A185BD852A3FFF295A85429BDA0AD14FAD9AEBD3149907315
Android\data\com.bluestacks.settings\files\blue.inputmethods\newlMEs\com.baidu.input.png	8371	87DD20AA88965CFCC7F8585D1EAA5F926F877987A9285856A20E994B08054896
Android\data\com.bluestacks.settings\files\blue.inputmethods\newlMEs\com.google.android.apps.inputmethod.cantonese.png	9107	8EC553AA28D65687246F0DA9E9D6B73F2109559997CFDD5313F5AAFB6BD090C59
Android\data\com.bluestacks.settings\files\blue.inputmethods\newlMEs\com.google.android.apps.inputmethod.hindi.png	11672	A807788AC58872367BE5BB7746AA7B6714651D1E14649B354EF4380B770AE38E
Android\data\com.bluestacks.settings\files\blue.inputmethods\newlMEs\com.google.android.apps.inputmethod.zhuyin.png	7855	0CD83205E9CCE9AB3BC1041A5DAA8F392E35BF9D2DE8090D4F2911C7C813C135
Android\data\com.bluestacks.settings\files\blue.inputmethods\newlMEs\com.google.android.inputmethod.japanese.png	9705	398E6D289D3C75DB81A6D19A8EC02C9792900EC35FDD0017AB3F88B8D5509696
Android\data\com.bluestacks.settings\files\blue.inputmethods\newlMEs\com.google.android.inputmethod.korean.png	8634	770B29C6F92053E431D706F406FFE43AA5440CBBE0E00EECF0BC84F847EE2B69
Android\data\com.bluestacks.settings\files\blue.inputmethods\newlMEs\com.google.android.inputmethod.pinyin.png	9090	D9ABC3F43E26180C8DCF47285D44B6FF2A7225AAC1BA1DFB5E439E62CD6B6C3D
Android\data\com.bluestacks.settings\files\blue.inputmethods\newlMEs\ime_names.json	1485	E04A41E555A3CFBCB6FDD8BEA66F079DA62D43F9A4B97DA1C9DF96F4E4021971
Android\data\org.telegram.messenger\cache\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
Android\data\org.telegram.messenger\cache\1_4908961531279966974.tgs	81816	98A0E7F70BC97875E16AB3C9BDC6455BEF926D0F2D3611CAFA5FD03051B47730
Android\data\org.telegram.messenger\cache\1_4976968996927570535.tgs	155236	95543F97D7BC20D84476B99DE006033B532AB5DAEA3BF02A49A0246DB688528C
Android\data\org.telegram.messenger\cache\1_5041975474436179119.tgs	50859	1A535CBEB72E0B67F8C83F6A1A9068E9E5F32FD294E823C11A35A75A5C9F40C3
Android\data\org.telegram.messenger\cache\1_5042076346038092999.tgs	93850	D6B7F563BAB82B14672AAA38F63071853EA71D8C4D83E849FBCF5BF063C3CF20
Android\data\org.telegram.messenger\cache\1_5042126055989576783.tgs	68218	EC7B132CDCBA28542E8DB2F6306BE5478C8EFAC598FD7E182F6781391A3BEC33
Android\data\org.telegram.messenger\cache\1_5044158348614697953.tgs	60894	AC157C824111D37BF6CFC46B35D63F94B5D1F5E5C3E650662A50964798E3BF0B
Android\data\org.telegram.messenger\cache\1_5044183478468346945.tgs	118867	4C41172F464775F21A81DF1D9CACA385D788BA91E43BD40A88D0C7F318C2B889
Android\data\org.telegram.messenger\cache\1_5044210558237148088.tgs	43732	F78118A21C061108C9F6A8A337E8717DCE2132D9924861826B510434F989E64D
Android\data\org.telegram.messenger\cache\1_5044214131649938714.tgs	60598	09641459A60ADEBFAB9EB1FA1D304643434090310BC79E19647E7CF25B5F11D1
Android\data\org.telegram.messenger\cache\1_5044252408398480540.tgs	119191	56172DADB5D3426EFB527FD117DC4F0C23059304283A2ADAB4BF6F99FBCABBAB

Figure 24: Hash (SHA-256) View

Figure 24 shows an example of a SHA-256 manifest used for file comparison.

4.7 Autopsy and database Analysis Design

The analysis used two complementary views. First, hash manifests were compared to identify files that were added, removed or changed. Second, Autopsy was used to search the acquired data for the controlled identifiers or known filenames. Where a readable database existed, SQLite examination was used to interpret records that were not visible as ordinary shared-storage files. This combination avoided relying on only one forensic tool. The same ingest modules were selected for Experiment 2 across applications to reduce analysis variability. Searches used exact markers and, where necessary, exact application-generated filenames. Positive results were recorded by count and location; negative results were recorded without being converted into claims of permanent absence.

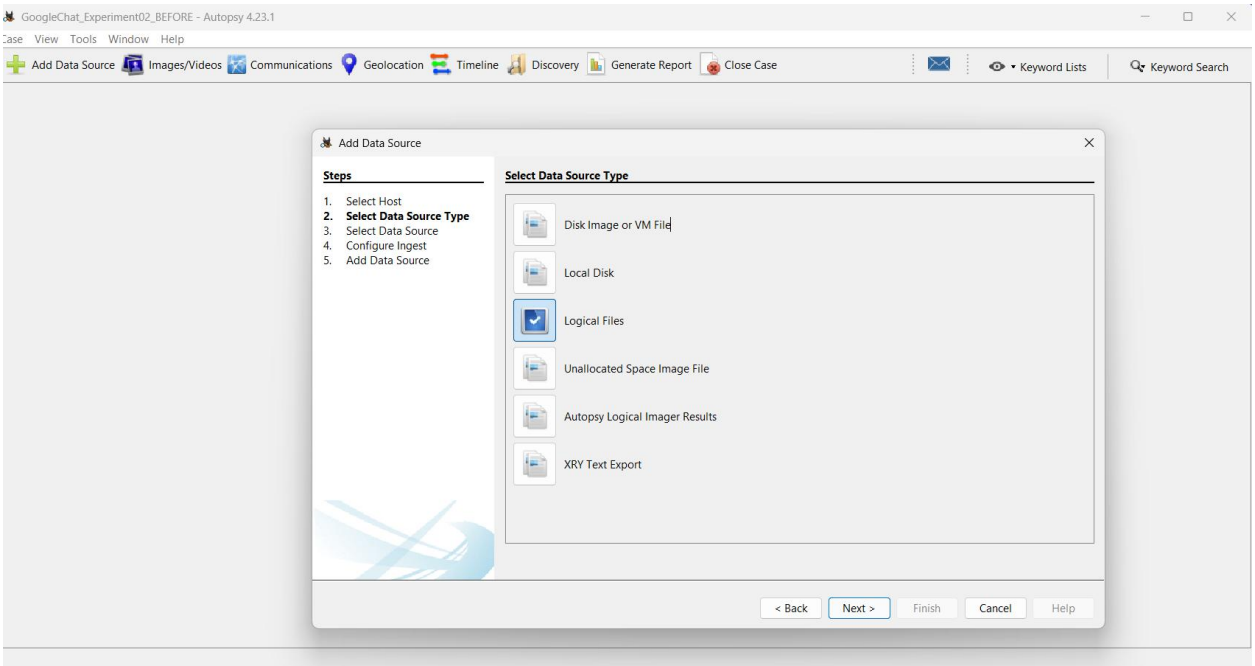


Figure 25: Autopsy Selection of Data Source Type

Figure 25 shows the data source type selected for the Autopsy examination.

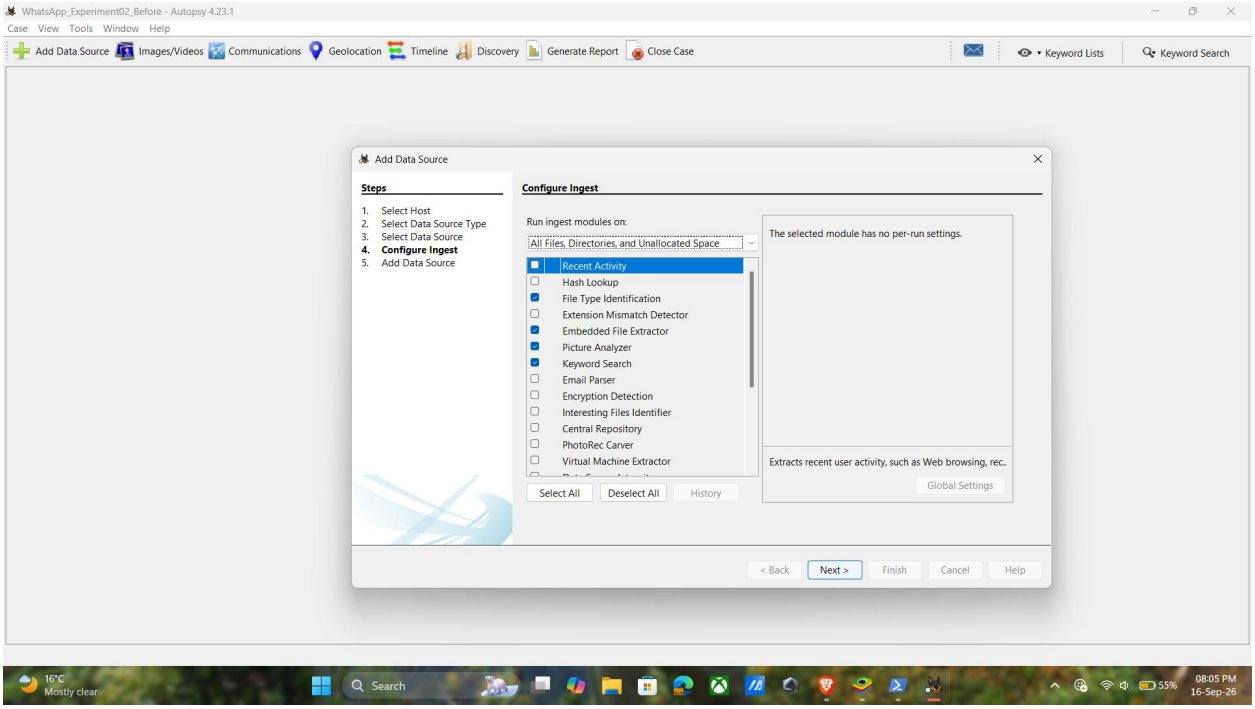


Figure 26: Autopsy Configuration Ingest 1

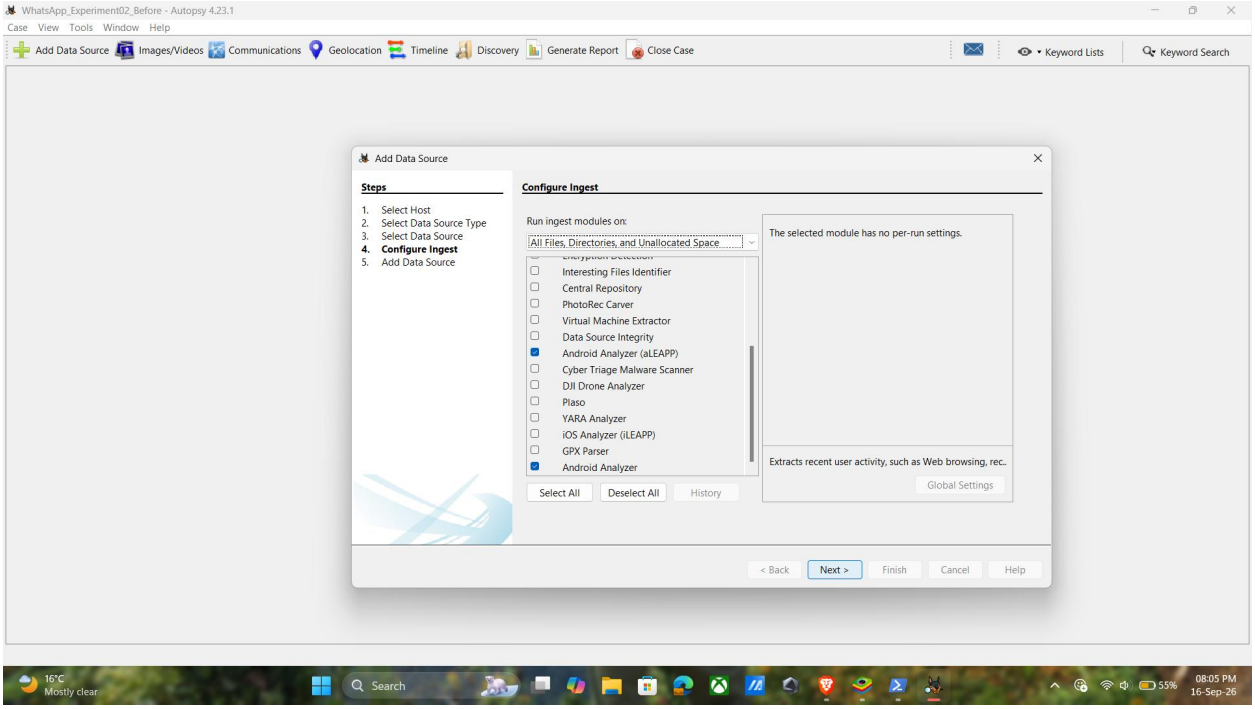


Figure 27: Autopsy Configuration Ingest 2

After seeing Figure 26 and Figure 27, show the ingest settings selected for the examination.

Google Chat Experiment 1 allowed SQLite examination of topic_messages. The tombstone record was documented using raw fields such as delete_time, is_tombstone and tombstone_metadata. WhatsApp's .crypt14 was not opened as if it were plaintext SQLite because no key had been acquired. Telegram's Backup Manager success was documented without claiming that a readable database had been produced.

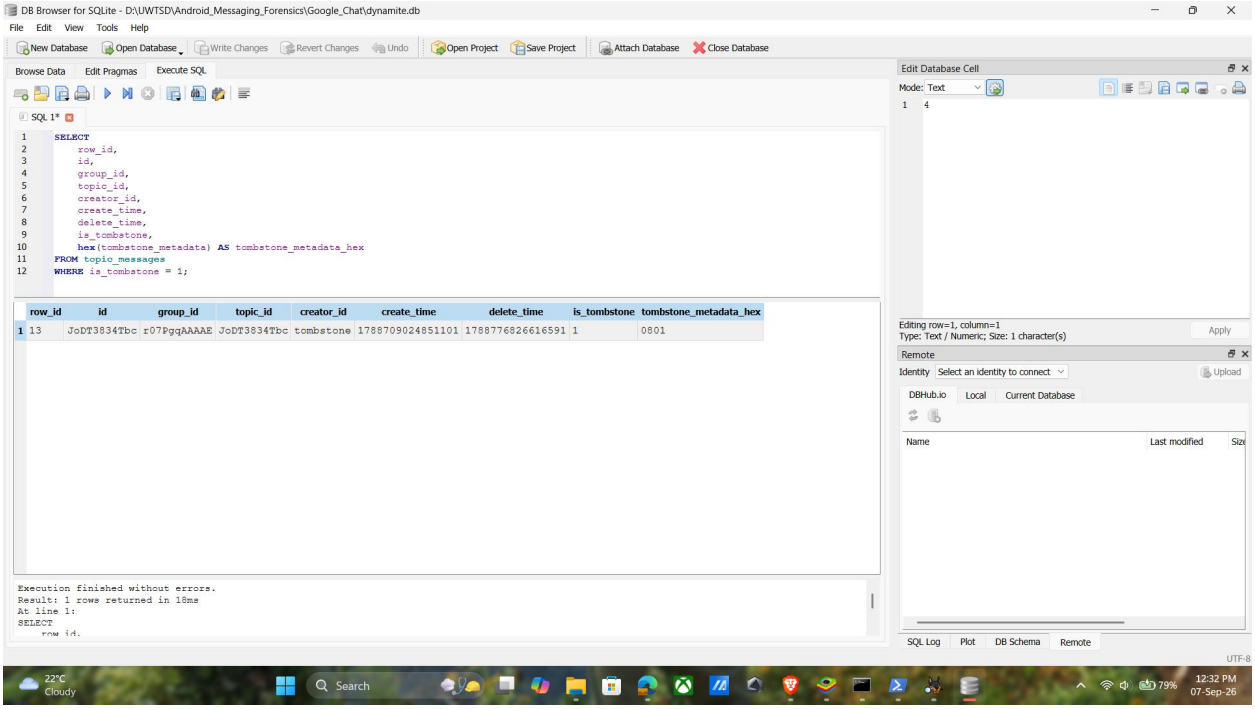


Figure 28: DB Browser for SQLite View

Figure 28 shows the DB Browser for SQLite interface used for readable database examination.

This design separates tool capability from evidence availability. DB Browser is useful only when the input is a readable database. Autopsy can index accessible files but cannot search data that was never acquired. ADB can copy accessible shared storage but cannot be described as a physical acquisition. These boundaries became part of the experiment design rather than after-the-fact caveats.

4.8 Design evaluation and transition to implementation

The revised design improved on Experiment 1 in three ways. First, multimedia evidence provided stronger positive controls than text markers that were not visible in shared storage before deletion. Second, sequential acquisition made the timing of file-level changes observable. Third, expanded logging and hashing created a clearer audit trail.

The design still retained limitations. It did not obtain full private-storage access, did not create matched readable Before and After private databases for all applications, and did not use repeated independent emulator instances. These limitations were accepted because the research objective was to compare evidence available through the selected methods rather than to exhaust every forensic acquisition technique.

The implementation chapter applies this design to the actual recorded evidence. It reports exact file counts, hashes, Autopsy results and access outcomes, while keeping the distinction between observation and interpretation established during the design phase.

Chapter 5: Implementation

This chapter describes how the two experiments were carried out using the BlueStacks Android emulator. It explains the steps taken to collect, preserve and examine evidence from Google Chat, WhatsApp and Telegram using ADB, Autopsy, DB Browser for SQLite and SHA-256 hashing.

5.1 Environment configuration and evidence handling

The implementation began by connecting ADB to the BlueStacks emulator at 127.0.0.1:5555 and confirming that the device was listed. The recorded application packages were com.google.android.apps.dynamite for Google Chat, com.whatsapp for WhatsApp and org.telegram.messenger for Telegram. Application-specific evidence directories under D:\UWTSD\Android_Messaging_Forensics were used to keep the three platforms separate.

For each experiment, accessible shared storage was acquired using adb pull /sdcard/ to a stage-specific destination. PowerShell was used to generate SHA-256 manifests and compare acquisitions. Experiment 2 additionally used Start-Transcript, CSV experiment logs and command-history preservation. The transcript was not stopped until all planned commands were complete, after which the finalized transcript was hashed separately.

Autopsy 4.23.1 was used with Logical Files data sources. In Experiment 2, the ingest configuration was kept consistent: File Type Identification, Embedded File Extractor, Picture Analyzer, Keyword Search, Android Analyzer (aLEAPP) and Android Analyzer. DB Browser for SQLite was used only for the readable Google Chat Experiment 1 database.

5.2 Google Chat Experiment 1 implementation

For Google Chat, the shared-storage acquisition did not reveal the controlled deleted text through Autopsy keyword searching. A backup-derived readable database, dynamite.db, was later examined in DB Browser for SQLite. The following Table 6 shows the outcome

Application	**Experiment 1 database outcome**
Google Chat	Readable dynamite.db; deletion-related tombstone identified

Table 6: Experiment 1 database-level implementation outcome 1

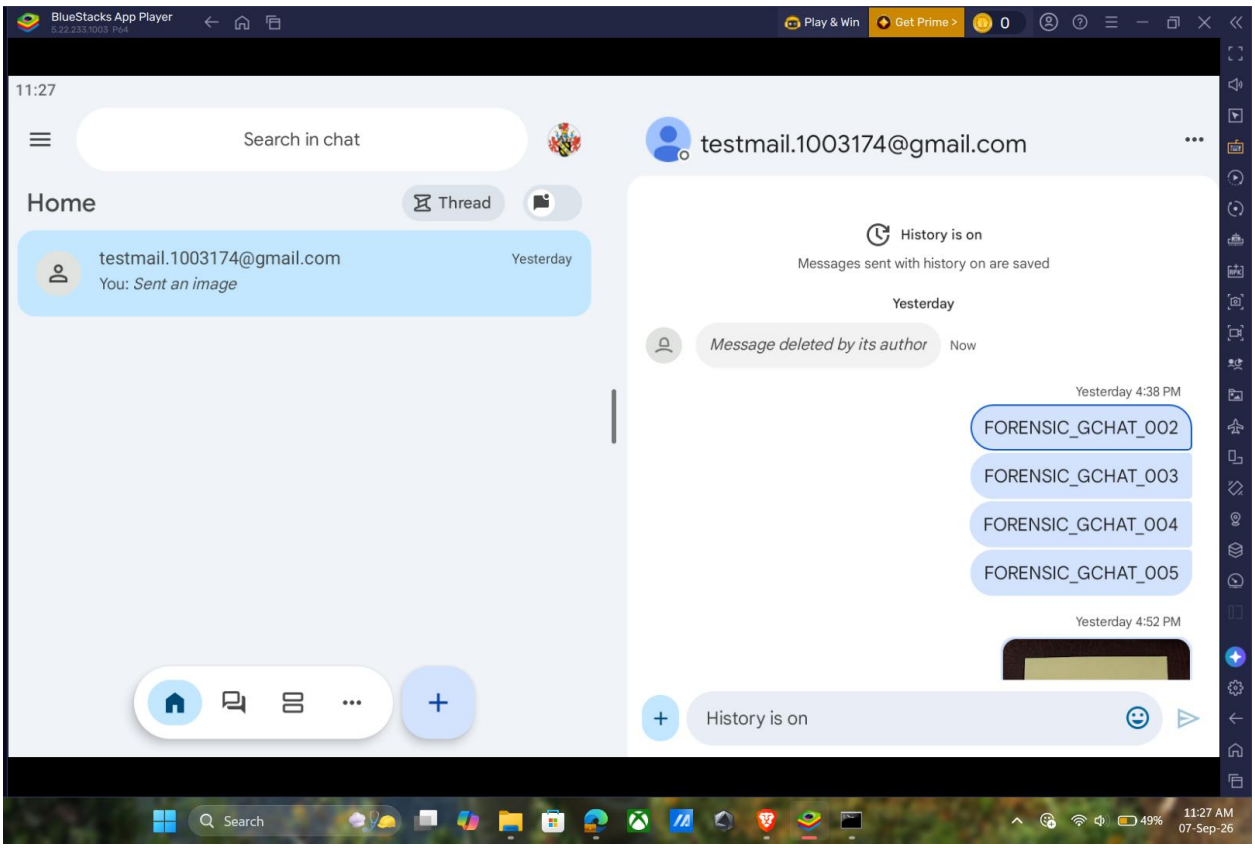
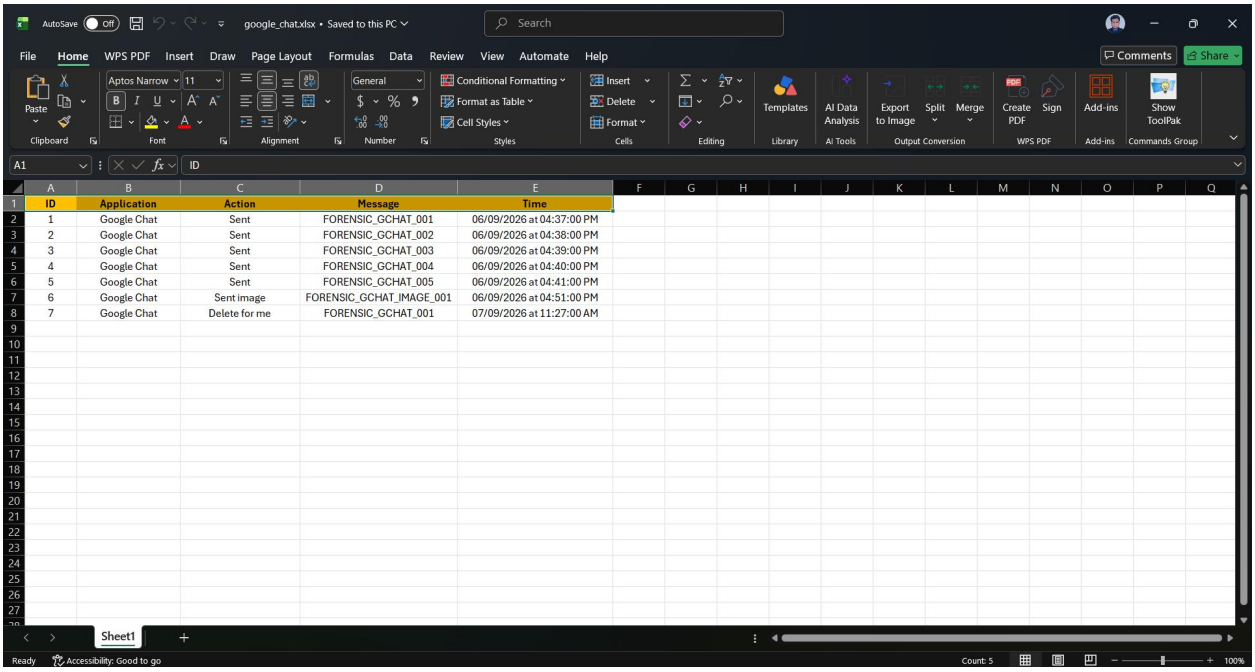


Figure 29: Experiment 1 Google Chat View

Figure 29 demonstrates the Google Chat conversation after the controlled text message was deleted.



ID	Application	Action	Message	Time
1	Google Chat	Sent	FORENSIC_GCHAT_001	06/09/2026 at 04:37:00 PM
2	Google Chat	Sent	FORENSIC_GCHAT_002	06/09/2026 at 04:38:00 PM
3	Google Chat	Sent	FORENSIC_GCHAT_003	06/09/2026 at 04:39:00 PM
4	Google Chat	Sent	FORENSIC_GCHAT_004	06/09/2026 at 04:40:00 PM
5	Google Chat	Sent	FORENSIC_GCHAT_005	06/09/2026 at 04:41:00 PM
6	Google Chat	Sent image	FORENSIC_GCHAT_IMAGE_001	06/09/2026 at 04:51:00 PM
7	Google Chat	Delete for me	FORENSIC_GCHAT_001	07/09/2026 at 11:27:00 AM

Figure 30: Experiment 1 Google Chat csv details

Five text messages were sent on 6 September 2026 from 4:37 PM to 4:41 PM, followed by FORENSIC_GCHAT_IMAGE_001 at 4:51 PM. FORENSIC_GCHAT_001 was

deleted on 7 September at 11:27 AM. The Before and After shared-storage manifests each contained 77 records, and corresponding relative paths and hashes matched.

```
D:\UWTSD\adb\platform-tools>adb connect 127.0.0.1:5555
connected to 127.0.0.1:5555

D:\UWTSD\adb\platform-tools>adb devices
List of devices attached
127.0.0.1:5555  device


D:\UWTSD\adb\platform-tools>adb shell getprop ro.build.version.release
9

D:\UWTSD\adb\platform-tools>adb shell run-as com.google.android.apps.dynamite pwd
run-as: setegid(AID_PACKAGE_INFO) failed: Operation not permitted

D:\UWTSD\adb\platform-tools>run-as
'run-as' is not recognized as an internal or external command,
operable program or batch file.
```

Figure 31: Running run-as command

Figure 31 shows that Private access attempts returned permission errors and run-as failed. Backup-related work nevertheless produced googlechat_backup.ab and a readable dynamite.db. The backup file was hashed as 57D244C8D4630C0683ED5DB36E9DCF3DABFC57167BCBBC0786B5B6F1BDD6DCCA, and dynamite.db as 628A35FEA8A5F47B669AB9BCE200FAD9E84A9239B5539700A4D5D897F780B662.

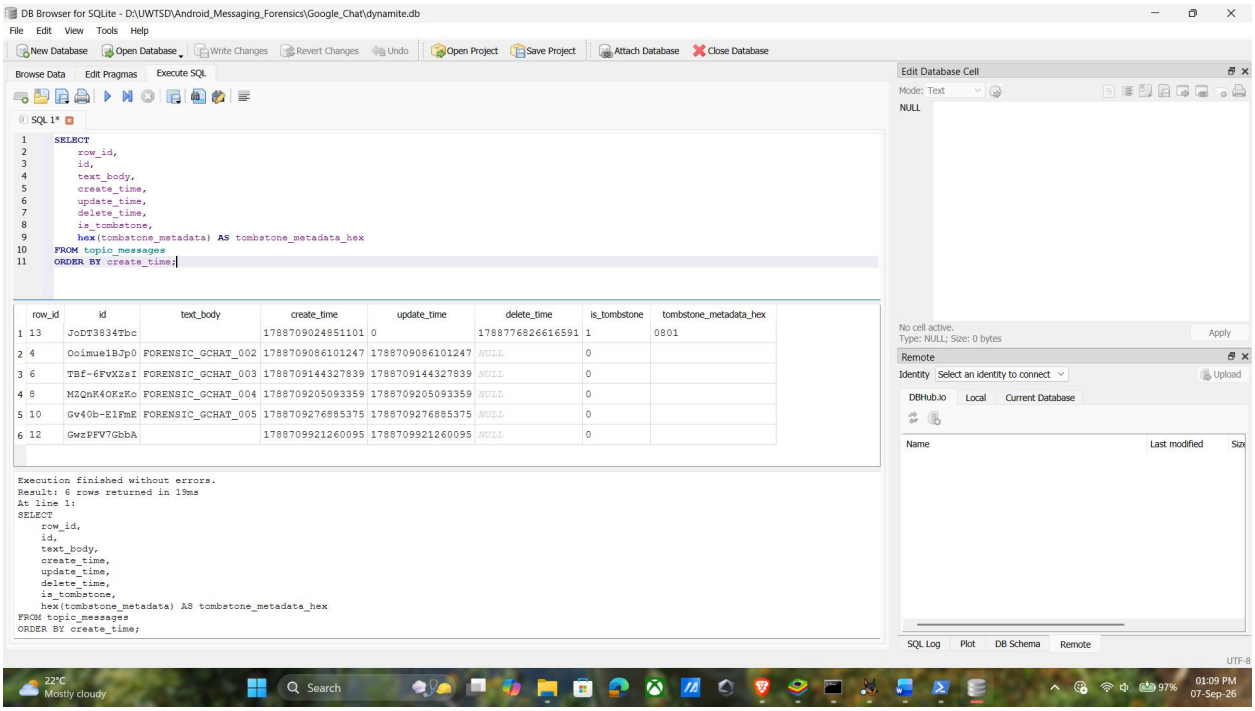


Figure 32: Google Chat tombstone record in DB Browser for SQLite

In DB Browser Figure 32, topic_messages contained visible records for FORENSIC_GCHAT_002, FORENSIC_GCHAT _003, FORENSIC_GCHAT _004 and FORENSIC_GCHAT_005. A separate record had row_id 13, id JoDT3834Tbc, empty

text_body, create_time 1788709024851101, delete_time 1788776826616591, is_tombstone 1 and tombstone_metadata 0801. The database therefore exposed a deletion-related tombstone, but not the original text of the deleted controlled message. Exact Autopsy searches for FORENSIC_GCHAT_001 returned no result in both shared-storage states. The following Figure 33 shows the query.

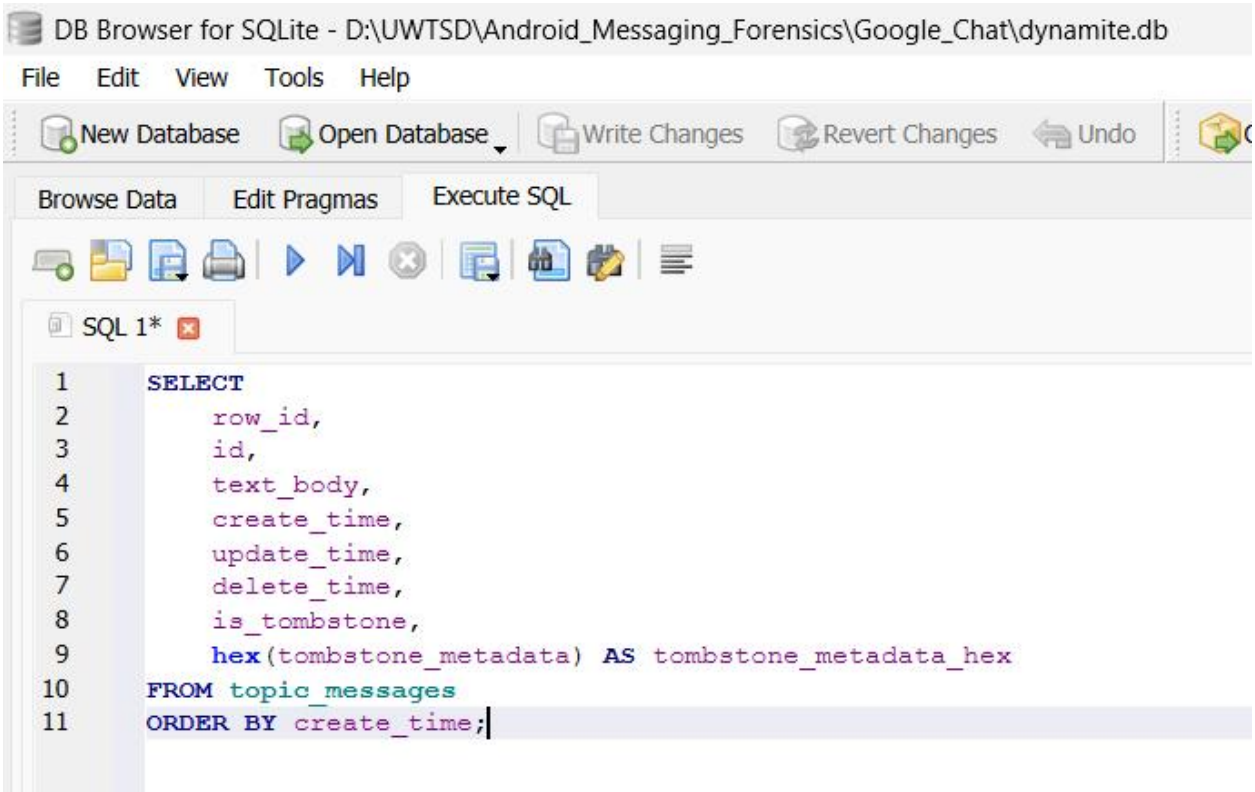


Figure 33: Google Chat SQL query

5.3 WhatsApp and Telegram Experiment 1 implementation

For WhatsApp, the available shared-storage database was msgstore.db.crypt14. It was preserved and hashed, but it could not be treated as a readable SQLite database without the required decryption material. For Telegram, Backup Manager reported successful package processing, but no readable private message database was obtained. These outcomes were documented rather than forcing the same database-analysis method onto all three applications.

Application	Experiment 1 database outcome
WhatsApp	Encrypted msgstore.db.crypt14; not readable as normal SQLite evidence
Telegram	No readable private message database acquired

Table 7: Experiment 1 database-level implementation outcome 2

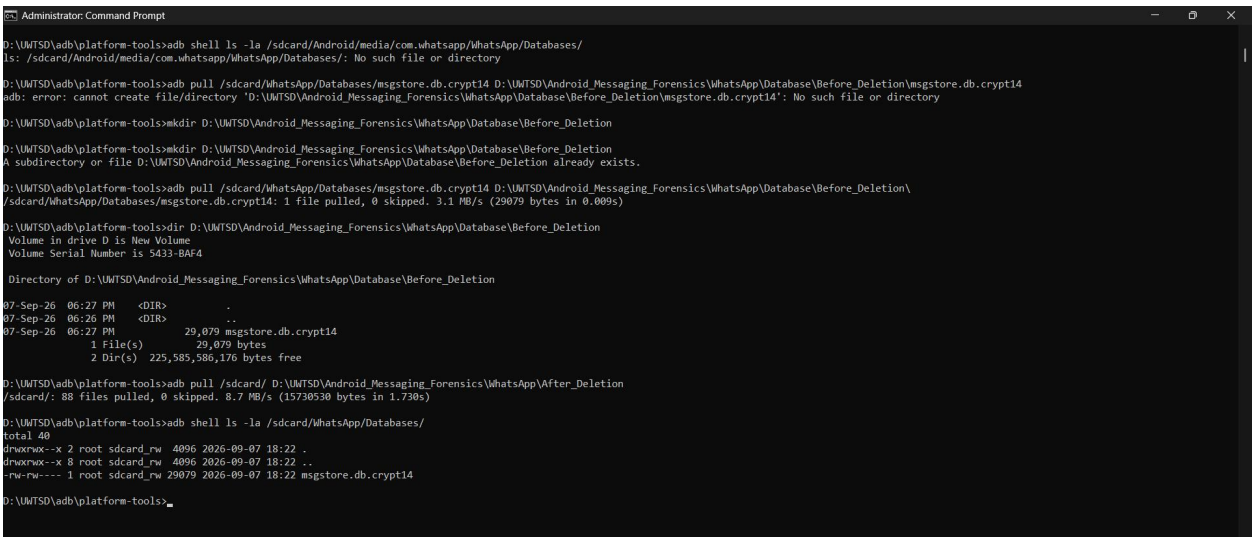


Figure 34: Acquisition of the encrypted WhatsApp database during Experiment 1

Figure 34 shows the successful acquisition of the WhatsApp database file (msgstore.db.crypt14) using ADB. The file was preserved in the Before_Deletion folder. As the acquired database was encrypted, it could not be examined as readable SQLite data without the required decryption material.

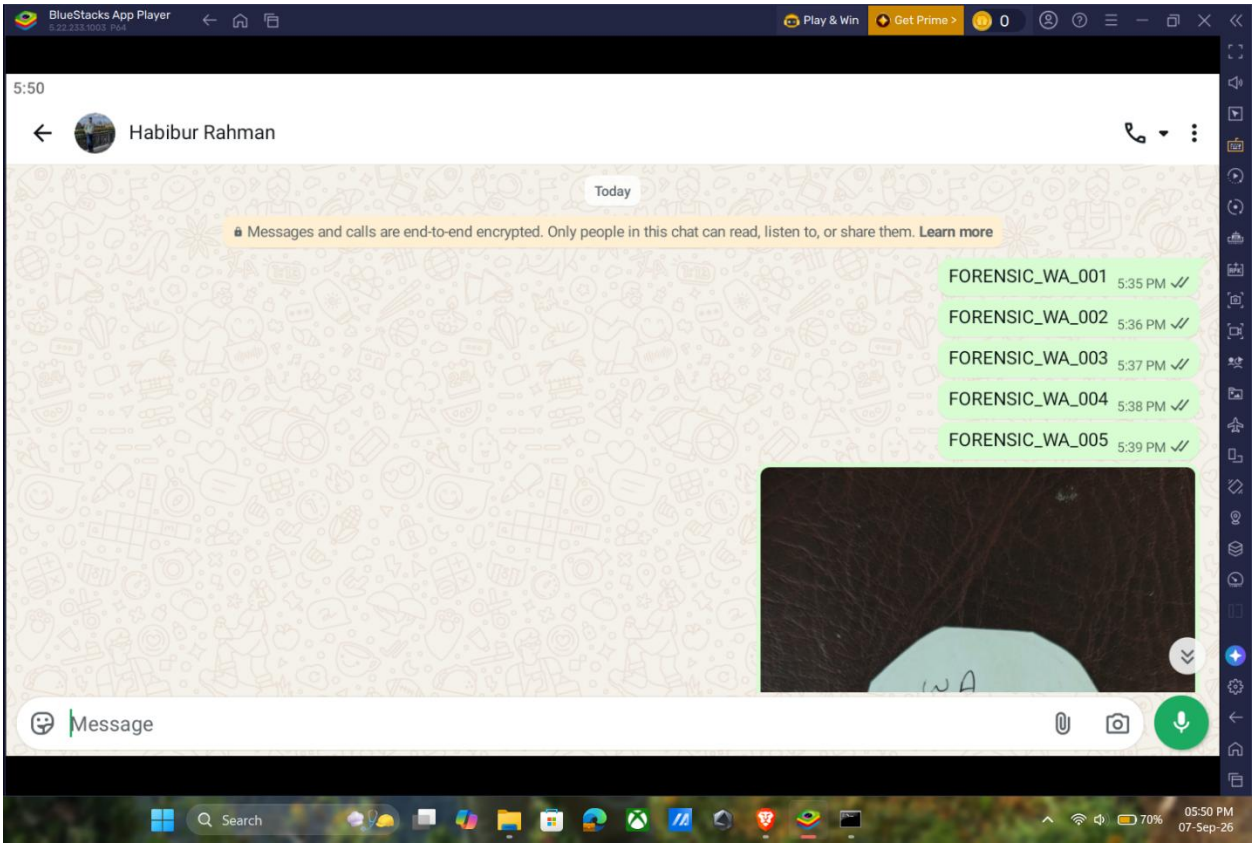


Figure 35: Experiment 1 of WhatsApp Before text Deletion

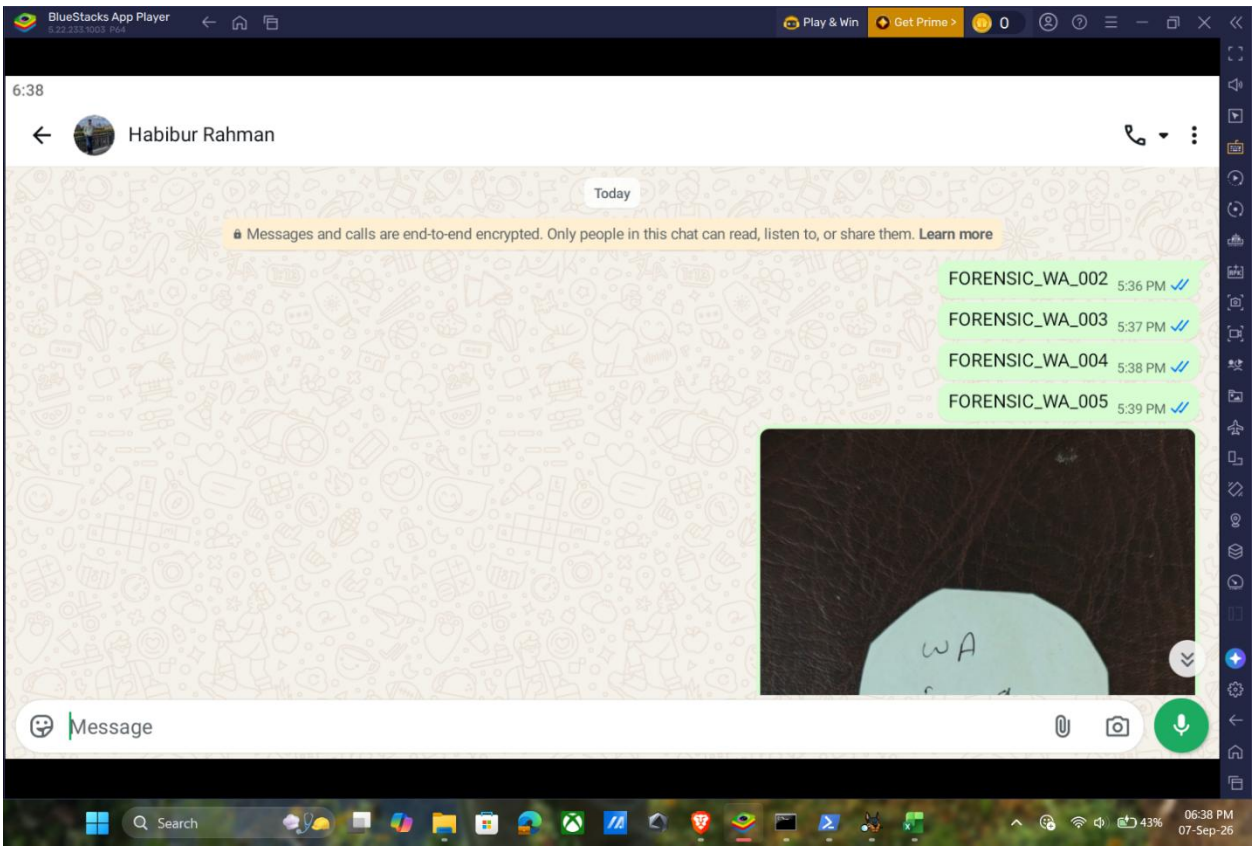
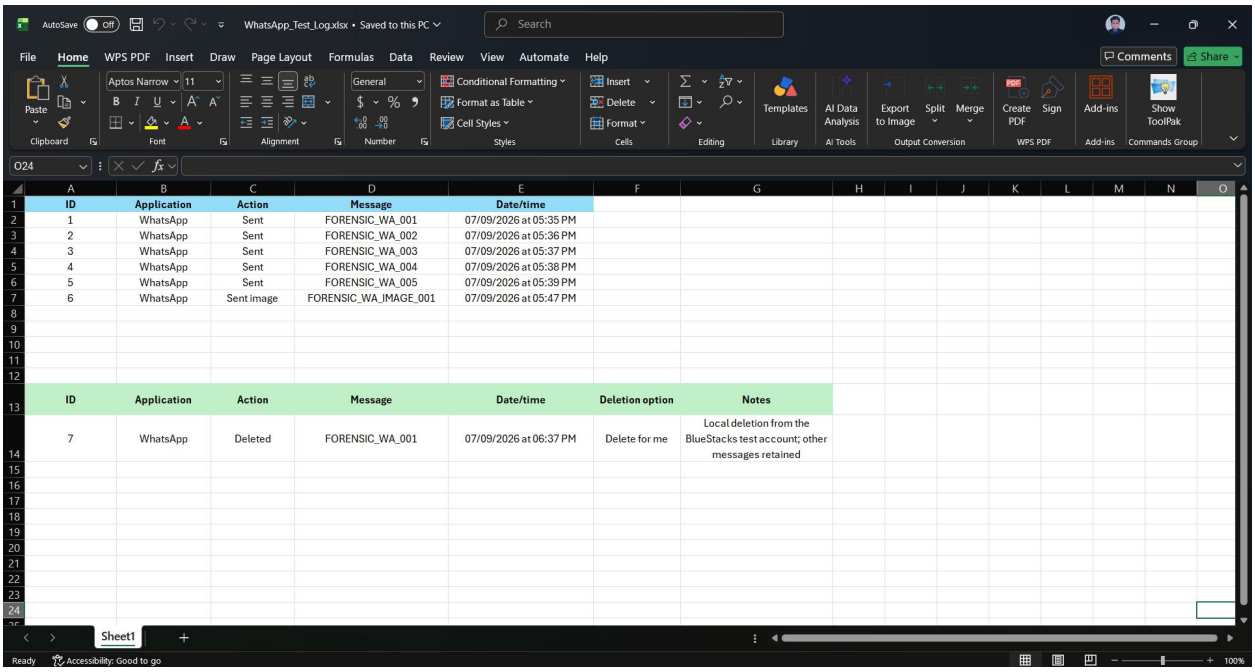


Figure 36: Experiment 1 of WhatsApp After text Deletion

Figure 35 and Figure 36 shows the Before and after view.



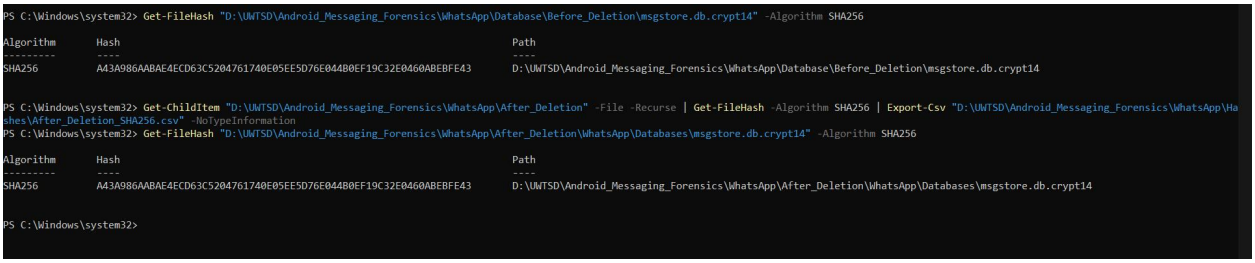
ID	Application	Action	Message	Date/time
1	WhatsApp	Sent	FORENSIC_WA_001	07/09/2026 at 05:35 PM
2	WhatsApp	Sent	FORENSIC_WA_002	07/09/2026 at 05:36 PM
3	WhatsApp	Sent	FORENSIC_WA_003	07/09/2026 at 05:37 PM
4	WhatsApp	Sent	FORENSIC_WA_004	07/09/2026 at 05:38 PM
5	WhatsApp	Sent	FORENSIC_WA_005	07/09/2026 at 05:39 PM
6	WhatsApp	Sent image	FORENSIC_WA_IMAGE_001	07/09/2026 at 05:47 PM

ID	Application	Action	Message	Date/time	Deletion option	Notes
7	WhatsApp	Deleted	FORENSIC_WA_001	07/09/2026 at 06:37 PM	Delete for me	Local deletion from the BlueStacks test account; other messages retained

Figure 37: Log file of WhatsApp Experiment 1

According to Figure 37 WhatsApp Experiment 1 created FORENSIC_WA_001 through _005 and an image. The BEFORE acquisition contained 80 files and 15,574,102 bytes.

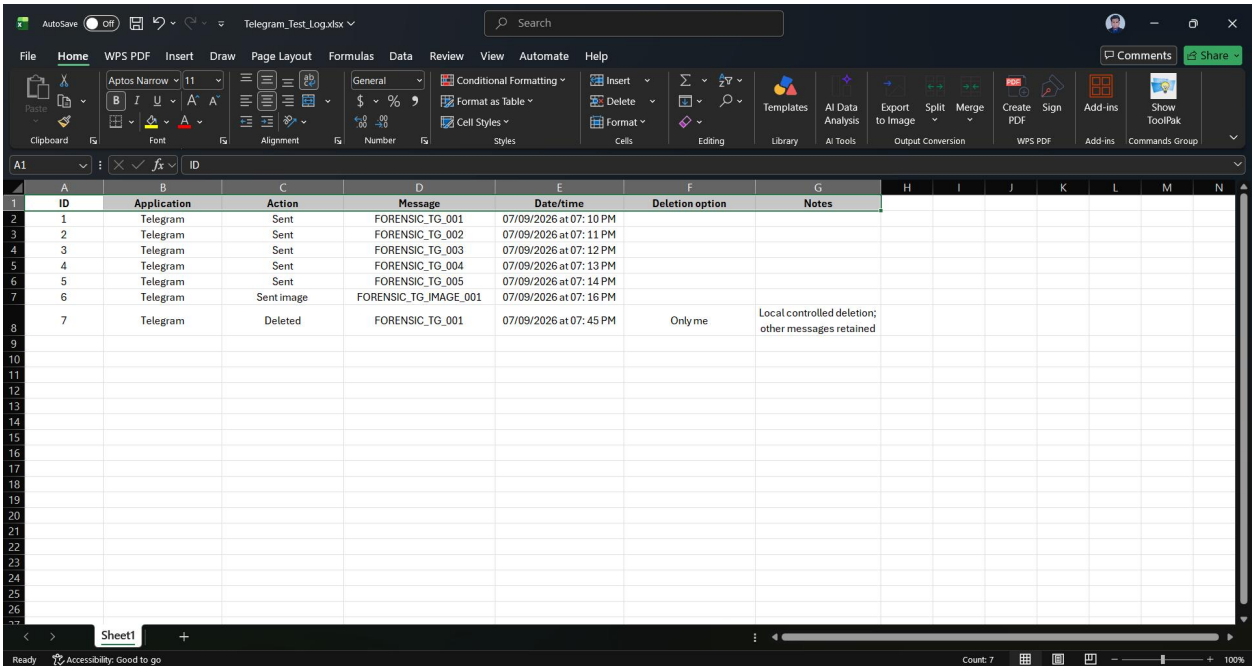
FORENSIC_WA_001 was deleted at 6:37 PM on 7 September using Delete for me. The AFTER acquisition contained 88 files and 15,730,530 bytes. Manifest comparison recorded 80 unchanged corresponding paths, 8 additions, 0 removals and 0 changes.



```
PS C:\Windows\system32> Get-FileHash "D:\UWTSD\Android_Messaging_Forensics\WhatsApp\Database\Before_Deletion\msgstore.db.crypt14" -Algorithm SHA256

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          A43A986AABAE4ECD63C5204761740E05EE5D76E044B0EF19C32E0460ABEBFE43       D:\UWTSD\Android_Messaging_Forensics\WhatsApp\Database\Before_Deletion\msgstore.db.crypt14


PS C:\Windows\system32> Get-ChildItem "D:\UWTSD\Android_Messaging_Forensics\WhatsApp\After_Deletion" -File -Recurse | Get-FileHash -Algorithm SHA256 | Export-Csv "D:\UWTSD\Android_Messaging_Forensics\WhatsApp\Ha
shes\After_Deletion_SHA256.csv" -NoTypeInformation
PS C:\Windows\system32> Get-FileHash "D:\UWTSD\Android_Messaging_Forensics\WhatsApp\After_Deletion\WhatsApp\Databases\msgstore.db.crypt14" -Algorithm SHA256

Algorithm       Hash                                                                   Path
---------       ----                                                                   ----
SHA256          A43A986AABAE4ECD63C5204761740E05EE5D76E044B0EF19C32E0460ABEBFE43       D:\UWTSD\Android_Messaging_Forensics\WhatsApp\After_Deletion\WhatsApp\Databases\msgstore.db.crypt14


PS C:\Windows\system32>
```

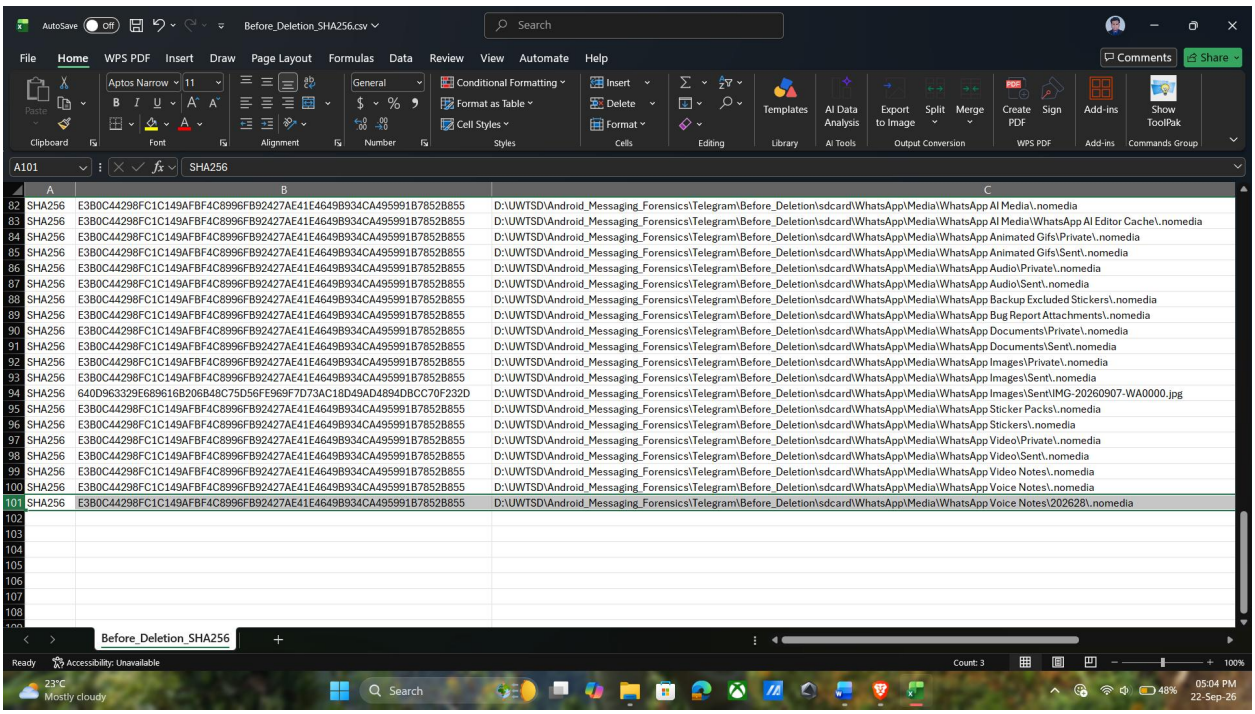
Figure 38: Hash Verification of WhatsApp Experiment 1

A shared-storage msgstore.db.crypt14 was separately preserved before deletion. It was 29,079 bytes with SHA-256 A43A986AABAE4ECD63C5204761740E05EE5D76E044B0EF19C32E0460ABEBFE43. The later captured copy had the same hash. Private access was denied, run-as failed, and the WhatsApp package backup was rejected by the transport. Autopsy did not find FORENSIC_WA_001 in either the Before or After acquisition. Figure 38 shows the hash verification of the preserved WhatsApp database.



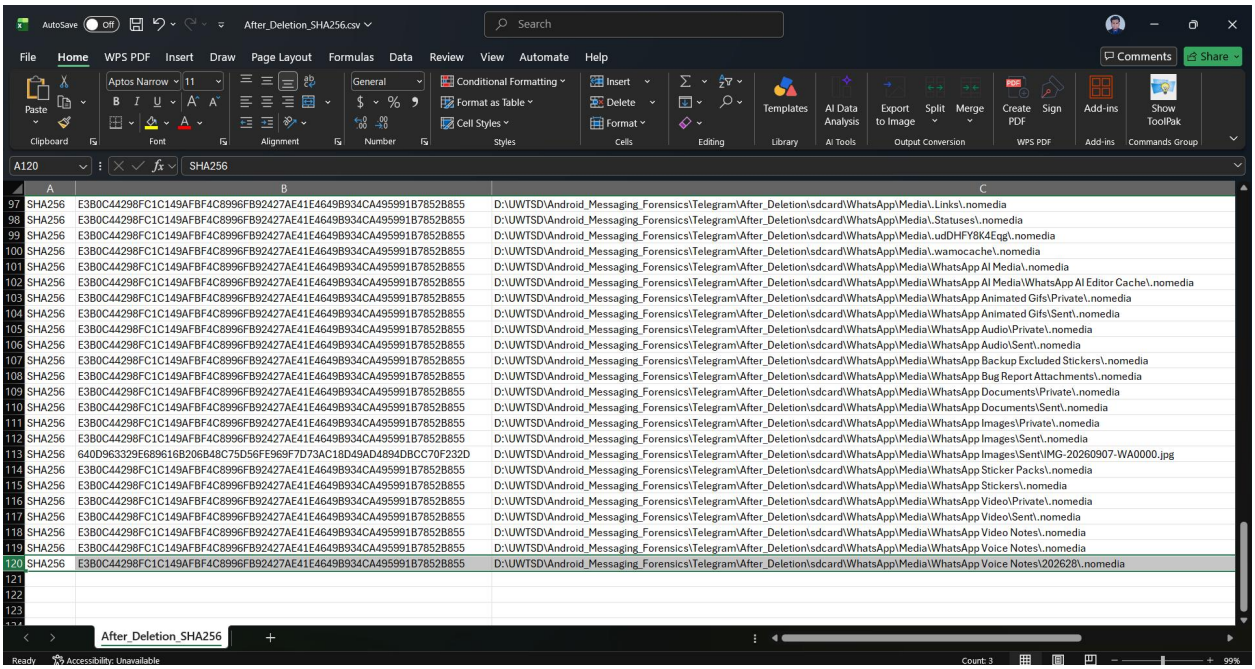
ID	Application	Action	Message	Date/time	Deletion option	Notes
1	Telegram	Sent	FORENSIC_TG_001	07/09/2026 at 07: 10 PM		
2	Telegram	Sent	FORENSIC_TG_002	07/09/2026 at 07: 11 PM		
3	Telegram	Sent	FORENSIC_TG_003	07/09/2026 at 07: 12 PM		
4	Telegram	Sent	FORENSIC_TG_004	07/09/2026 at 07: 13 PM		
5	Telegram	Sent	FORENSIC_TG_005	07/09/2026 at 07: 14 PM		
6	Telegram	Sent image	FORENSIC_TG_IMAGE_001	07/09/2026 at 07: 16 PM		
7	Telegram	Deleted	FORENSIC_TG_001	07/09/2026 at 07: 45 PM	Only me	Local controlled deletion; other messages retained

Figure 39: Telegram Log of Experiment 1



	A	B	C
82	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp AI Media\.nomedia
83	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp AI Media\WhatsApp AI Editor Cache\.nomedia
84	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Animated Gifs\Private\.nomedia
85	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Animated Gifs\Sent\.nomedia
86	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Audio\Private\.nomedia
87	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Audio\Sent\.nomedia
88	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Backup Excluded Stickers\.nomedia
89	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Bug Report Attachments\.nomedia
90	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Documents\Private\.nomedia
91	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Documents\Sent\.nomedia
92	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Images\Private\.nomedia
93	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Images\Sent\.nomedia
94	SHA256	640D963329E689616B206B48C75D56FE969F7D73AC18D49AD4894DBCC70F232D	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Images\Sent\IMG-20260907-WA0000.jpg
95	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Sticker Packs\.nomedia
96	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Stickers\.nomedia
97	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Video\Private\.nomedia
98	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Video\Sent\.nomedia
99	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Video Notes\.nomedia
100	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Voice Notes\.nomedia
101	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\Before_Deletion\sdcard\WhatsApp\Media\WhatsApp Voice Notes\202628\.nomedia

Figure 40: Telegram before text deletion Experiment 1



	A	B	C
97	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\.Links\.nomedia
98	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\.Statuses\.nomedia
99	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\.udDHFY8K4Eqg\.nomedia
100	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\.wamocache\.nomedia
101	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp AI Media\.nomedia
102	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp AI Media\WhatsApp AI Editor Cache\.nomedia
103	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Animated Gifs\Private\.nomedia
104	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Animated Gifs\Sent\.nomedia
105	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Audio\Private\.nomedia
106	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Audio\Sent\.nomedia
107	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Backup Excluded Stickers\.nomedia
108	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Bug Report Attachments\.nomedia
109	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Documents\Private\.nomedia
110	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Documents\Sent\.nomedia
111	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Images\Private\.nomedia
112	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Images\Sent\.nomedia
113	SHA256	640D963329E689616B206B48C75D56FE969F7D73AC18D49AD4894DBCC70F232D	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Images\Sent\IMG-20260907-WA0000.jpg
114	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Sticker Packs\.nomedia
115	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Stickers\.nomedia
116	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Video\Private\.nomedia
117	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Video\Sent\.nomedia
118	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Video Notes\.nomedia
119	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Voice Notes\.nomedia
120	SHA256	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855	D:\UWTSD\Android_Messaging_Forensics\Telegram\After_Deletion\sdcard\WhatsApp\Media\WhatsApp Voice Notes\202628\.nomedia

Figure 41: Telegram After text deletion Experiment 1

Telegram Experiment 1 used FORENSIC_TG_001 through _005 and an image. Before contained 100 files and 23,987,305 bytes. FORENSIC_TG_001 was deleted at 7:45 PM using Only me. After contained 119 files and 27,171,662 bytes. The comparison recorded 19 additions, consisting of 16 .tgs and 3 .pcache2 files, with 0 removed and 0 changed. Which is shown in Figure 39, Figure 40, and Figure

41.

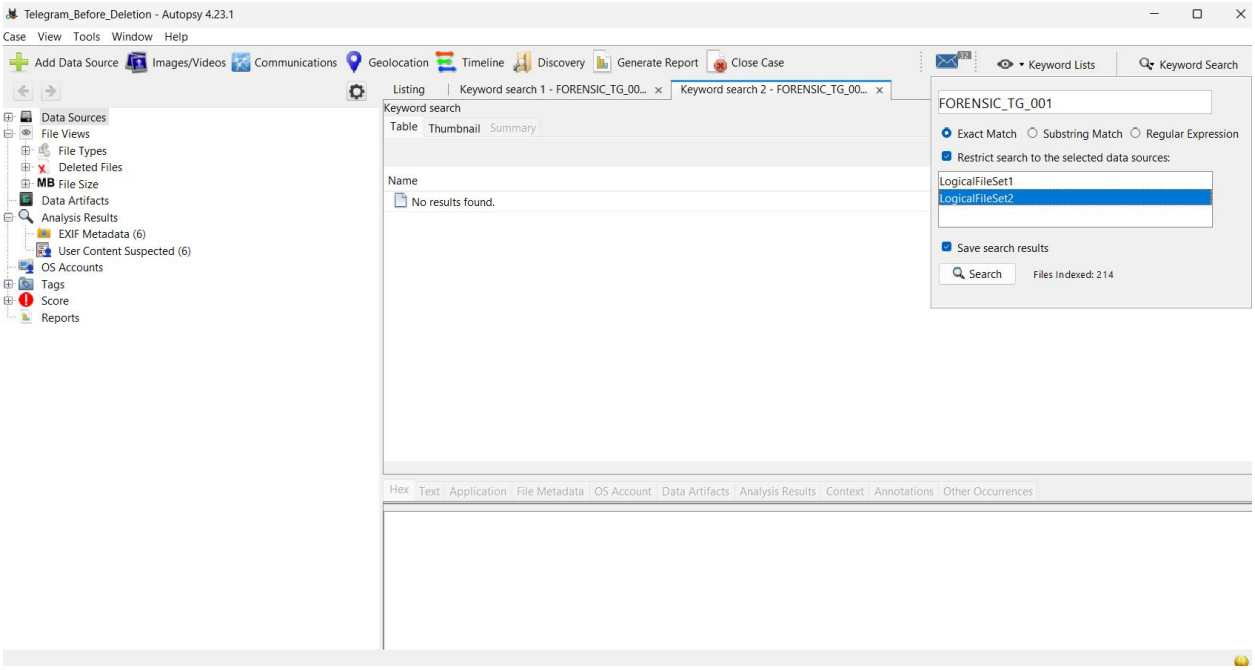


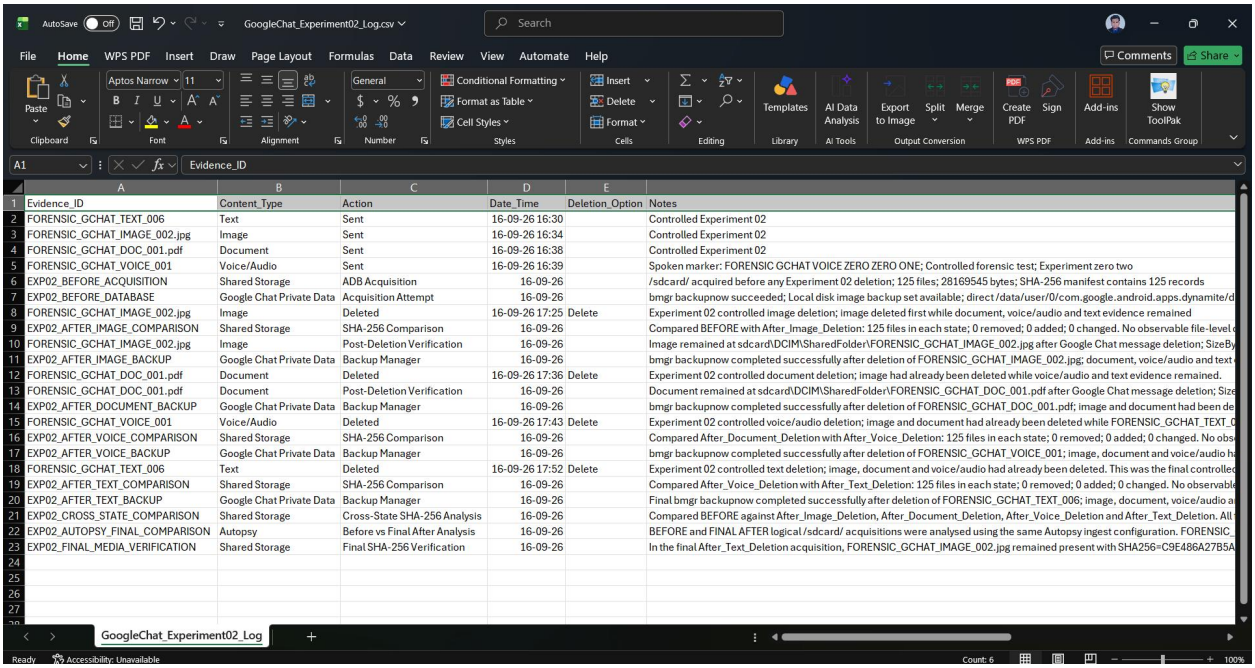
Figure 42: Autopsy Result of Telegram Experiment 1

Private access was denied and run-as failed. Backup Manager reported Telegram package Success, but no readable message database or exported backup was obtained. Autopsy returned no result for the deleted marker in either state. Which is demonstrate in Figure 42.

5.4 Experiment 2 implementation

Experiment 2 used four controlled evidence types for each application. After the Before acquisition, each evidence type was deleted in the planned order and /sdcard/ was acquired again. Separate hash manifests were generated for each state. The final analysis compared both consecutive stages and the original Before state with the Final state.

5.5 Google Chat implementation



Evidence_ID	Content_Type	Action	Date_Time	Deletion_Option	Notes
FORENSIC_GCHAT_TEXT_006	Text	Sent	16-09-26 16:30		Controlled Experiment 02
FORENSIC_GCHAT_IMAGE_002.jpg	Image	Sent	16-09-26 16:34		Controlled Experiment 02
FORENSIC_GCHAT_DOC_001.pdf	Document	Sent	16-09-26 16:38		Controlled Experiment 02
FORENSIC_GCHAT_VOICE_001	Voice/Audio	Sent	16-09-26 16:39		Spoken marker: FORENSIC GCHAT VOICE ZERO ZERO ONE; Controlled forensic test; Experiment zero two
EXP02_BEFORE_ACQUISITION	Shared Storage	ADB Acquisition	16-09-26		/sdcard/ acquired before any Experiment 02 deletion; 125 files; 28169545 bytes; SHA-256 manifest contains 125 records
EXP02_BEFORE_DATABASE	Google Chat Private Data	Acquisition Attempt	16-09-26		bmgr backupnow succeeded; Local disk image backup set available; direct /data/user/0/com.google.android.apps.dynamite/d
FORENSIC_GCHAT_IMAGE_002.jpg	Image	Deleted	16-09-26 17:25	Delete	Experiment 02 controlled image deletion; image deleted first while document, voice/audio and text evidence remained
EXP02_AFTER_IMAGE_COMPARISON	Shared Storage	SHA-256 Comparison	16-09-26		Compared BEFORE with After_Image_Deletion: 125 files in each state; 0 removed; 0 added; 0 changed. No observable file-level c
FORENSIC_GCHAT_IMAGE_002.jpg	Image	Post-Deletion Verification	16-09-26		Image remained at sdcard\DCIM\SharedFolder\FORENSIC_GCHAT_IMAGE_002.jpg after Google Chat message deletion; SizeBy
EXP02_AFTER_IMAGE_BACKUP	Google Chat Private Data	Backup Manager	16-09-26		bmgr backupnow completed successfully after deletion of FORENSIC_GCHAT_IMAGE_002.jpg; document, voice/audio and text
FORENSIC_GCHAT_DOC_001.pdf	Document	Deleted	16-09-26 17:36	Delete	Experiment 02 controlled document deletion; image had already been deleted while voice/audio and text evidence remained.
FORENSIC_GCHAT_DOC_001.pdf	Document	Post-Deletion Verification	16-09-26		Document remained at sdcard\DCIM\SharedFolder\FORENSIC_GCHAT_DOC_001.pdf after Google Chat message deletion; Size
EXP02_AFTER_DOCUMENT_BACKUP	Google Chat Private Data	Backup Manager	16-09-26		bmgr backupnow completed successfully after deletion of FORENSIC_GCHAT_DOC_001.pdf; image and document had been de
FORENSIC_GCHAT_VOICE_001	Voice/Audio	Deleted	16-09-26 17:43	Delete	Experiment 02 controlled voice/audio deletion; image and document had already been deleted while FORENSIC_GCHAT_TEXT_0
EXP02_AFTER_VOICE_COMPARISON	Shared Storage	SHA-256 Comparison	16-09-26		Compared After_Document_Deletion with After_Voice_Deletion: 125 files in each state; 0 removed; 0 added; 0 changed. No obs
EXP02_AFTER_VOICE_BACKUP	Google Chat Private Data	Backup Manager	16-09-26		bmgr backupnow completed successfully after deletion of FORENSIC_GCHAT_VOICE_001; image, document and voice/audio h
FORENSIC_GCHAT_TEXT_006	Text	Deleted	16-09-26 17:52	Delete	Experiment 02 controlled text deletion; image, document and voice/audio had already been deleted. This was the final controlled
EXP02_AFTER_TEXT_COMPARISON	Shared Storage	SHA-256 Comparison	16-09-26		Compared After_Voice_Deletion with After_Text_Deletion: 125 files in each state; 0 removed; 0 added; 0 changed. No observable
EXP02_AFTER_TEXT_BACKUP	Google Chat Private Data	Backup Manager	16-09-26		Final bmgr backupnow completed successfully after deletion of FORENSIC_GCHAT_TEXT_006; image, document, voice/audio a
EXP02_CROSS_STATE_COMPARISON	Shared Storage	Cross-State SHA-256 Analysis	16-09-26		Compared BEFORE against After_Image_Deletion, After_Document_Deletion, After_Voice_Deletion and After_Text_Deletion. All
EXP02_AUTOPSY_FINAL_COMPARISON	Autopsy	Before vs Final After Analysis	16-09-26		BEFORE and FINAL AFTER logical /sdcard/ acquisitions were analysed using the same Autopsy ingest configuration. FORENSIC_
EXP02_FINAL_MEDIA_VERIFICATION	Shared Storage	Final SHA-256 Verification	16-09-26		In the final After_Text_Deletion acquisition, FORENSIC_GCHAT_IMAGE_002.jpg remained present with SHA256=C9E486A27B5A

Figure 43: Google Chat Log File Analysis for Experiment 2

Figure 43 illustrates that Google Chat Experiment 2 used FORENSIC_GCHAT_TEXT_006, FORENSIC_GCHAT_IMAGE_002.jpg, FORENSIC_GCHAT_DOC_001.pdf and FORENSIC_GCHAT_VOICE_001. The items were sent on 16 September 2026 between 4:30 PM and 4:39 PM. Deletion occurred sequentially: image at 5:25 PM, document at 5:36 PM, voice at 5:43 PM and text at 5:52 PM.

The Before acquisition and all four post-deletion acquisitions each contained 125 files and 28,169,545 bytes. Every adjacent comparison produced, 0 removed, 0 added and 0 changed files. Direct Before then Final comparison also produced 0/0/0. These results provide clear file-system observation: the acquired shared-storage collection did not change at the file level during the controlled deletion sequence.

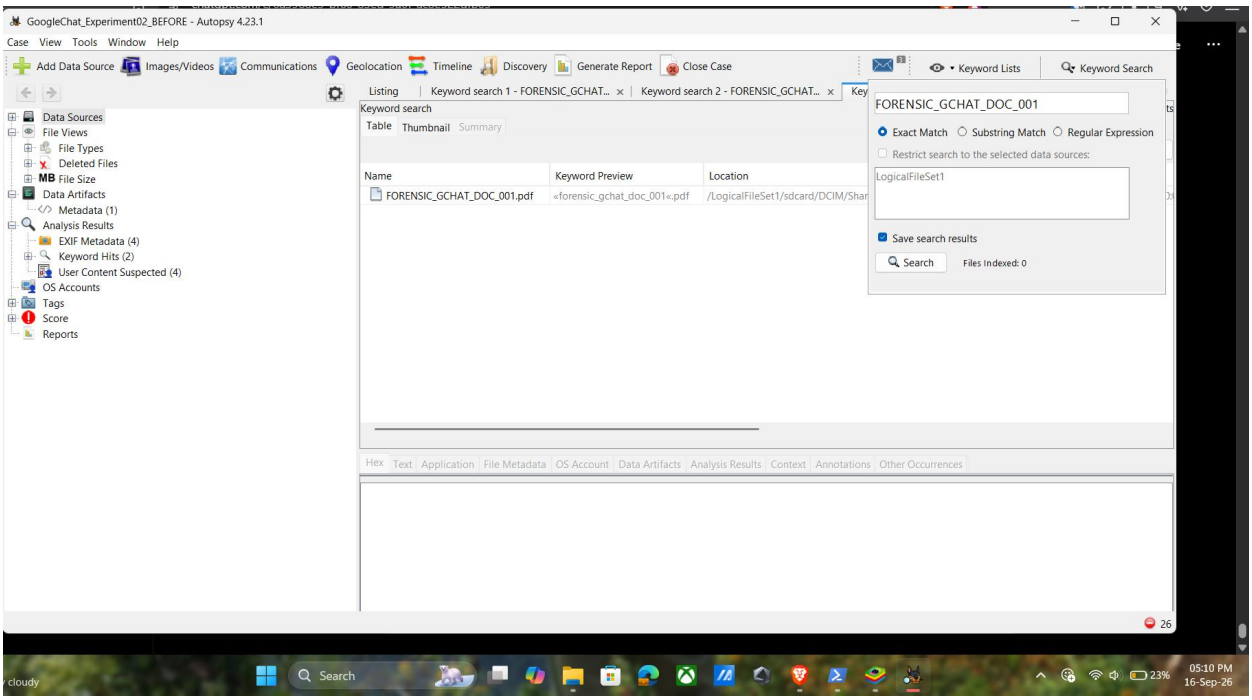


Figure 44: Google Chat Before Autopsy Evidence Experiment 2

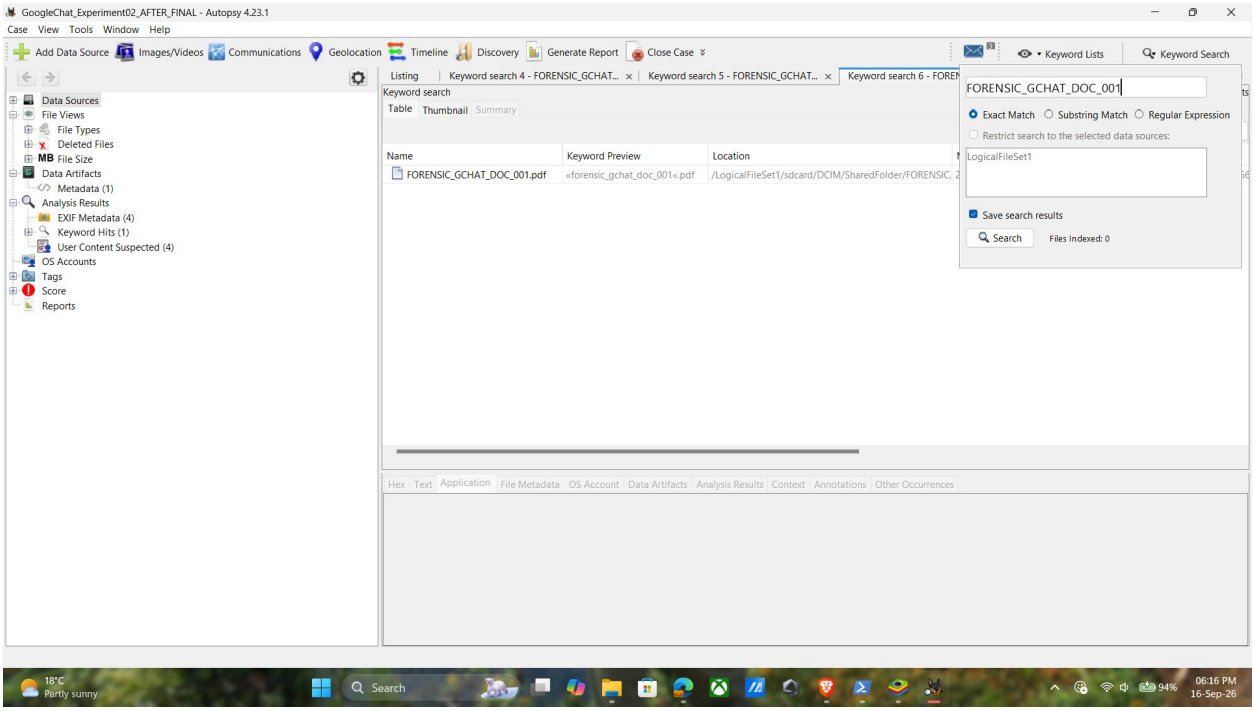
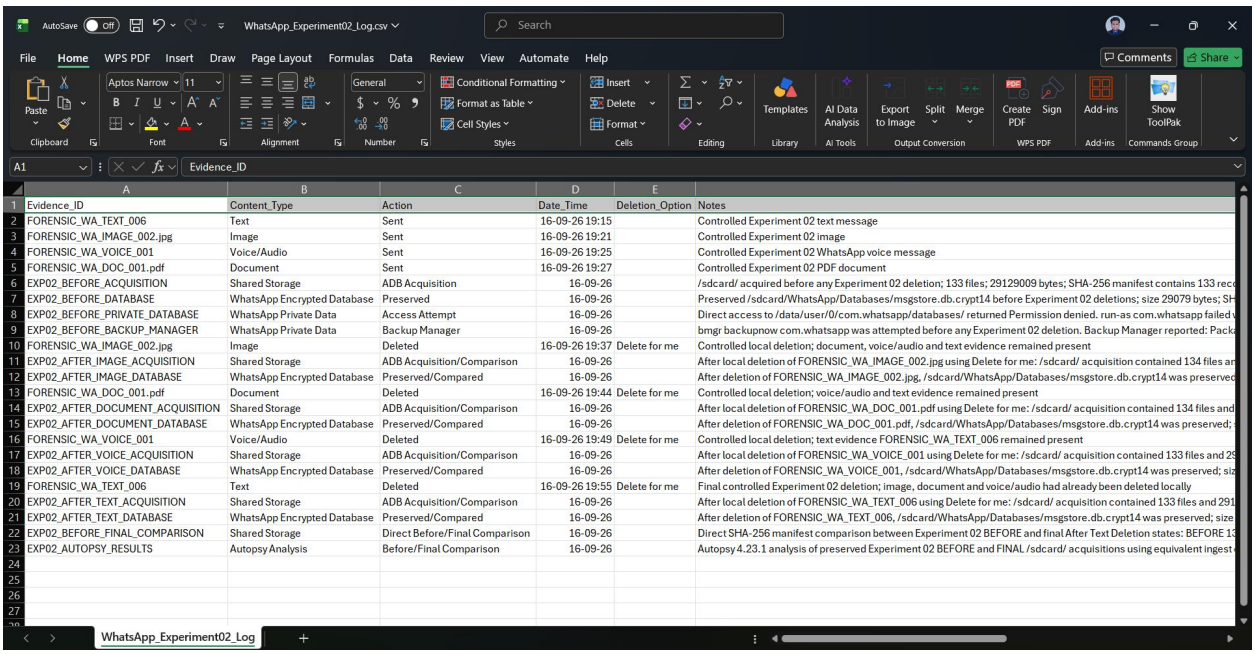


Figure 45: Google Chat After Autopsy Evidence Experiment 2

Autopsy found the image marker in Before and Final and found the PDF in Before and Final. The text marker was not found in either state. No corresponding controlled voice file was identified in the acquired shared storage. The final image SHA-256 C9E486A27B5A6897C02D53CBC2EB06B1A86098FE2ABDB19FCDC412220275B7EC matched the original. The final PDF SHA-256 08A4ABE09BC095D7545729590AD219E84123595E2318D41BD9A1B5DF6F39BB7

5 also matched the original. The implementation therefore confirmed byte-for-byte persistence of those two shared-storage test files.

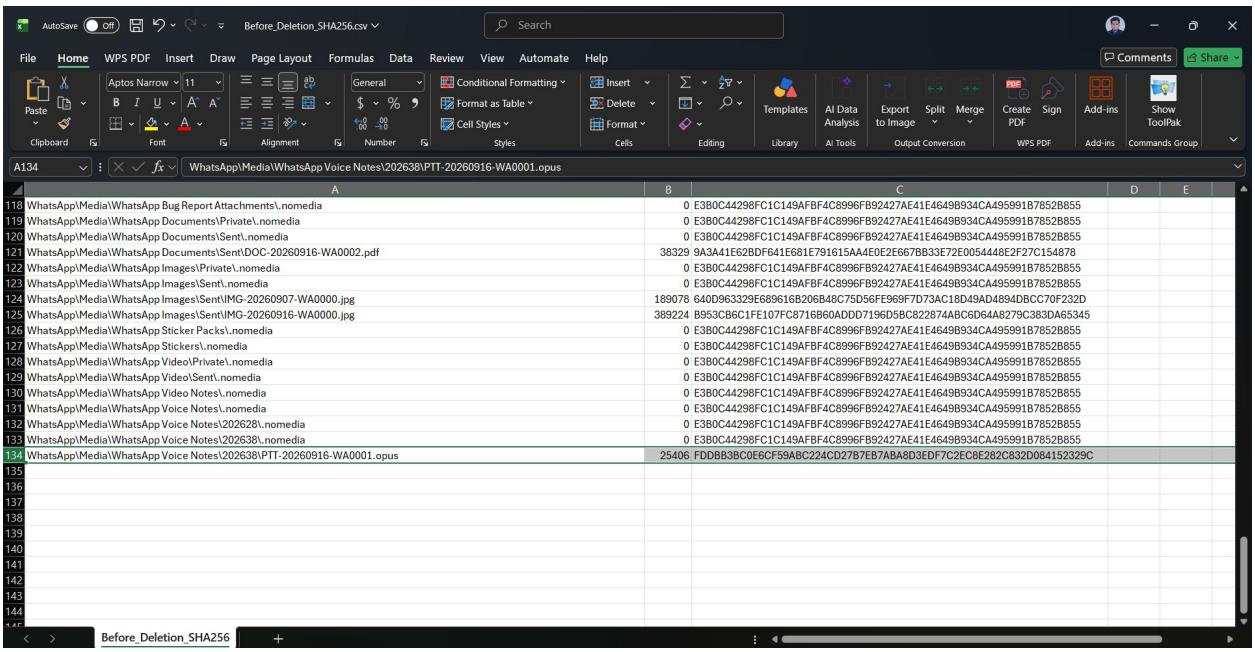
5.6 WhatsApp implementation



Evidence_ID	Content_Type	Action	Date_Time	Deletion_Option	Notes
FORENSIC_WA_TEXT_006	Text	Sent	16-09-26 19:15		Controlled Experiment 02 text message
FORENSIC_WA_IMAGE_002.jpg	Image	Sent	16-09-26 19:21		Controlled Experiment 02 image
FORENSIC_WA_VOICE_001	Voice/Audio	Sent	16-09-26 19:25		Controlled Experiment 02 WhatsApp voice message
FORENSIC_WA_DOC_001.pdf	Document	Sent	16-09-26 19:27		Controlled Experiment 02 PDF document
EXP02_BEFORE_ACQUISITION	Shared Storage	ADB Acquisition	16-09-26		/sdcard/ acquired before any Experiment 02 deletion; 133 files; 29129009 bytes; SHA-256 manifest contains 133 rec
EXP02_BEFORE_DATABASE	WhatsApp Encrypted Database	Preserved	16-09-26		Preserved /sdcard/WhatsApp/Databases/msgstore.db.crypt14 before Experiment 02 deletions; size 29079 bytes; SH
EXP02_BEFORE_PRIVATE_DATABASE	WhatsApp Private Data	Access Attempt	16-09-26		Direct access to /data/user/0/com.whatsapp/databases/ returned Permission denied. run-as com.whatsapp failed
EXP02_BEFORE_BACKUP_MANAGER	WhatsApp Private Data	Backup Manager	16-09-26		bmgr backupnow com.whatsapp was attempted before any Experiment 02 deletion. Backup Manager reported: Pack
FORENSIC_WA_IMAGE_002.jpg	Image	Deleted	16-09-26 19:37	Delete for me	Controlled local deletion; document, voice/audio and text evidence remained present
EXP02_AFTER_IMAGE_ACQUISITION	Shared Storage	ADB Acquisition/Comparison	16-09-26		After local deletion of FORENSIC_WA_IMAGE_002.jpg using Delete for me: /sdcard/ acquisition contained 134 files an
EXP02_AFTER_IMAGE_DATABASE	WhatsApp Encrypted Database	Preserved/Compared	16-09-26		After deletion of FORENSIC_WA_IMAGE_002.jpg, /sdcard/WhatsApp/Databases/msgstore.db.crypt14 was preserved
FORENSIC_WA_DOC_001.pdf	Document	Deleted	16-09-26 19:44	Delete for me	Controlled local deletion; voice/audio and text evidence remained present
EXP02_AFTER_DOCUMENT_ACQUISITION	Shared Storage	ADB Acquisition/Comparison	16-09-26		After local deletion of FORENSIC_WA_DOC_001.pdf using Delete for me: /sdcard/ acquisition contained 134 files and
EXP02_AFTER_DOCUMENT_DATABASE	WhatsApp Encrypted Database	Preserved/Compared	16-09-26		After deletion of FORENSIC_WA_DOC_001.pdf, /sdcard/WhatsApp/Databases/msgstore.db.crypt14 was preserved;
FORENSIC_WA_VOICE_001	Voice/Audio	Deleted	16-09-26 19:49	Delete for me	Controlled local deletion; text evidence FORENSIC_WA_TEXT_006 remained present
EXP02_AFTER_VOICE_ACQUISITION	Shared Storage	ADB Acquisition/Comparison	16-09-26		After local deletion of FORENSIC_WA_VOICE_001 using Delete for me: /sdcard/ acquisition contained 133 files and 29
EXP02_AFTER_VOICE_DATABASE	WhatsApp Encrypted Database	Preserved/Compared	16-09-26		After deletion of FORENSIC_WA_VOICE_001, /sdcard/WhatsApp/Databases/msgstore.db.crypt14 was preserved; siz
FORENSIC_WA_TEXT_006	Text	Deleted	16-09-26 19:55	Delete for me	Final controlled Experiment 02 deletion; image, document and voice/audio had already been deleted locally
EXP02_AFTER_TEXT_ACQUISITION	Shared Storage	ADB Acquisition/Comparison	16-09-26		After local deletion of FORENSIC_WA_TEXT_006 using Delete for me: /sdcard/ acquisition contained 133 files and 291
EXP02_AFTER_TEXT_DATABASE	WhatsApp Encrypted Database	Preserved/Compared	16-09-26		After deletion of FORENSIC_WA_TEXT_006, /sdcard/WhatsApp/Databases/msgstore.db.crypt14 was preserved; size
EXP02_BEFORE_FINAL_COMPARISON	Shared Storage	Direct Before/Final Comparison	16-09-26		Direct SHA-256 manifest comparison between Experiment 02 BEFORE and final After Text Deletion states: BEFORE 13
EXP02_AUTOPSY_RESULTS	Autopsy Analysis	Before/Final Comparison	16-09-26		Autopsy 4.23.1 analysis of preserved Experiment 02 BEFORE and FINAL /sdcard/ acquisitions using equivalent ingest

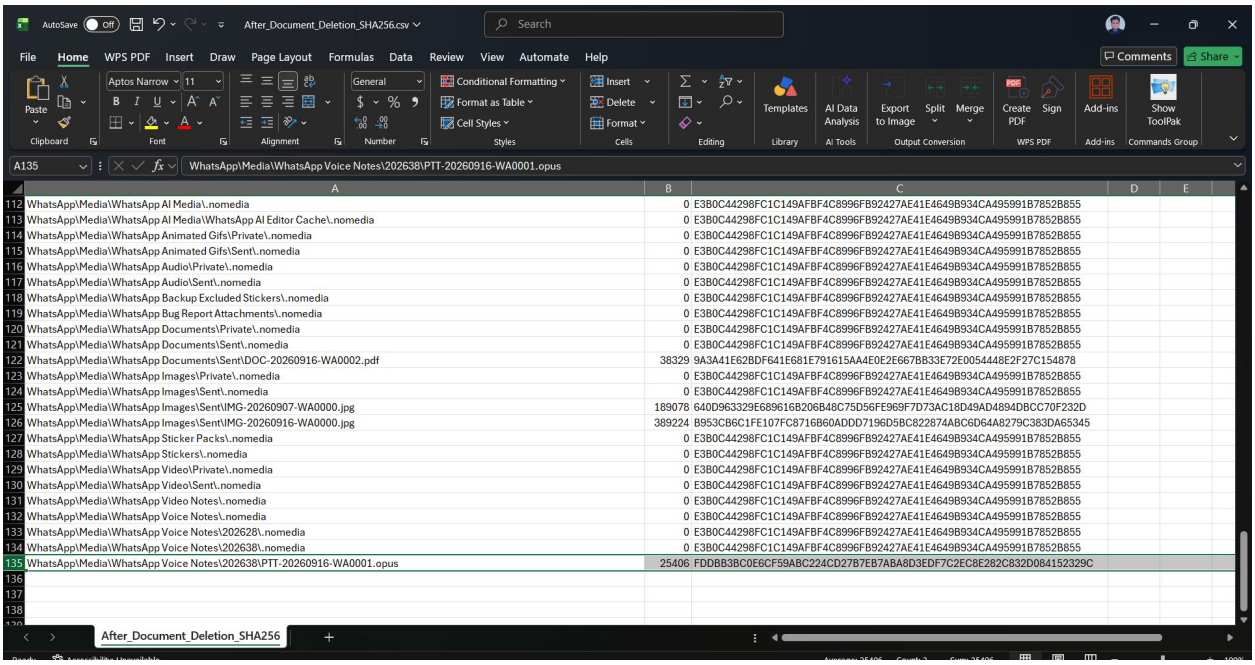
Figure 46: WhatsApp Log File Analysis for Experiment 2

According to Figure 46, WhatsApp Experiment 2 used FORENSIC_WA_TEXT_006, FORENSIC_WA_IMAGE_002.jpg, FORENSIC_WA_VOICE_001 and FORENSIC_WA_DOC_001.pdf. The evidence was sent between 7:15 PM and 7:27 PM. Image was deleted at 7:37 PM, document at 7:44 PM, voice at 7:49 PM and text at 7:55 PM, each using Delete for me.



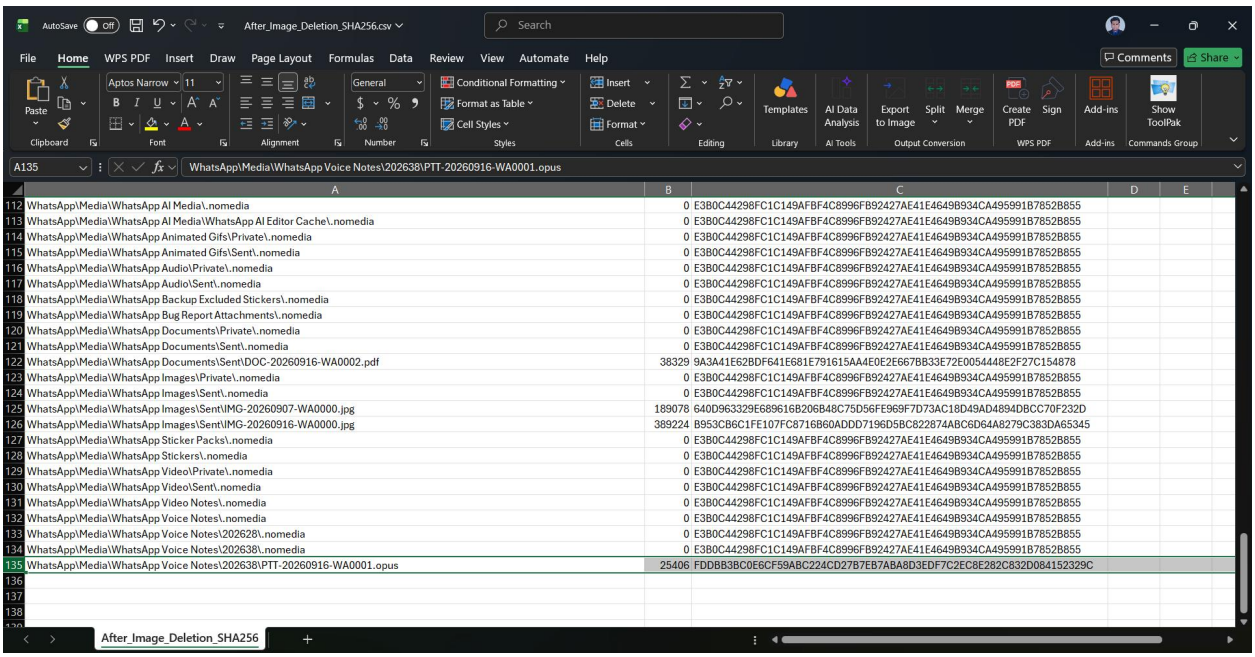
	A	B	C
118	WhatsApp\Media\WhatsApp Bug Report Attachments\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
119	WhatsApp\Media\WhatsApp Documents\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
120	WhatsApp\Media\WhatsApp Documents\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
121	WhatsApp\Media\WhatsApp Documents\Sent\DOC-20260916-WA0002.pdf	38329	9A3A41E62BDF641E681E791615AA4E0E2E667BB33E72E0054448E2F27C154878
122	WhatsApp\Media\WhatsApp Images\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
123	WhatsApp\Media\WhatsApp Images\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
124	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260907-WA0000.jpg	189078	640D963329E689616B206B48C75D56FE969F7D73AC18D49AD4894DBCC70F232D
125	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260916-WA0000.jpg	389224	B953CB6C1FE107FC8716B60ADDD7196D5BC822874ABC6D64A8279C383DA65345
126	WhatsApp\Media\WhatsApp Sticker Packs\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
127	WhatsApp\Media\WhatsApp Stickers\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
128	WhatsApp\Media\WhatsApp Video\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
129	WhatsApp\Media\WhatsApp Video\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
130	WhatsApp\Media\WhatsApp Video Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
131	WhatsApp\Media\WhatsApp Voice Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
132	WhatsApp\Media\WhatsApp Voice Notes\202628\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
133	WhatsApp\Media\WhatsApp Voice Notes\202638\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
134	WhatsApp\Media\WhatsApp Voice Notes\202638\PTT-20260916-WA0001.opus	25406	FDDBB3BC0E6CF59ABC224CD27B7EB7ABA8D3EDF7C2EC8E282C832D084152329C

Figure 47: Before Deletion WhatsApp in Experiment 2



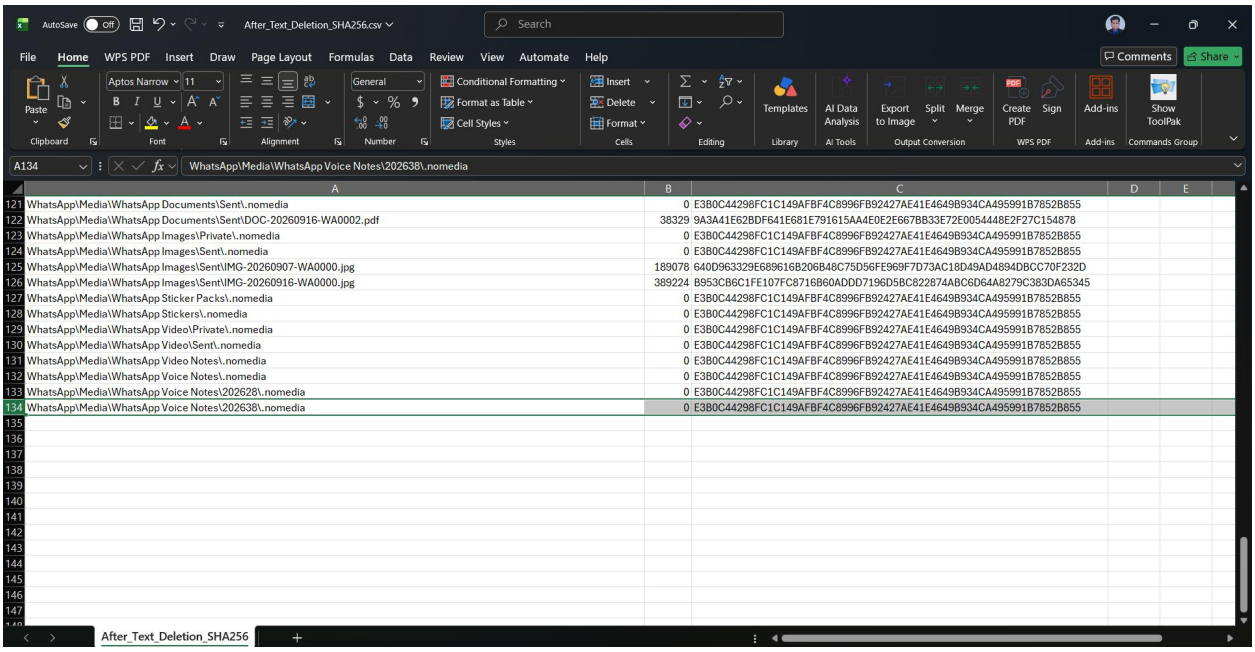
	A	B	C
112	WhatsApp\Media\WhatsApp AI Media\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
113	WhatsApp\Media\WhatsApp AI Media\WhatsApp AI Editor Cache\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
114	WhatsApp\Media\WhatsApp Animated Gifs\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
115	WhatsApp\Media\WhatsApp Animated Gifs\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
116	WhatsApp\Media\WhatsApp Audio\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
117	WhatsApp\Media\WhatsApp Audio\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
118	WhatsApp\Media\WhatsApp Backup Excluded Stickers\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
119	WhatsApp\Media\WhatsApp Bug Report Attachments\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
120	WhatsApp\Media\WhatsApp Documents\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
121	WhatsApp\Media\WhatsApp Documents\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
122	WhatsApp\Media\WhatsApp Documents\Sent\DOC-20260916-WA0002.pdf	38329	9A3A41E62BDF641E681E791615AA4E0E2E667BB33E72E0054448E2F27C154878
123	WhatsApp\Media\WhatsApp Images\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
124	WhatsApp\Media\WhatsApp Images\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
125	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260907-WA0000.jpg	189078	640D963329E689616B206B48C75D56FE969F7D73AC18D49AD4894DBCC70F232D
126	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260916-WA0000.jpg	389224	B953CB6C1FE107FC8716B60ADDD7196D5BC822874ABC6D64A8279C383DA65345
127	WhatsApp\Media\WhatsApp Sticker Packs\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
128	WhatsApp\Media\WhatsApp Stickers\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
129	WhatsApp\Media\WhatsApp Video\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
130	WhatsApp\Media\WhatsApp Video\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
131	WhatsApp\Media\WhatsApp Video Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
132	WhatsApp\Media\WhatsApp Voice Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
133	WhatsApp\Media\WhatsApp Voice Notes\202628\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
134	WhatsApp\Media\WhatsApp Voice Notes\202638\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
135	WhatsApp\Media\WhatsApp Voice Notes\202638\PTT-20260916-WA0001.opus	25406	FDDBB3BC0E6CF59ABC224CD27B7EB7ABA8D3EDF7C2EC8E282C832D084152329C

Figure 48: After Deletion of Document in WhatsApp in Experiment 2



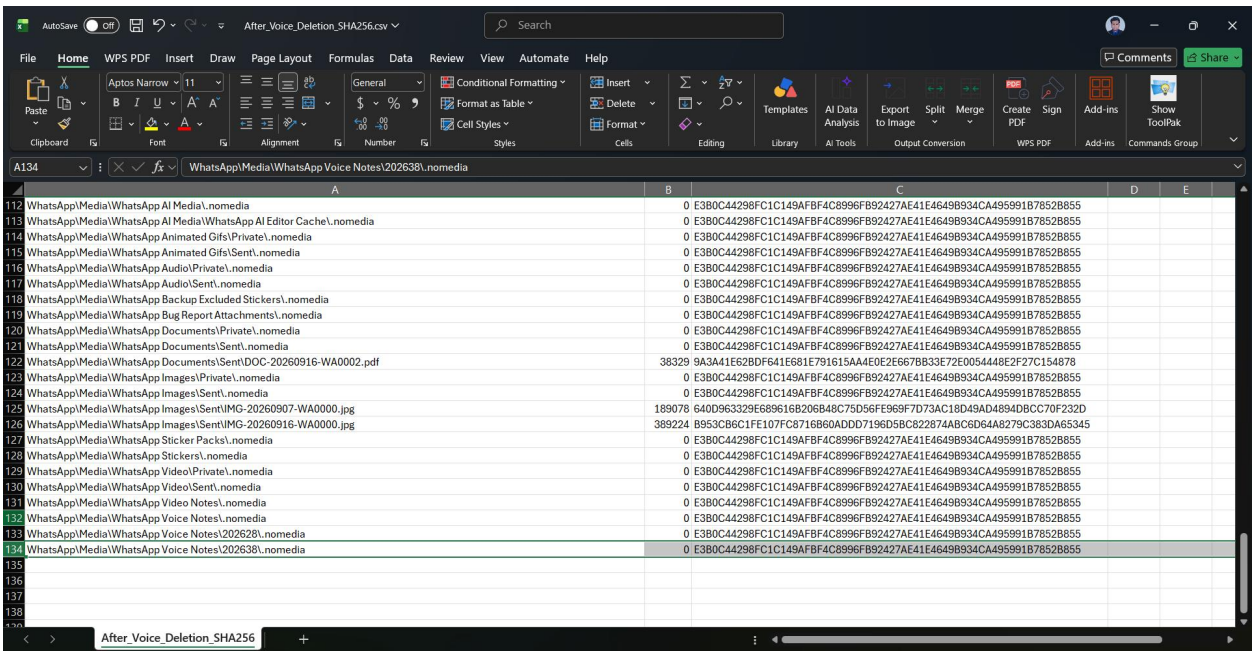
	A	B	C
112	WhatsApp\Media\WhatsApp AI Media\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
113	WhatsApp\Media\WhatsApp AI Media\WhatsApp AI Editor Cache\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
114	WhatsApp\Media\WhatsApp Animated Gifs\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
115	WhatsApp\Media\WhatsApp Animated Gifs\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
116	WhatsApp\Media\WhatsApp Audio\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
117	WhatsApp\Media\WhatsApp Audio\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
118	WhatsApp\Media\WhatsApp Backup Excluded Stickers\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
119	WhatsApp\Media\WhatsApp Bug Report Attachments\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
120	WhatsApp\Media\WhatsApp Documents\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
121	WhatsApp\Media\WhatsApp Documents\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
122	WhatsApp\Media\WhatsApp Documents\Sent\DOC-20260916-WA0002.pdf	38329	9A3A41E62BDF641E681E791615AA4E0E2E667BB33E72E0054448E2F27C154878
123	WhatsApp\Media\WhatsApp Images\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
124	WhatsApp\Media\WhatsApp Images\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
125	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260907-WA0000.jpg	189078	640D963329E689616B206B48C75D56FE969F7D73AC18D49AD4894DBCC70F232D
126	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260916-WA0000.jpg	389224	B953CB6C1FE107FC8716B60ADDD7196D5BC822874ABC6D64A8279C383DA65345
127	WhatsApp\Media\WhatsApp Sticker Packs\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
128	WhatsApp\Media\WhatsApp Stickers\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
129	WhatsApp\Media\WhatsApp Video\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
130	WhatsApp\Media\WhatsApp Video\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
131	WhatsApp\Media\WhatsApp Video Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
132	WhatsApp\Media\WhatsApp Voice Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
133	WhatsApp\Media\WhatsApp Voice Notes\202628\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
134	WhatsApp\Media\WhatsApp Voice Notes\202638\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
135	WhatsApp\Media\WhatsApp Voice Notes\202638\PTT-20260916-WA0001.opus	25406	FDDBB3BC0E6CF59ABC224CD27B7EB7ABA8D3EDF7C2EC8E282C832D084152329C

Figure 49: After Deletion of Image in WhatsApp in Experiment 2



	A	B	C
121	WhatsApp\Media\WhatsApp Documents\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
122	WhatsApp\Media\WhatsApp Documents\Sent\DOC-20260916-WA0002.pdf	38329	9A3A41E62BDF641E681E791615AA4E0E2E667BB33E72E0054448E2F27C154878
123	WhatsApp\Media\WhatsApp Images\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
124	WhatsApp\Media\WhatsApp Images\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
125	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260907-WA0000.jpg	189078	640D963329E689616B206B48C75D56FE969F7D73AC18D49AD4894DBCC70F232D
126	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260916-WA0000.jpg	389224	B953CB6C1FE107FC8716B60ADDD7196D5BC822874ABC6D64A8279C383DA65345
127	WhatsApp\Media\WhatsApp Sticker Packs\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
128	WhatsApp\Media\WhatsApp Stickers\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
129	WhatsApp\Media\WhatsApp Video\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
130	WhatsApp\Media\WhatsApp Video\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
131	WhatsApp\Media\WhatsApp Video Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
132	WhatsApp\Media\WhatsApp Voice Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
133	WhatsApp\Media\WhatsApp Voice Notes\202628\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
134	WhatsApp\Media\WhatsApp Voice Notes\202638\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855

Figure 50: After Deletion of Text in WhatsApp in Experiment 2



	A	B	C
112	WhatsApp\Media\WhatsApp AI Media\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
113	WhatsApp\Media\WhatsApp AI Media\WhatsApp AI Editor Cache\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
114	WhatsApp\Media\WhatsApp Animated Gifs\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
115	WhatsApp\Media\WhatsApp Animated Gifs\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
116	WhatsApp\Media\WhatsApp Audio\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
117	WhatsApp\Media\WhatsApp Audio\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
118	WhatsApp\Media\WhatsApp Backup Excluded Stickers\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
119	WhatsApp\Media\WhatsApp Bug Report Attachments\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
120	WhatsApp\Media\WhatsApp Documents\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
121	WhatsApp\Media\WhatsApp Documents\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
122	WhatsApp\Media\WhatsApp Documents\Sent\DOC-20260916-WA0002.pdf	38329	9A3A41E62BDF641E681E791615AA4E0E2E667BB33E72E0054448E2F27C154878
123	WhatsApp\Media\WhatsApp Images\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
124	WhatsApp\Media\WhatsApp Images\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
125	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260907-WA0000.jpg	189078	640D963329E689616B206B48C75D56FE969F7D73AC18D49AD4894DBCC70F232D
126	WhatsApp\Media\WhatsApp Images\Sent\IMG-20260916-WA0000.jpg	389224	B953CB6C1FE107FC8716B60ADDD7196D5BC822874ABC6D64A8279C383DA65345
127	WhatsApp\Media\WhatsApp Sticker Packs\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
128	WhatsApp\Media\WhatsApp Stickers\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
129	WhatsApp\Media\WhatsApp Video\Private\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
130	WhatsApp\Media\WhatsApp Video\Sent\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
131	WhatsApp\Media\WhatsApp Video Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
132	WhatsApp\Media\WhatsApp Voice Notes\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
133	WhatsApp\Media\WhatsApp Voice Notes\202628\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855
134	WhatsApp\Media\WhatsApp Voice Notes\202638\.nomedia	0	E3B0C44298FC1C149AFBF4C8996FB92427AE41E4649B934CA495991B7852B855

Figure 51: After Deletion of Voice in WhatsApp in Experiment 2

According to Figure 47 to Figure 51 shows that acquisition results at each stage of WhatsApp Experiment 2. Before contained 133 files and 29,129,009 bytes. After Image contained 134 files with the same total bytes. The one addition was WhatsApp\Media\.Statuses\.nomedia, a zero-byte file. After Document remained 134 files with no file-level changes. After Voice fell to 133 files and 29,103,603 bytes. The removed file was WhatsApp\Media\WhatsApp Voice Notes\202638\PTT-20260916-WA0001.opus, 25,406 bytes, SHA-256 FDDBB3BC0E6CF59ABC224CD27B7EB7ABA8D3EDF7C2EC8E282C832D084152329C. The reduction in total bytes matched the removed voice-note size. After Text produced no additional file-level changes.

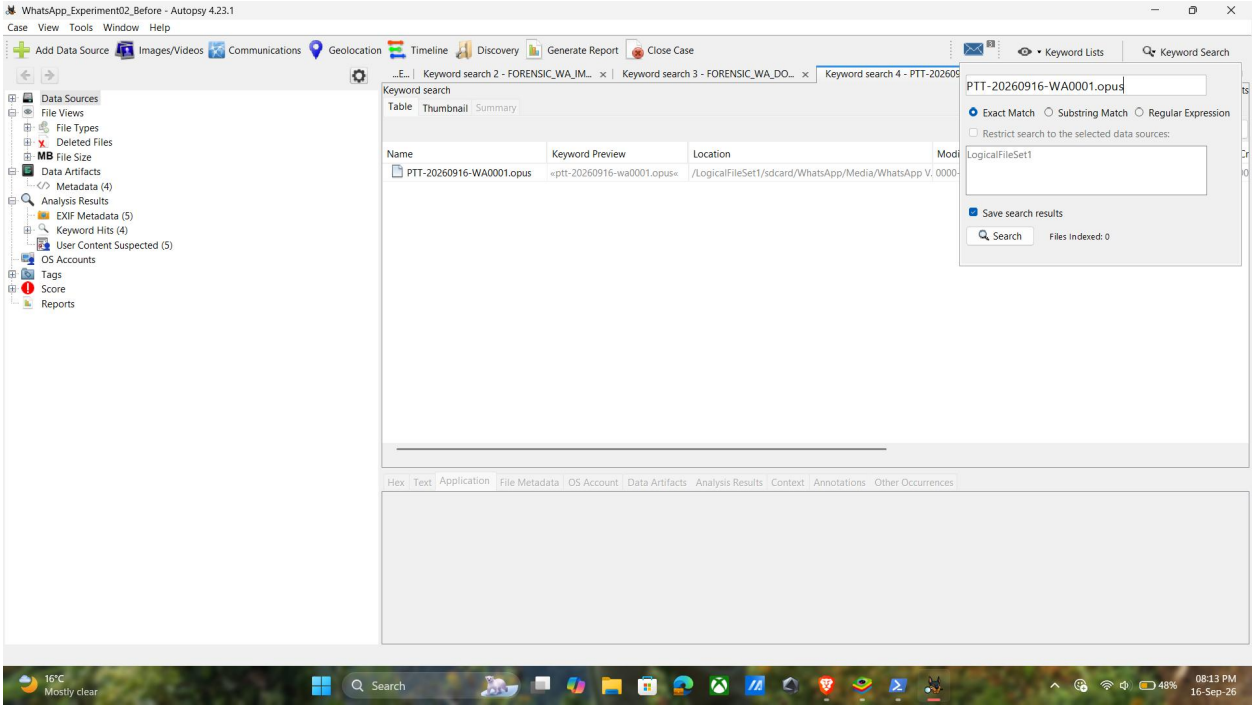


Figure 52: WhatsApp voice evidence before deletion for Experiment 2

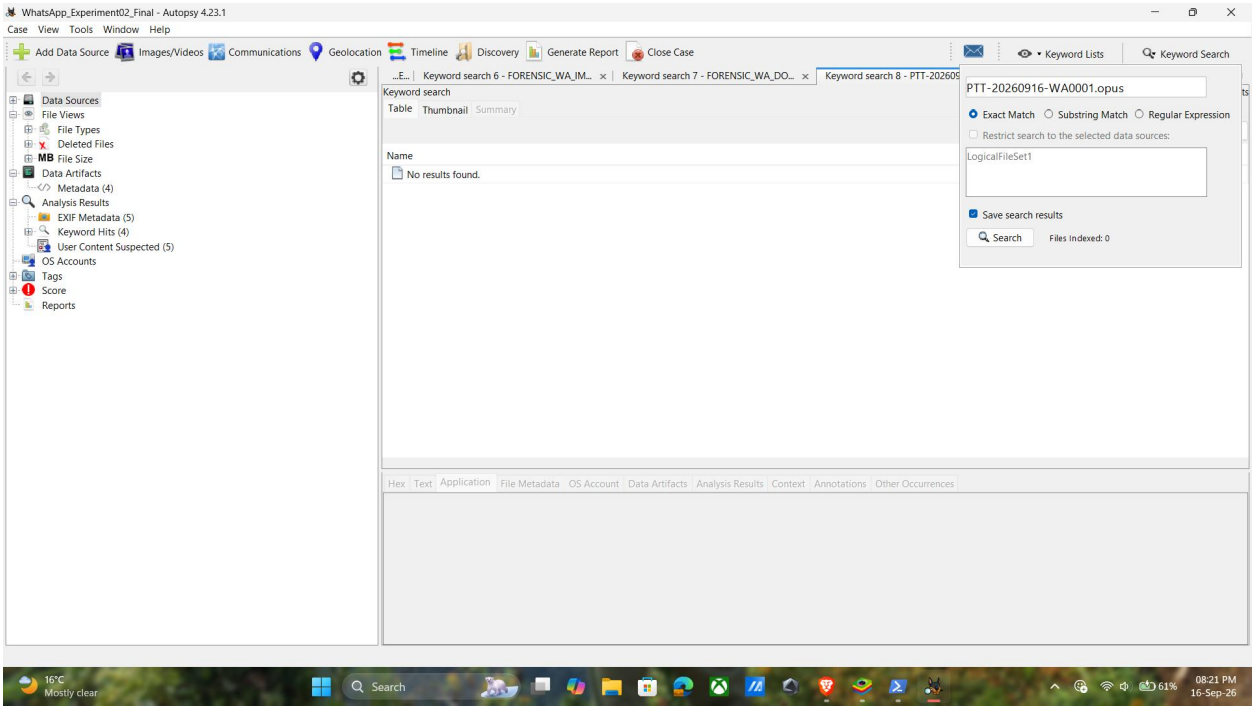
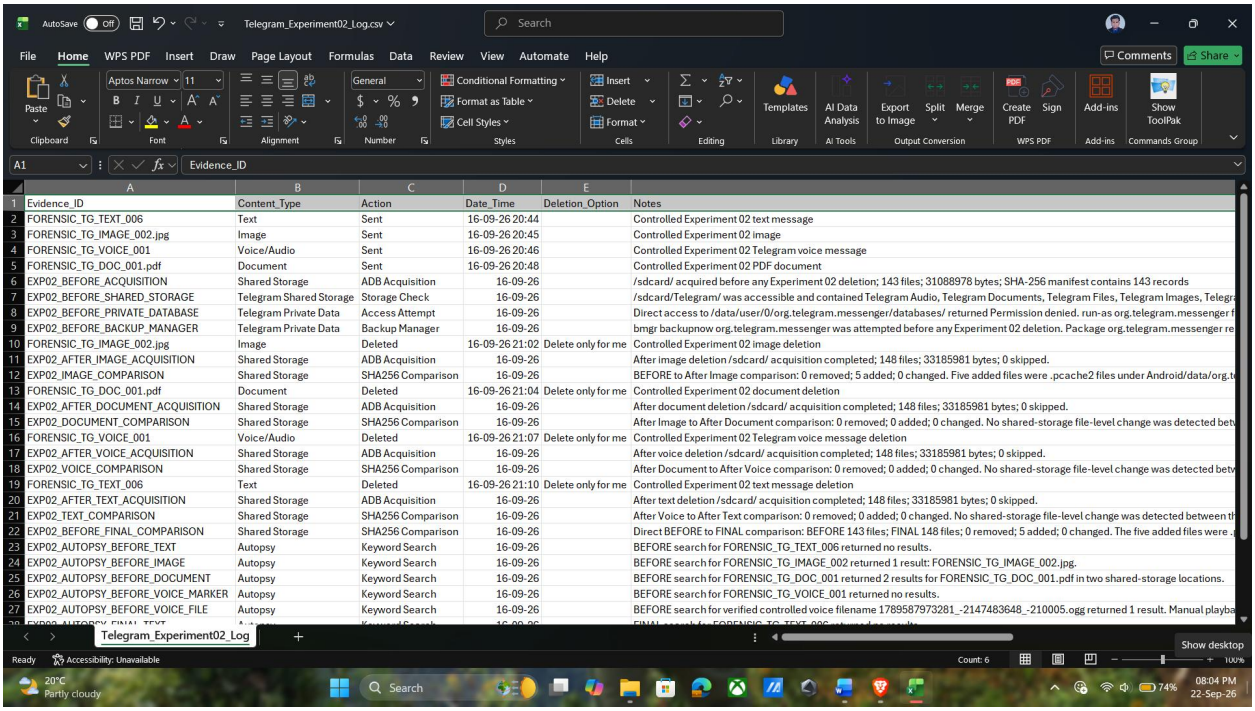


Figure 53: WhatsApp final search after voice deletion for Experiment 2

According to Figure 52 and Figure 53 it shows that, Autopsy found the exact voice filename once in Before and no result in Final, corroborating the manifest comparison. The controlled image remained discoverable, and the document produced three results in both Before and Final. The text marker was not found in either state. The shared-storage msgstore.db.crypt14 remained unchanged from Experiment 1 and was treated as stale rather than as a current readable message.

5.7 Telegram implementation



Evidence_ID	Content_Type	Action	Date_Time	Deletion_Option	Notes
FORENSIC_TG_TEXT_006	Text	Sent	16-09-26 20:44		Controlled Experiment 02 text message
FORENSIC_TG_IMAGE_002.jpg	Image	Sent	16-09-26 20:45		Controlled Experiment 02 image
FORENSIC_TG_VOICE_001	Voice/Audio	Sent	16-09-26 20:46		Controlled Experiment 02 Telegram voice message
FORENSIC_TG_DOC_001.pdf	Document	Sent	16-09-26 20:48		Controlled Experiment 02 PDF document
EXP02_BEFORE_ACQUISITION	Shared Storage	ADB Acquisition	16-09-26		/sdcard/ acquired before any Experiment 02 deletion; 143 files; 31088978 bytes; SHA-256 manifest contains 143 records
EXP02_BEFORE_SHARED_STORAGE	Telegram Shared Storage	Storage Check	16-09-26		/sdcard/Telegram/ was accessible and contained Telegram Audio, Telegram Documents, Telegram Files, Telegram Images, Telegra
EXP02_BEFORE_PRIVATE_DATABASE	Telegram Private Data	Access Attempt	16-09-26		Direct access to /data/user/0/org.telegram.messenger/databases/ returned Permission denied. run-as org.telegram.messenger f
EXP02_BEFORE_BACKUP_MANAGER	Telegram Private Data	Backup Manager	16-09-26		bmgr backupnow org.telegram.messenger was attempted before any Experiment 02 deletion. Package org.telegram.messenger re
FORENSIC_TG_IMAGE_002.jpg	Image	Deleted	16-09-26 21:02	Delete only for me	Controlled Experiment 02 image deletion
EXP02_AFTER_IMAGE_ACQUISITION	Shared Storage	ADB Acquisition	16-09-26		After image deletion /sdcard/ acquisition completed; 148 files; 33185981 bytes; 0 skipped.
EXP02_IMAGE_COMPARISON	Shared Storage	SHA256 Comparison	16-09-26		BEFORE to After Image comparison: 0 removed; 5 added; 0 changed. Five added files were .pcache2 files under Android/data/org.te
FORENSIC_TG_DOC_001.pdf	Document	Deleted	16-09-26 21:04	Delete only for me	Controlled Experiment 02 document deletion
EXP02_AFTER_DOCUMENT_ACQUISITION	Shared Storage	ADB Acquisition	16-09-26		After document deletion /sdcard/ acquisition completed; 148 files; 33185981 bytes; 0 skipped.
EXP02_DOCUMENT_COMPARISON	Shared Storage	SHA256 Comparison	16-09-26		After Image to After Document comparison: 0 removed; 0 added; 0 changed. No shared-storage file-level change was detected betw
FORENSIC_TG_VOICE_001	Voice/Audio	Deleted	16-09-26 21:07	Delete only for me	Controlled Experiment 02 Telegram voice message deletion
EXP02_AFTER_VOICE_ACQUISITION	Shared Storage	ADB Acquisition	16-09-26		After voice deletion /sdcard/ acquisition completed; 148 files; 33185981 bytes; 0 skipped.
EXP02_VOICE_COMPARISON	Shared Storage	SHA256 Comparison	16-09-26		After Document to After Voice comparison: 0 removed; 0 added; 0 changed. No shared-storage file-level change was detected betw
FORENSIC_TG_TEXT_006	Text	Deleted	16-09-26 21:10	Delete only for me	Controlled Experiment 02 text message deletion
EXP02_AFTER_TEXT_ACQUISITION	Shared Storage	ADB Acquisition	16-09-26		After text deletion /sdcard/ acquisition completed; 148 files; 33185981 bytes; 0 skipped.
EXP02_TEXT_COMPARISON	Shared Storage	SHA256 Comparison	16-09-26		After Voice to After Text comparison: 0 removed; 0 added; 0 changed. No shared-storage file-level change was detected between th
EXP02_BEFORE_FINAL_COMPARISON	Shared Storage	SHA256 Comparison	16-09-26		Direct BEFORE to FINAL comparison: BEFORE 143 files; FINAL 148 files; 0 removed; 5 added; 0 changed. The five added files were .p
EXP02_AUTOPSY_BEFORE_TEXT	Autopsy	Keyword Search	16-09-26		BEFORE search for FORENSIC_TG_TEXT_006 returned no results.
EXP02_AUTOPSY_BEFORE_IMAGE	Autopsy	Keyword Search	16-09-26		BEFORE search for FORENSIC_TG_IMAGE_002 returned 1 result: FORENSIC_TG_IMAGE_002.jpg.
EXP02_AUTOPSY_BEFORE_DOCUMENT	Autopsy	Keyword Search	16-09-26		BEFORE search for FORENSIC_TG_DOC_001 returned 2 results for FORENSIC_TG_DOC_001.pdf in two shared-storage locations.
EXP02_AUTOPSY_BEFORE_VOICE_MARKER	Autopsy	Keyword Search	16-09-26		BEFORE search for FORENSIC_TG_VOICE_001 returned no results.
EXP02_AUTOPSY_BEFORE_VOICE_FILE	Autopsy	Keyword Search	16-09-26		BEFORE search for verified controlled voice filename 1789587973281_-2147483648_-210005.ogg returned 1 result. Manual playba

Figure 54: Telegram Log File Analysis of Experiment 2

Figure 54 records the controlled activity in Telegram Experiment 2. Telegram Experiment 2 used FORENSIC_TG_TEXT_006, FORENSIC_TG_IMAGE_002.jpg, FORENSIC_TG_VOICE_001 and FORENSIC_TG_DOC_001.pdf. The evidence was sent between 8:44 PM and 8:48 PM on 16 September. The image was deleted at 9:02 PM, document at 9:04 PM, voice at 9:07 PM and text at 9:10 PM using Delete only for me.

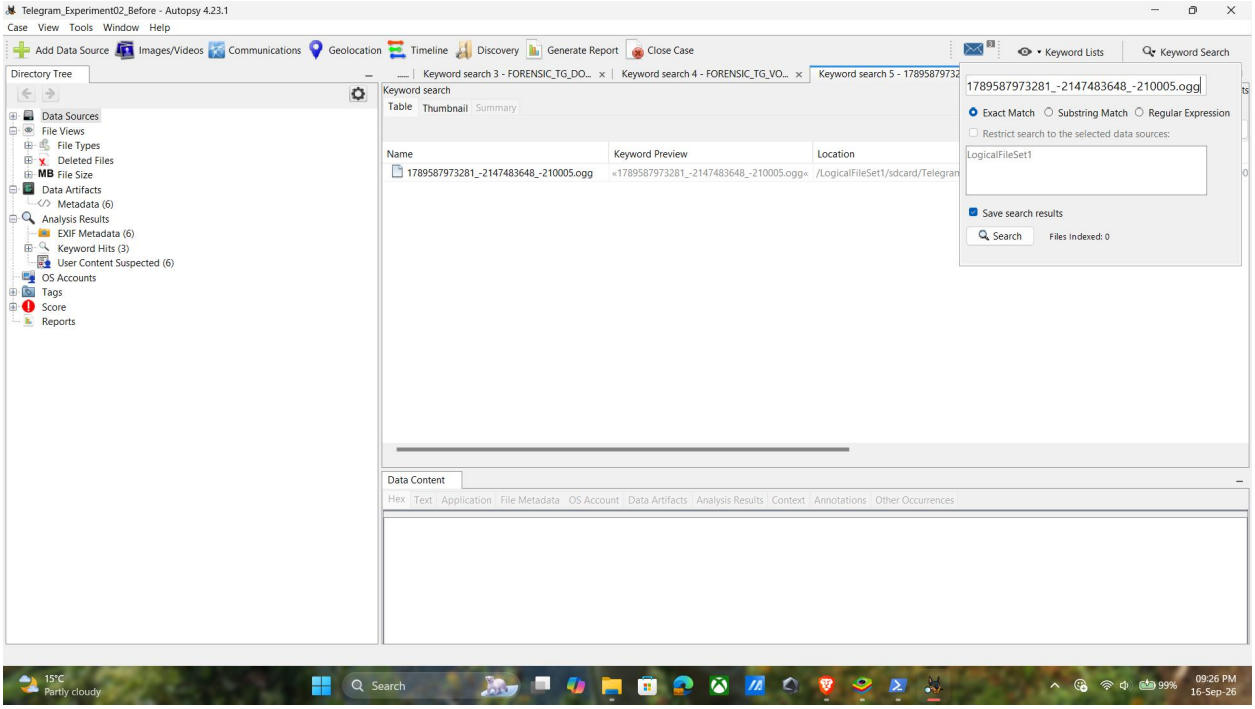


Figure 55: Telegram voice evidence Before deletion for Experiment 2

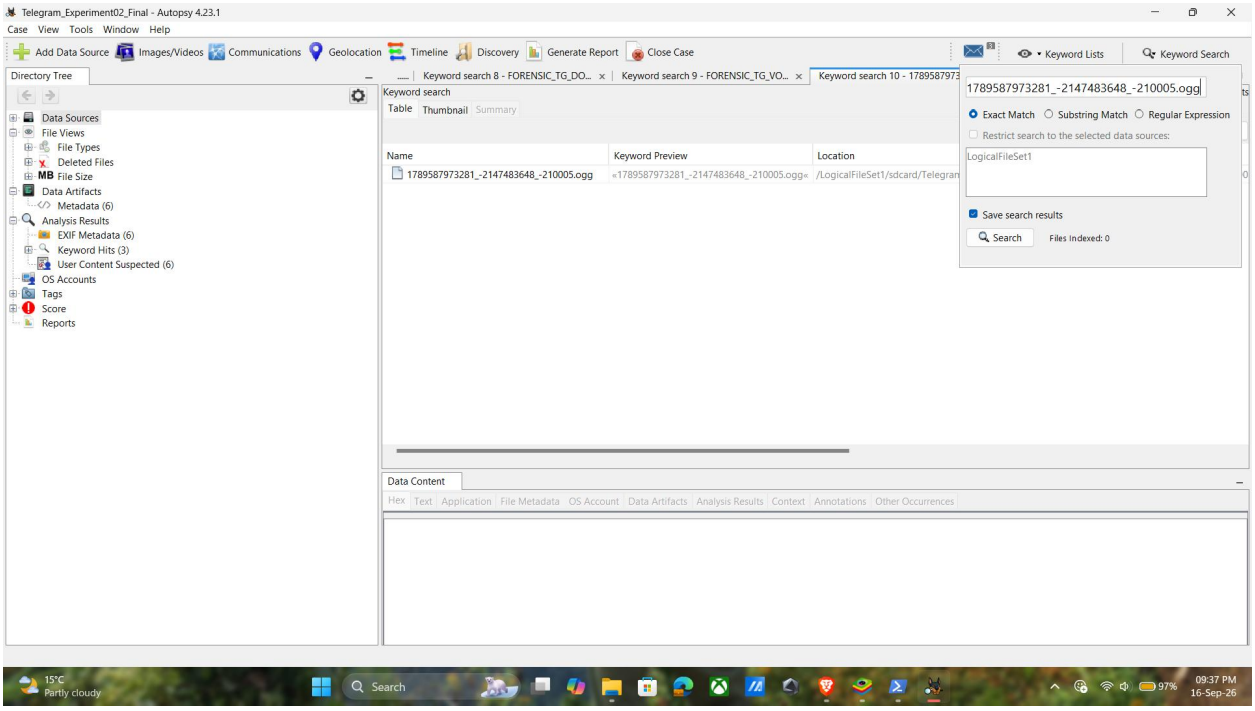


Figure 56: Telegram voice evidence After deletion for Experiment 2

The following Figure 55 and Figure 56 shows the Before and After found the voice file. The Before /sdcard/ acquisition contained 143 files and 31,088,978 bytes. After Image contained 148 files and 33,185,981 bytes, with 0 removed, 5 added and 0 changed. The additions were .pcache2 artifacts under Android/data/org.telegram.messenger/cache/acache/. After Document, After Voice and After Text each remained at 148 files and 33,185,981 bytes with 0 removed, 0 added

and 0 changed relative to the previous stage. Direct Before then Final comparison was 143 then 148 files with 0 removed, 5 added and 0 changed.

The controlled voice marker was not present in the filename, so the Before Telegram Audio directory was examined. Manual playback confirmed that 1789587973281_-2147483648_-210005.ogg contained the controlled voice phrase. The file was 65,525 bytes in Before and Final and had identical SHA-256 487E9FFD9D54554F5A05CD05738CD447EBA114E7FB65559E3AB44E38E1523D2D. Autopsy found the verified filename once in both cases.

Autopsy also returned no result for the text marker, one result for the image in each state, and two results for the PDF in each state. Direct private database access returned Permission denied, run-as failed, and Backup Manager reported package and overall Success. No readable private database or exported backup was acquired, so the success result was documented without treating it as database recovery.

5.8 Implementation summary

Application	Image after deletion	Document after deletion	Voice result	Text marker
Google Chat	Persisted	Persisted	No corresponding shared-storage voice file identified	Not found Before or Final
WhatsApp	Persisted	Persisted	Verified .opus removed after voice deletion	Not found Before or Final
Telegram	Persisted	Persisted	Verified .ogg persisted unchanged	Not found Before or Final

Table 8: Main Experiment 2 implementation observations

The implementation produced two complementary evidence sets. Experiment 1 established the initial text-deletion comparison and exposed application-specific differences in database accessibility. Experiment 2 provided a more detailed multimedia comparison and isolated each deletion event through sequential acquisition.

The evidence collected during implementation provides the basis for the comparison and interpretation presented in Chapter 6. The next chapter evaluates these findings while considering the limitations of the acquisition methods used.

Chapter 6: Evaluation of Results

This chapter discusses the findings from the two experiments and what they show about the evidence left behind after deletion. It compares the results from Google Chat, WhatsApp and Telegram, highlights the challenges faced during the investigation, and explains how the findings relate to the research objectives.

6.1 Evaluation framework

The results are evaluated against the study's practical objectives rather than against an assumption that deleted messages must be recoverable. Three dimensions are used: evidence availability, evidence change after deletion, and interpretive strength. Evidence availability asks what was actually acquired and readable. Evidence change asks whether controlled deletion corresponded with observable file or database changes. Interpretive strength asks whether an observation supports original-content recovery, a deletion indicator, file persistence, file removal or only a limitation.

This framework is necessary because the applications produced different forms of evidence. Google Chat Experiment 1 produced a readable SQLite database with a deletion-related tombstone. WhatsApp preserved encrypted database material but not a usable decryption key. Telegram's Backup Manager operation succeeded without yielding a readable message database. Experiment 2 then showed different multimedia behavior in shared storage. Comparing these results as if they were equivalent “recoveries” would obscure the actual forensic value of each observation.

The evaluation therefore separates positive evidence from absence of evidence. A positive hash match can demonstrate that a known file remained unchanged. A removed relative path can demonstrate that a file present in one acquisition was absent from the next. A negative keyword search cannot demonstrate permanent deletion when the marker was also absent in BEFORE or when private storage was not acquired.

6.2 Google Chat Evaluation

Google Chat produced the strongest database-level evidence in Experiment 1 because dynamite.db was readable. The tombstone record contained a populated delete_time, is_tombstone=1 and empty text_body. These fields are consistent with a deletion-related state in the acquired database. However, the original deleted marker was not recovered

from that record, and the available evidence did not independently establish a definitive mapping between the tombstone identifier and FORENSIC_GCHAT_001. The correct evaluation is therefore deletion-related artifact identified, original text not recovered.

The Experiment 1 shared-storage comparison was comparatively uninformative for text deletion. Both manifests contained 77 records with matching corresponding paths and hashes, and Autopsy did not find the deleted marker in either BEFORE or AFTER. Because the marker was not positively observed in BEFORE shared storage, its absence in AFTER cannot be attributed to deletion.

Experiment 2 strengthened the Google Chat evaluation for multimedia. All five acquisitions contained 125 files and 28,169,545 bytes, and every comparison produced 0 removed, 0 added and 0 changed. The controlled image and PDF remained discoverable in Autopsy, and their final hashes matched the originals. This is strong evidence that those particular shared-storage copies persisted byte-for-byte through the deletion sequence. The text marker remained absent in both states, while no corresponding voice file was identified. Thus, Google Chat showed stable shared-storage multimedia evidence under the tested conditions, but the private message state remained largely outside the Experiment 2 acquisition scope.

6.3 WhatsApp Evaluation

WhatsApp Experiment 1 demonstrates the limitation created by encrypted evidence. The preserved msgstore.db.crypt14 had a stable SHA-256 value, but without the necessary key it could not be examined as plaintext SQLite. The package-specific Backup Manager operation was rejected. Private access and run-as also failed. Consequently, the experiment could preserve and characterize the encrypted file but could not compare readable message records before and after deletion.

The Experiment 1 shared-storage manifest changed from 80 to 88 files, but there were no removals or changed corresponding files. The eight additions could not be attributed automatically to the deletion. Autopsy did not find FORENSIC_WA_001 in either state. This means the first experiment did not provide direct shared-storage evidence of the text deletion.

Experiment 2 produced a substantially stronger file-level result. The controlled voice-note file PTT-20260916-WA0001.opus was present before voice deletion and was the

only removed file in the After Document then After Voice comparison. Its size was 25,406 bytes, and the total acquisition size decreased by exactly the same amount. Autopsy independently found the filename in Before and not in Final. These mutually consistent observations support a strong conclusion that the acquired shared-storage copy disappeared during the interval containing the controlled voice deletion.

By contrast, image and document evidence remained discoverable after their local deletion, and text deletion produced no detectable shared-storage file-level change. The zero-byte .Statuses\.nomedia addition occurred earlier but cannot be assigned causally to image deletion. WhatsApp therefore demonstrates content-type-dependent behavior within the tested shared-storage environment.

6.4 Telegram Evaluation

Telegram Experiment 1 provided no readable private database despite successful Backup Manager processing. This is a useful methodological result because it shows why process success and evidence usability must be distinguished. The Before-to-After shared-storage comparison recorded 19 additions and no changes. The additions were cache or animation-related files, and no evidence established that the controlled text deletion caused them. Autopsy did not find the target marker in either state.

Experiment 2 provided stronger positive multimedia controls. The controlled image was found once in both Before and Final, the PDF twice in both states, and the verified voice .ogg once in both states. The voice file was particularly informative because manual playback associated the application-generated filename with the controlled recording. Its size and SHA-256 were identical in Before and Final. This supports byte-for-byte persistence of the acquired shared-storage voice artifact after Delete only for me.

The direct Before then Final comparison showed 0 removed, 5 added and 0 changed. The five additions first appeared between BEFORE and After Image and were .pcache2 files. Because they persisted through later stages and no controlled file was removed, they are best interpreted as temporally associated cache activity rather than proven deletion artifacts. The text marker was not found in either Autopsy case, so the experiment cannot evaluate text persistence through shared-storage keyword search.

6.5 Cross-Platform comparison

Measure	Google Chat	WhatsApp	Telegram
Before then Final file count	125 → 125	133 → 133	143 → 148
Removed files	0	1 controlled voice file	0
Added files	0	1 .nomedia	5 .pcache2 cache files
Changed files	0	0	0
Controlled image	Persisted	Persisted	Persisted
Controlled document	Persisted	Persisted	Persisted
Controlled voice	No corresponding shared-storage file identified	Removed	Persisted unchanged
Readable private/message DB evidence	Readable Google Chat DB available in Exp. 1	Encrypted crypt14 only	No readable private message DB acquired

Table 9: Cross-platform comparison of the main findings

Table 9 compares the main findings from Google Chat, WhatsApp and Telegram. The results show that the controlled image and document remained in the acquired shared storage for all three applications, even after they were deleted from the chats. However, the voice files showed different results. The WhatsApp voice file was removed from the acquired shared storage, while the Telegram voice file remained unchanged. No corresponding shared-storage voice file was identified for Google Chat. The table also shows that access to private message databases was different for each application. These findings highlight why the results need to be examined separately for each application and why deleting content from a chat does not always mean that its associated files have been removed from accessible storage.

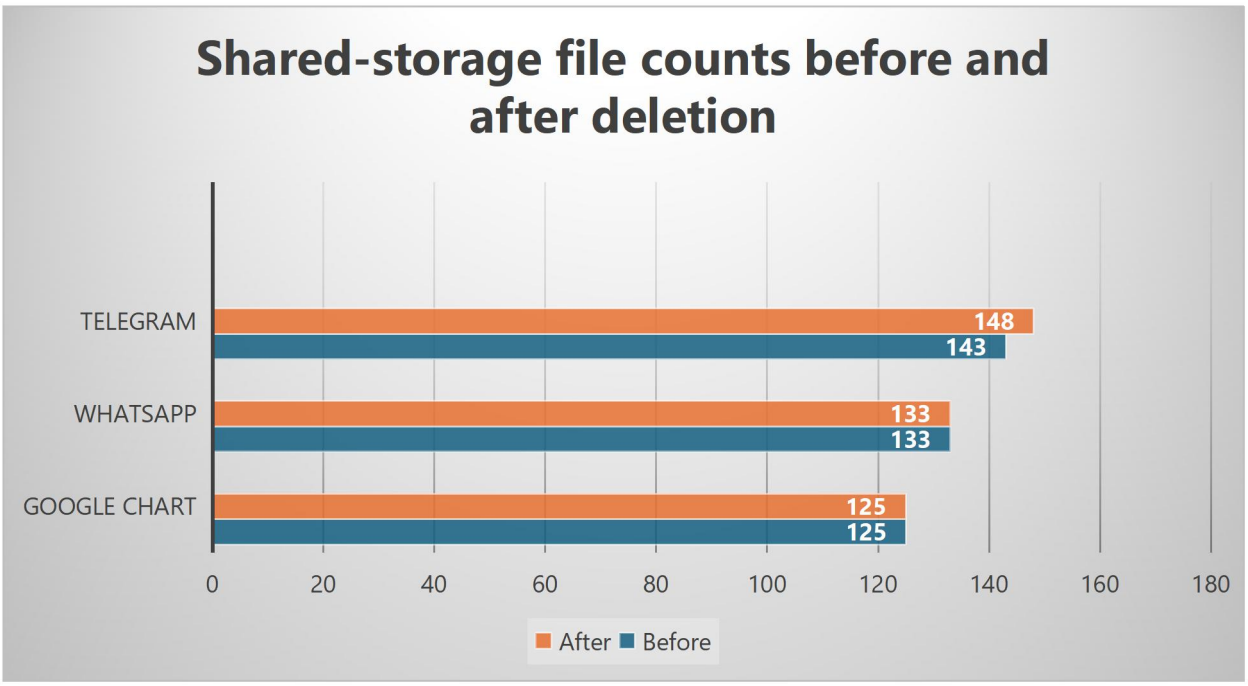


Figure 57: Cross-platform Experiment 2 comparison

Figure 57 compares the total number of files collected before and after the deletion stages in Experiment 2. Google Chat remained at 125 files, while WhatsApp remained at 133 files because one controlled voice file was removed and one .nomedia file was added. Telegram increased from 143 to 148 files due to five additional cache files. The chart shows the overall file counts, while Table 9 provides further details about the individual file changes and the controlled image, document and voice evidence.

6.6 Research objectives and interpretation

The project achieved its central objective of identifying and comparing accessible forensic artefacts from three Android messaging applications. It also demonstrated that the ability to analyse an artefact depends heavily on where it is stored and whether the acquisition method can access it. Google Chat allowed database-level interpretation in Experiment 1, whereas WhatsApp and Telegram were more restricted at that level. Experiment 2 provided a more balanced file-level comparison because the same staged deletion design was applied across all three applications.

The results should not be read as a ranking of the applications or as a claim that one application permanently retains or securely removes all deleted content. They describe the behaviour observed in this BlueStacks-based environment, with these app versions, controlled test data and acquisition methods.

6.7 Limitations

Limitation	Effect on interpretation
Non-rooted emulated environment	Private application storage was not equally accessible.
WhatsApp encryption	The available crypt14 database could not be examined as readable SQLite evidence.
Telegram private DB unavailable	Equivalent database-level comparison with Google Chat was not possible.
Shared-storage focus in Experiment 2	Findings do not describe all private, cloud or server-side artefacts.
Single controlled environment	Results should not automatically be generalised to every Android device/version.

Table 10: Main limitations of the study

Table 10 shows the main limitations of this study and how they affected the results. The experiments were carried out using a non-rooted BlueStacks emulator. This made it easier to control the testing process, but it also limited access to some private application files.

The WhatsApp database was encrypted, and the required decryption key was not available. Because of this, I could not examine its message records using DB Browser for SQLite. I also could not obtain a readable private message database from Telegram. This meant that I could not compare the internal databases of all three applications in the same way.

In Experiment 2, I mainly examined files collected from accessible shared storage. I compared the files before and after deletion using their names, sizes and SHA-256 hashes. However, this method did not provide access to all private files or data stored in the cloud. Therefore, if a file was not found in the collected evidence, it did not necessarily mean that the file had been permanently deleted.

Finally, the experiments were carried out in one emulator using specific application versions. The results may be different on a physical Android phone or when using other application versions. For this reason, the findings should be understood within the conditions of this study.

Chapter 7: Conclusion and Recommendations

This research examined Google Chat, WhatsApp and Telegram through two controlled Android messaging-forensics experiments in a BlueStacks environment. The work focused on what could be acquired and interpreted through ADB-accessible shared storage, Autopsy examination, SHA-256 comparison and, where possible, SQLite analysis. Experiment 1 tested controlled text-message deletion. Experiment 2 extended the investigation to image, document, voice/audio and text evidence using sequential deletion and acquisition.

The study's central conclusion is not that one application is universally more or less recoverable. Instead, it demonstrates that forensic evidence availability depends on application storage behavior, content type, encryption, permissions and the acquisition method used. The three applications exposed materially different evidence under the same broad workflow.

The research also demonstrates the importance of terminology. A backup command reporting success is not automatically a readable backup. An encrypted database file is not automatically analyzable. A tombstone is not the same as the original deleted message. A negative search result is not proof of permanent erasure. These distinctions shaped both the findings and the recommendations.

7.1 Conclusion for Google Chat

Google Chat provided the most significant database-level deletion artifact in Experiment 1. The readable dynamite.db contained a tombstone with is_tombstone=1, a populated delete_time and empty text_body. This supports a finding that deletion-related state existed in the acquired database. However, the original deleted text was not recovered, and the available record did not independently prove a definitive mapping to FORENSIC_GCHAT_001. The conclusion must therefore remain narrower than “deleted message recovered.”

The Experiment 1 shared-storage evidence did not show a deletion-linked change. BEFORE and AFTER manifests each contained 77 corresponding records with matching hashes, and the controlled deleted marker was not found in either Autopsy search. The database evidence was therefore more informative than the shared-storage evidence for text deletion.

Experiment 2 produced a different but complementary result. All five Google Chat shared-storage acquisitions contained 125 files and the same total byte count, with 0 removed, 0 added and 0 changed at every stage. The controlled image and PDF remained discoverable, and their final hashes matched the originals. Within the acquired shared storage, those files therefore persisted byte-for-byte after deletion. No equivalent voice file was identified, and the text marker remained undiscoverable. Google Chat consequently illustrates both the value of readable database evidence and the limitation of shared-storage-only text analysis.

7.2 Conclusion for WhatsApp

WhatsApp's principal Experiment 1 limitation was encryption and private-data access. The preserved msgstore.db.crypt14 was stable and could be hashed, but no decryption key was acquired. The package-specific Backup Manager request was rejected, direct private access was denied and run-as failed. As a result, no readable message database was available for a database-level comparison equivalent to Google Chat.

Experiment 2 nevertheless produced one of the study's strongest shared-storage findings. The controlled voice-note .opus was present before voice deletion, then removed in the immediately following acquisition. Its 25,406-byte size exactly matched the reduction in total acquired bytes, and Autopsy changed from one exact filename result in BEFORE to no result in FINAL. This convergence of evidence supports a deletion-linked disappearance of that shared-storage voice-note copy.

Image and document artifacts remained discoverable after Delete for me, and text deletion produced no detectable file-level change. The .Statuses\.nomedia addition cannot be assigned causally to deletion. The unchanged .crypt14 was treated as stale and encrypted rather than as a current message snapshot. WhatsApp therefore demonstrates why media evidence can sometimes provide stronger observable deletion behavior than an inaccessible or encrypted message database.

7.3 Conclusion for Telegram

Telegram Experiment 1 did not yield a readable private message database. Backup Manager reported package Success, but the outcome did not produce a readable exported database suitable for SQLite examination. Shared-storage comparison showed

additions but no removals or changed files, and the deleted text marker was not found in either Autopsy state.

Experiment 2 provided clearer multimedia evidence. The controlled image remained discoverable, the PDF remained discoverable in two shared-storage locations, and the verified controlled voice .ogg remained present after Delete only for me. Manual playback established the association between the application-generated .ogg filename and the controlled recording. Before and Final copies were both 65,525 bytes and shared the same SHA-256, providing strong evidence of byte-for-byte persistence of that shared-storage artifact.

Five .pcache2 files appeared between Before and After Image, but the experiment cannot establish that deletion created them. Later deletion stages produced no file-level changes. The text marker remained absent from both Autopsy cases. Telegram therefore demonstrates persistent shared-storage multimedia evidence under the tested local-deletion workflow while also showing the limitation created by inaccessible private message data.

7.4 Overall Contribution and Cross-Platform Conclusion

Contribution	**Practical value**
Controlled two-experiment design	Provides a repeatable way to compare text and multimedia deletion.
Cross-platform evidence comparison	Shows different observable storage behaviour across three messaging apps.
Integrity checking	Uses SHA-256 manifests to support evidence comparison and traceability.
Documented negative results	Records access restrictions and unavailable databases rather than hiding them.
Staged deletion in Experiment 2	Makes temporal interpretation of file changes clearer.

Table 11: Main Contributions of the Project

The study therefore shows that application-interface deletion should not be assumed to have one forensic meaning. It may correspond with removal of a shared-storage file, persistence of a shared-storage file, a database tombstone, cache activity or no observable change in the acquired evidence. The correct interpretation depends on the evidence source and acquisition scope

7.5 Recommendations

Future research could carry out the same experiments on a physical Android phone to see whether the findings from BlueStacks are also found on a real device. BlueStacks provided a controlled environment for this study, but a physical phone may store and manage application data differently. Repeating the experiments on different phones and Android versions would help show whether the results are consistent across different devices.

Before carrying out experiments on a physical Android phone, future researchers should prepare a separate experimental plan rather than assume that the BlueStacks procedures will work in the same way. The plan should include selecting suitable forensic acquisition methods, preparing controlled test data and documenting each stage of the investigation. A pilot test could help identify technical problems and set realistic time requirements before the main experiments begin. Researchers should also discuss the proposed work with their supervisor and confirm whether additional ethical approval is required. The findings from the physical phone could then be compared with the BlueStacks results to identify similarities and differences.

Another area for future research is gaining more complete access to application data through authorised forensic methods. In this study, access to private application storage was limited. The WhatsApp database was encrypted, and a readable Telegram message database could not be obtained. Future studies could use an authorised test environment to collect readable databases before and after deletion. This would allow researchers to compare message records in more detail and examine whether any information remains after deletion.

The experiments could also be repeated several times for each type of content. Starting with a fresh emulator environment or resetting the test device between trials would help reduce the influence of earlier application activity. Repeating the tests would make it easier to understand whether the same file changes happen each time or whether some changes are caused by normal background activity.

Future research should also confirm that each test message or media file can be found in the acquired evidence before it is deleted. In this study, the controlled text markers were not found in the shared-storage evidence before deletion. For this reason, their absence after deletion could not be directly linked to the deletion action. Finding the test data in

the first acquisition would provide a clearer starting point for comparing the evidence collected after deletion.

It would also be helpful to keep a more detailed record of when each experimental step takes place. This should include the time a message is sent, the time it is deleted and the start and end times of each acquisition. Researchers should also record the network connection and whether the application is open or running in the background. These details may help explain why certain files appear, change or disappear during the experiments.

Finally, future studies should continue to organise and preserve the collected evidence carefully. Evidence from each acquisition stage should be stored in a separate folder, and SHA-256 hashes should be calculated to check file integrity. Command logs, screenshots and file comparison results should also be kept so that another examiner can follow the process. When reporting the findings, researchers should explain the difference between recovering the original deleted content, identifying a deletion-related record and finding a file that remains after deletion. This would help ensure that the conclusions are clear and supported by the evidence collected.

7.6 Final Conclusion

This study examined what forensic evidence could be found after messages and multimedia content were deleted from Google Chat, WhatsApp and Telegram in a BlueStacks Android environment. Through the two experiments, I was able to compare the evidence collected before and after deletion and understand some of the challenges involved in examining messaging applications.

The findings showed that deleting something from an application does not always mean that its associated files disappear from accessible storage. At the same time, finding a remaining file or a deletion-related record does not mean that the original deleted message has been recovered. This was an important distinction throughout the investigation, particularly when some application data could not be accessed or examined.

Although the study had limitations, the experiments provided a clear way to collect, preserve and compare the available evidence. Keeping separate acquisition folders, checking SHA-256 hashes and recording the results helped me understand the changes

observed at each stage. Overall, this project shows why forensic findings must be based on the evidence actually collected rather than assumptions about what deletion means. Future research using physical Android phones and more complete acquisition methods could help develop these findings further.

Chapter 8: Reflection

When I started this project, I was mainly focused on completing the experiments and producing a dissertation that explained my findings. My research examined what happened to forensic evidence after messages and media files were deleted from Google Chat, WhatsApp and Telegram in an Android emulator. I expected the practical work to be challenging, but I did not realise how much planning, organisation and patience the project would require.

As the work progressed, I learned more about myself as a student. I became more confident in working independently, but I also recognised areas where I could improve, particularly in managing my time, keeping records and reviewing my work. This chapter reflects on those experiences and how I would approach a similar project in the future.

8.1 Managing My Time

Time management was one of my main challenges. Initially, I planned to complete the experiments before concentrating on the dissertation. I thought writing would be easier once I had collected the results. However, preparing test data, acquiring evidence, checking files and recording observations took longer than I expected.

When something did not work as planned, I sometimes spent too much time trying to solve the problem. I wanted to complete each task before moving on, but this left less time for other activities. Working on one application at a time helped me stay organised, although I could have set clearer limits on how long to spend on individual problems.

I also underestimated the time needed for writing. My experimental notes were useful, but turning them into clear explanations required careful thought. I had to decide what belonged in each chapter and check that my descriptions matched the evidence.

If I started again, I would create a weekly schedule with smaller deadlines for reading, experiments, analysis and writing. I would leave extra time for unexpected difficulties and review my progress regularly. Most importantly, I would begin writing while the practical work was taking place rather than leaving most of it until later.

8.2 How My Study Skills Improved

The project improved the way I organise information. At first, I concentrated on completing each experimental step without always considering how I would explain it later. When I began writing, I realised that brief notes were not enough to describe every decision or identify the evidence collected at a particular stage.

As the project progressed, I became more organised in the way I studied and completed practical tasks. I learned to review my notes regularly and check my work before moving on to the next stage. This helped me become more confident in managing a large research project independently.

My academic writing also developed. Initially, I found it difficult to separate the methodology, implementation and evaluation chapters because they all discussed the same project. Through revision, I learned to explain the planned approach in the methodology, describe the completed work in the implementation and discuss the findings in the evaluation.

One area I could have improved was academic reading. I could have returned to relevant literature more regularly during the experiments instead of focusing mainly on practical tasks. In future, I would keep a reading log showing the main ideas from each source and how they relate to my research. This would help me connect academic knowledge with my own work more effectively.

8.3 What I Learned About the Way I Study

I learned that I work best when I divide a large task into smaller parts. Thinking about the entire dissertation sometimes made the workload feel overwhelming. Focusing on one experiment or section at a time helped me make steady progress.

I also realised that completing a task does not always mean I fully understand it. There were times when I had finished an experimental step but struggled to explain why I had taken that approach. Writing a short explanation immediately afterwards would have helped me identify gaps in my understanding earlier.

Another important lesson was how I respond to unexpected results. At first, I found unsuccessful acquisition attempts frustrating because I felt they were delaying my

progress. Gradually, I became more comfortable recording what had happened and considering what the available evidence could support.

This changed the way I viewed difficulties. I no longer felt that every experiment needed to produce the result I expected. Understanding a limitation and explaining it accurately could also contribute to the research. I want to carry this approach into future academic and professional work.

8.4 Connecting Theory with Practice

Before starting the experiments, I had studied the importance of following a clear research method and supporting conclusions with evidence. During the project, I saw why these principles mattered.

For example, collecting evidence before and after deletion was only useful when I could identify which files belonged to each stage. Clear records allowed me to compare the acquisitions and explain the changes I observed. This showed me that documentation is part of the investigation itself, not simply something to complete afterwards.

I also became more careful about interpreting results. When a search did not identify a test message, I initially wanted to treat that as proof that the message had been removed. However, I had to consider whether it might exist in storage I could not access. I learned to distinguish between what I observed and what I could not determine.

Applying these principles helped me understand how the ideas I had studied relate to decisions made during practical research. It also made me more confident in explaining findings without making claims that went beyond the evidence.

8.5 How I Could Have Improved My Work

Looking back, I would spend more time preparing before beginning the main experiments. A small pilot test could have helped me identify practical difficulties, estimate how long each stage would take and decide what evidence I needed to collect.

I would also prepare a checklist for every experiment. It would include the actions to complete, files to preserve, screenshots to capture and observations to record. Although I organised my evidence as the project progressed, planning these details earlier would have made the process more consistent.

Another improvement would be to write a short summary at the end of each working session. I would record what I completed, any unresolved problems and what I intended to do next. This would make it easier to return to the work after a break and provide a useful record when writing the dissertation.

I could also have asked for feedback earlier. In future, I would prepare specific questions for my supervisor rather than waiting until a large section of work was complete. This could help me identify problems with my approach or writing before they affected later stages.

Finally, I would leave more time to review the dissertation before submission. This would help me identify repeated explanations, correct mistakes and make sure that the tables and figures match the discussion.

8.6 Conclusion

This project helped me become more confident in working independently, organising information and responding when things did not go as expected. It also showed me that I need to improve how I estimate the time required for practical tasks, connect my reading with my research and begin writing earlier.

If I completed another project, I would follow a clearer schedule, keep detailed records from the beginning and seek feedback throughout the process. I would also allow enough time to review the final report rather than treating writing as the last stage.

The main lesson I learned is that completing research involves more than obtaining results. I need to understand my decisions, recognise what I could have done better and use those experiences to improve the way I approach future work.

Chapter 9: References

References

[1] R. F. Abu Hweidi, et al., "Forensics Investigation on Social Media Apps and Web Apps Messaging in Android Smartphone," 2023-07-25. doi: 10.1109/smartnets58706.2023.10216267.

[2] StatCounter Global Stats. "Mobile Operating System Market Share Worldwide." StatCounter Global Stats. https://gs.statcounter.com/os-market-share/mobile/worldwide (accessed 6 August).

[3] A. Almuqren, et al., "A Systematic Literature Review on Digital Forensic Investigation on Android Devices," 2024. doi: 10.1016/j.procs.2024.04.126.

[4] C. Anglano, "Forensic analysis of WhatsApp Messenger on Android smartphones," *Digital Investigation,* vol. 11, no. 3, -05-27 2014. doi: 10.1016/j.diin.2014.04.003.

[5] D. Walnycky, et al., "Network and device forensic analysis of Android social-messaging applications," *Digital Investigation,* vol. 14, -08 2015. doi: 10.1016/j.diin.2015.05.009.

[6] R. Bramesta Kusumadewa and Z. Sari, "Comparative Analysis of Forensic Digital Evidence on Android Smartphone based Instant Messaging Using NIST Framework,"

[7] R. Ayers, S. Brothers and W. Jansen, "Guidelines on mobile device forensics," 2014.

[8] J. Son, et al., "Forensic analysis of instant messengers: Decrypt Signal, Wickr, and Threema," *Forensic Science International: Digital Investigation,* vol. 40, -01-31 2022. doi: 10.1016/j.fsidi.2022.301347.

[9] D. Millatina, E. H. Gunawan and B. Sugiantoro, "Forensic Analysis of WhatsApp, Instagram, and Telegram on Virtual Android Device," 2024-04-29. doi: 10.1109/isdfs60797.2024.10527308.

[10] A. M. Azahari, A. Oliveri and D. Balzarotti, "Resilience of forensic evidence acquisition under database schema drift," *Forensic Science International: Digital Investigation,* vol. 56, -03-24 2026. doi: 10.1016/j.fsidi.2026.302059.

[11] E. Daraghmi, et al., "Forensic Operations for Recognizing SQLite Content (FORC): An Automated Forensic Tool for Efficient SQLite Evidence Extraction on Android Devices," *Applied Sciences,* vol. 13, no. 19, -09-27 2023. doi: 10.3390/app131910736.

[12] A. R. Onik, et al., "A Systematic Literature Review of Secure Instant Messaging Applications from a Digital Forensics Perspective," *ACM Comput. Surv.,* vol. 57, no. 9, -05-07 2025. doi: 10.1145/3727641.

[13] A. Afzal, et al., "Encrypted Network Traffic Analysis of Secure Instant Messaging Application: A Case Study of Signal Messenger App," *Applied Sciences,* vol. 11, no. 17, -08-24 2021. doi: 10.3390/app11177789.