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

A Forensic Framework for Investigating Dropbox Artifacts on Android Devices Using Cellebrite UFED

by

Md Saidul Islam Arif

Supervisor: Renuga Jayakumar

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

February 2025

DECLARATION

I, Md Saidul Islam Arif 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: Saidul

Date: 02/02/2026

FORM OF CONSENT

I, Md Saidul Islam Arif hereby consent that my Project, submitted in candidature for the MSc 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 Saidul

Date: 02/02/2026

ABSTRACT:
This dissertation designs and demonstrates a practical forensic framework for investigating Dropbox client-side artifacts on Android devices using industry-standard mobile forensic tooling. The study is motivated by a common investigative constraint in cloud storage cases: direct access to provider-side logs and server-held content is frequently delayed, legally restricted, or infeasible in educational and resource-constrained settings. Consequently, the Android endpoint becomes a primary and testable source of evidence through application databases, caches, configuration remnants, and file-operation metadata. An applied, empirical methodology is adopted. A controlled experiment is conducted on a Samsung Galaxy A14 running Android 13, executing a predefined sequence of Dropbox actions - login, upload, download, enable offline access, delete, and logout - using a fixed dataset of four images, two PDF files, and three video files. Data acquisition is performed via Cellebrite UFED 4PC using a File System extraction, with SHA-256 hashing applied to support evidential integrity verification. The extracted dataset is then parsed and examined in Cellebrite Physical Analyzer, guided by an action-to-artifact mapping model that structures analysis around account attribution artifacts, file metadata/timestamps, cache/offline remnants, and deletion-state indicators. The results demonstrate that the framework can reliably support user attribution and activity reconstruction from device-resident artifacts. In particular, the investigation confirms recovery of a Dropbox account identifier (user email) and file-operation timestamps, enabling the initial construction of a coherent activity timeline aligned with the controlled ground-truth actions. The framework's effectiveness is evaluated through action-to-artifact traceability, artifact coverage, integrity validation, and consistency checks, while transparently documenting limitations arising from single-device scope, app/OS version variability, and the inherent boundary of client-side visibility. Overall, this research contributes a repeatable, integrity-focused workflow that strengthens practical Dropbox investigations on modern Android devices and provides a defensible basis for reporting in mobile-cloud forensic contexts

Keywords: Cloud storage forensics, Dropbox forensics, Android forensics, Mobile device forensics, Digital evidence acquisition, Cellebrite UFED, File system extraction, Physical Analyzer, Evidence integrity (hashing).

TABLE OF CONTENTS

Table of Contents 4
List of Figures 6
List of Tables 6

CHAPTER 1: INTRODUCTION 7
1.1 Background and Context 7
1.2 Problem Statement 7
1.3 Aim of the Research 8
1.4 Research Objectives 8
1.5 Research Questions 9
1.6 Scope of the Study 9
1.7 Significance of the Research 9
1.8 Overview of the Proposed Forensic Framework 10

CHAPTER 2: LITERATURE REVIEW 10
2.1 Overview 10
2.2 Necessity, Impact and Motivation for This Research.....10
2.3 Digital Forensics Foundations and Process Models 12
2.4 Android Forensics Context 13
2.5 Integrity, Validation, and Reliability 14
2.6 Cloud Computing and Cloud Forensics Challenges 14
2.7 Cloud Storage Forensics and Client-side Artifacts 15
2.8 Mobile Cloud Forensic Frameworks 15
2.9 Dropbox Artifact Research 16
2.10 Identified Gaps and Research Motivation 16

CHAPTER 3: RESEARCH METHODOLOGY 18
3.1 Research Approach and Design 18
3.2 Research Environment and Materials 19
3.3 Proposed Forensic Framework Methodology 19
3.4 Data Collection Procedure 20
3.5 Data Acquisition Method 21
3.6 Artifact Identification Strategy 21
3.7 Data Analysis Method 22
3.8 Validity, Reliability, and Trustworthiness 22
3.9 Ethical and Legal Considerations 23
3.10 Limitations 23

CHAPTER 4: DESIGN PROCESS (EXPERIMENT DESIGN) 24
4.1 Framework-driven Design Rationale 24
4.2 Experimental Environment and Assumptions 24
4.3 Controlled Dataset Design 25
4.4 Action Scenario Design 25
4.5 Preservation and Contamination Controls 25
4.6 Acquisition Design (UFED 4PC File System) 26
4.7 Artifact Model Design 26

4.8 Action-to-Artifact Mapping Design 27
4.9 Evaluation Design 27
4.10 Framework Diagram.....28

CHAPTER 5: IMPLEMENTATION (EXPERIMENT) 29
5.1 Implementation Environment and Case Setup 29
5.2 Preservation and Pre-Acquisition Handling 30
5.3 UFED 4PC File System Extraction 31
5.4 Physical Analyzer Parsing and Setup 40
5.5 Dropbox Artifact Identification and Analysis 42
5.6 Locating Dropbox data sources in the extracted file system……...43
5.7 Reporting and Documentation 52

CHAPTER 6: EVALUATION OF RESULTS 54
6.1 Evaluation Criteria 54
6.2 Summary of Outcomes 55
6.3 Action-to-Artifact Evaluation Results 56
6.4 Integrity Verification Results 57
6.5 Framework Effectiveness Discussion 58
6.6 Limitations and Threats to Validity 58

CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS 60
7.1 Summary of Research Outcomes 60
7.2 Answers to Research Questions 60
7.3 Contribution of the Study 61
7.4 Recommendations for Forensic Practice 62
7.5 Recommendations for Future Work 62
7.6 Closing Statement 63

CHAPTER 8: REFLECTION 64
8.1 Learning and Skills Development 64
8.2 Methodological Reflection 64
8.3 Practical Challenges 65
8.4 Improvements If Repeated 65
8.5 Professional Development Impact 66
8.6 Reflection on Topic Pivot 66

REFERENCES 68
BIBLIOGRAPH……71
APPENDIX.....71

List of Figures:

Figure:3.1 Follow Charts

Figure 4.1: Forensic Framework for Investigating Dropbox Artifacts on Android Using Cellebrite UFED

Figure 5.1: Device information screenshot (model + Android version)

Figure 5.2: UFED case creation screen

Figures 5.3: Device detected in UFED

Figures 5.4: Extraction type selection (Full File System)

Figure 5.5: UFED extraction completion summary (screenshot)

Figure 5.6: Hash log (SHA-256) for UFED

Figure 5.7: Physical Analyzer evidence imports screen

Figure 5.8: Parsing progress/completion screen (screenshot)

Figure 5.9: Dropbox application entry / package evidence views in Physical Analyzer

Figure 5.10: Screenshot showing recovered email identifier

Figures 5.11: Example metadata record for a test file (filename + timestamp)

Figure 5.12: deletion-state indicator

Figure 5.13: Action-to-timestamp evidence mapping

Figures 5.14: Physical Analyzer timeline view showing Dropbox-relevant events

Figure 6.1: Screenshot of hash generation output

List of Tables:

Table 4.1: Action-to-Artifact Mapping Model

Table 6.1: Action → Artifact Traceability Evaluation (Dropbox on Android 13)

CHAPTER 1: INTRODUCTION

1.1 Background and Context

The widespread adoption of smartphones has transformed how individuals and organizations store, share, and access digital information. Modern mobile devices increasingly rely on cloud storage platforms to synchronize content across devices, support collaboration, and enable rapid data recovery. Among these platforms, Dropbox remains one of the most commonly used cloud storage applications due to its multi-platform availability and straightforward synchronization model [1]. However, the same convenience that cloud storage provides to users can also be exploited for illicit purposes, such as data exfiltration, concealment of contraband material, unauthorized distribution of sensitive information, and destruction or manipulation of evidence.

From a digital forensics' perspective, cloud services introduce fundamental investigative challenges. Traditional approaches often assume that an examiner can seize a physical device and acquire data directly from local storage. In cloud-based ecosystems, evidence may be distributed across endpoints (mobile devices, desktops) and remote infrastructure controlled by a cloud service provider. Investigators may face legal and procedural barriers when attempting to obtain server-side logs or user content stored in the cloud, and these barriers can delay or limit evidence acquisition [2]. Consequently, client-side artifacts stored on mobile devices - such as application databases, cached files, authentication remnants, metadata, and event timestamps - remain crucial sources for reconstructing user activity and establishing evidential narratives. Prior research emphasizes that cloud-related investigations frequently depend on artifacts collected from endpoints rather than directly from the cloud service itself [3].

Mobile device forensics, as an established sub-discipline, provides structured guidance on the handling, acquisition, examination, analysis, and reporting of digital evidence from mobile devices. The National Institute of Standards and Technology (NIST) outlines principles and procedural considerations that support forensic soundness and reproducibility, including evidence preservation, validation, and documentation [4].

1.2 Problem Statement

Although Dropbox activity may be conducted in the cloud, evidence of its use often persists on the user's mobile device. These client-side remnants can include account identifiers, timestamps of file operations, cached copies of content, thumbnails, and database entries that reflect user actions (e.g., upload, download, deletion, and offline access). Research has shown that cloud storage applications may leave forensic artifacts across private app storage and related system locations, which can assist an examiner in reconstructing events even when some content is deleted or no longer available in the cloud [5].

Despite the importance of mobile artifacts, investigators and students often face practical obstacles: rapid platform updates, device encryption, variations in Android versions, and differences in tool capabilities. In many educational or resource-constrained environments, it is not feasible to perform complex cloud-side acquisition or legal processes to access provider logs [6]. This creates a need for a practical, repeatable, and well-structured forensic approach that focuses on the Android client device as the primary evidence source for Dropbox usage.

Therefore, the central problem addressed in this thesis is: how to design and apply a forensic framework that supports reliable identification, acquisition, preservation, and analysis of Dropbox artifacts on an Android device using industry-standard mobile forensic tooling.

1.3 Aim of the Research

The aim of this research is to design and demonstrate a forensic framework for investigating Dropbox artifacts on an Android device, using Cellebrite UFED 4PC for data acquisition and Cellebrite Physical Analyzer for examination and analysis, with emphasis on preserving evidential integrity and supporting activity reconstruction.

1.4 Research Objectives

To achieve the stated aim, this thesis pursues the following objectives:

I. Design a structured forensic framework for investigating Dropbox usage on Android devices that aligns with established mobile forensic guidance (e.g., preservation, validation, and reporting).

II. Conduct a controlled experiment on a Samsung A14 (Android 13) using a realistic set of Dropbox actions (login, upload, download, offline access, deletion, logout).

III. Acquire data using a File System extraction method via UFED 4PC and preserve integrity through cryptographic hashing (SHA-256).

IV. Identify and categorize Dropbox artifacts likely to be of forensic relevance (e.g., account identifiers, timestamps, file metadata, caches, and relevant databases), guided by prior cloud-storage forensic studies.

V. Reconstruct user activity using extracted timestamps and metadata to support a coherent timeline of Dropbox usage on the device.

VI. Evaluate the framework's effectiveness using traceability of actions to artifacts, integrity validation, and consistency of findings with expected user behavior.

1.5 Research Questions

This study is guided by the following research questions:

RQ1: What Dropbox-related forensic artifacts can be recovered from an Android 13 device following typical user actions (login, upload, download, offline access, deletion, logout)?

RQ2: How can a structured forensic framework support systematic acquisition, preservation, and interpretation of Dropbox artifacts using Cellebrite UFED 4PC and Physical Analyzer?

RQ3: To what extent can recovered artifacts (e.g., user email and file timestamps) enable timeline reconstruction and activity attribution on the device, under a repeatable experimental design?

1.6 Scope of the Study

This thesis is deliberately scoped to maintain feasibility and methodological depth:

- **Device scope:** One Android device (Samsung A14) running Android 13.
- **Application scope:** Dropbox mobile application only.
- **Tool scope:** Cellebrite UFED 4PC for acquisition and Cellebrite Physical Analyzer for analysis.
- **Acquisition scope**: File System extraction (as successfully completed in practice).
- **Evidence scope:** Client-side artifacts on the mobile device (databases, cache, metadata, timestamps, account traces).

The study does not attempt to:

- perform direct server-side acquisition from Dropbox infrastructure,
- bypass security controls unlawfully,
- generalize results to all Android versions, devices, or cloud applications.

This scoping approach is consistent with the reality that many clouds forensic investigations must rely on endpoint artifacts due to access constraints, and it aligns with established research that examines client-side remnants of cloud storage usage.

1.7 Significance of Research

This research contributes value in three practical ways:

1. Framework contribution: It provides a structured, repeatable framework tailored to a realistic and common investigative scenario: a cloud storage app on a mobile device.

Framework-oriented approaches are frequently recommended to address the complexity and fragmentation of mobile-cloud evidence.

2. Practical contribution: It demonstrates an implementable workflow using widely deployed forensic tools and a contemporary Android device, improving applicability for practitioners and students. Studies have highlighted the importance of tool-driven workflows and artifact-focused methodologies for mobile-cloud investigations.
3. Evidential contribution: It emphasizes integrity (hashing) and validation to support forensic soundness, aligning with guidance on preservation and reporting in mobile device forensics.

1.8 Overview of the Proposed Forensic Framework

At a high level, the framework implemented in this thesis follows a phased forensic process:

1. Seizure and Preservation: device isolation and documentation (to minimize contamination).
2. Acquisition: UFED 4PC File System extraction and hashing (SHA-256) for integrity validation.
3. Artifact Identification: locating Dropbox databases, caches, configuration remnants, and metadata.
4. Analysis and Reconstruction: correlating artifacts to user actions and building a timeline.
5. Reporting and Validation: presenting results with hash verification and traceability mapping.

This structure aligns with established forensic process models and mobile forensic guidance emphasizing preservation, acquisition validation, and systematic reporting.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview

This chapter reviews the academic and professional literature underpinning a forensic framework for investigating Dropbox artifacts on Android devices using Cellebrite UFED 4PC and Physical Analyzer. The review is structured around four themes:

(i) Core digital and mobile forensic principles and process models.
(ii) Android platform characteristics that shape evidence acquisition.
(iii) Cloud and cloud-storage forensics, including client-side artifacts.
(iv) Reliability, integrity, and validation concerns when using commercial forensic tools.

The chapter concludes by identifying specific gaps that motivate the proposed framework and the focus on Dropbox activities (login, upload, download, offline use, deletion, and logout) on Android 13.

Search strategy: The literature search was performed across five widely used scholarly sources: IEEE Xplore, the ACM Digital Library, SpringerLink, ScienceDirect, and Google Scholar. The query set was built around the following core terms and close variants: cloud forensics, volatile data, cloud-native technology, forensic tools, and data division.

Eligibility criteria: Studies were included if they were peer-reviewed journal articles or conference papers, published between 2020 and 2025, and explicitly addressed challenges relevant to cloud forensics, the handling or preservation of volatile data, cloud-native environments. Items were excluded if they were non-peer-reviewed, published before 2020, or primarily conceptual/opinion-based without a clear empirical, experimental, or technical contribution.

Study selection outcome: The database search returned 200 records. After removing 40 duplicates, 160 unique studies remained for title-and-abstract screening. From these, 47 papers were advanced to full-text assessment.

2.2. Necessity, Impact and Motivation for This Research

The necessity of this research is driven by the growing overlap between mobile device forensics and cloud storage usage in everyday activity and investigations. Cloud storage applications such as Dropbox increasingly function as a second file system for users, supporting upload, download, caching, and offline access - behaviours that can produce device-resident traces even when primary content is stored remotely. Prior empirical research demonstrates that cloud storage clients (including Dropbox) can leave forensically relevant artifacts on Android and iOS endpoints, such as account identifiers and file-related metadata, which can support investigative reconstruction when direct cloud-side access is unavailable or delayed [1].

Despite evidence that such artifacts exist, a key operational gap remains. Investigators need repeatable, action-driven guidance to link user behaviour (login, upload, download, offline access, deletion, logout) to observable artifacts on modern Android devices and app versions. Artifact persistence and visibility are affected by platform security controls (sandboxing, encryption), app version changes, and parsing differences across tools [2]. Therefore, a framework-based approach is needed to convert scattered artifact knowledge into a structured investigation method that improves traceability from action → artifact category → evidential interpretation. In addition, research on file system interpretation highlights that reliability requires careful validation and documentation because interpretation errors can arise from complex data structures and investigator over-reliance on tool summaries [3]. This reinforces why the thesis is necessary as a method contribution, not only an artifact listing.

This research is necessary in practical terms because mobile forensic work increasingly depends on commercial tools and semi-automated interpretation, which can create risk if artifact meaning is assumed rather than validated. Recent empirical work shows that digital forensic tools' presentation choices (e.g., naming conventions, aggregation, timestamp context) can contribute to misinterpretation and investigator error, even among practitioners [4]. For Dropbox investigations specifically, this means that simply finding data is not enough; investigators need a defensible process that:

1. prioritises strong attribution anchors (e.g., account email),
2. reconstructs activity using corroborated timestamps and metadata, and
3. communicates limitations transparently (e.g., incomplete visibility into cloud-side history).

2.3 Digital forensics foundations and investigation process models

Most digital forensic methodologies converge on a phased workflow that supports evidential integrity and courtroom defensibility. NIST SP 800-86 describes a widely adopted structure - collection, examination, analysis, and reporting - emphasizing controlled acquisition, documentation, and repeatability [5]. In parallel, NIST SP 800-101 Rev. 1 defines mobile device forensics as the recovery of digital evidence under forensically sound conditions and highlights the importance of preservation, acquisition, examination, analysis, and reporting for mobile devices [6]. These works are frequently treated as baseline guidance because they formalize principles (integrity, documentation, controlled handling) rather than prescribing one tool or one technical method.

A recurring theme in mobile forensics research is the lack of consistent, standardized process models tailored to the diversity of mobile platforms and their rapid evolution. Al-Dhaqm et al. [7] reviewed a large body of mobile forensic investigation process models and argued that variability in device infrastructures and artifacts leads to fragmented practice, recommending harmonization

of steps and terminology. This is particularly relevant to app-centric investigations (e.g., Dropbox), where evidence is distributed across application databases, caches, authentication tokens, and synchronized cloud metadata, and where each Android version or vendor customization can subtly shift artifact locations and availability.

2.4 Android forensics context: platform characteristics and evidence locations

Android investigations are shaped by application sandboxing, permission models, and encryption. In typical deployments, application private data is stored within app-specific directories (commonly under /data/data/<package>), while media and user-accessible content may also appear on shared/external storage. From a forensic perspective, this architecture creates two immediate consequences:

1. high evidential value often resides in app-private databases and cache directories and
2. acquiring that data can be restricted by encryption, device lock state, and acquisition method (logical vs. file system vs. physical). NIST's mobile forensic guidance highlights how device configurations and security features influence what can be collected and how [6].

Modern Android analysis increasingly focuses on application artifacts rather than only system-level traces. Anglano et al. [8] presented AnForA as a tool to automate forensic analysis of Android applications and, in doing so, reinforced the broader research view that app artifacts (SQLite databases, shared preferences, cached files, logs) are central to reconstructing user behavior. Similarly, Azfar et al. [9] proposed taxonomies for extracting and interpreting mobile application forensic artifacts, showing that meaningful reconstructions commonly come from correlating multiple artifact types (timestamps, identifiers, content remnants) rather than relying on a single file or database [10]. These studies collectively support an approach where Dropbox investigation is evidence-le, identify which artifacts are expected for each user action, then validate those expectations empirically on the target device and OS version.

Encryption and secure hardware features substantially affect acquisition depth. Recent work highlights that encrypted smartphones can limit traditional full physical approaches, shifting practice toward filesystem and logical extractions combined with targeted artifact interpretation [11], [12]. For Android 13 devices, this reinforces the practical value of a framework that is explicit about,

(a) what a file system extraction can reliably recover from Dropbox,

(b) what may remain cloud-resident only, and

(c) which conclusions require corroboration across multiple traces.

2.5 Evidence integrity and validation in tool-driven mobile forensics

The literature on forensic reliability, validation, and integrity is directly relevant. Measurement-science perspectives argue that digital forensics must demonstrate reliability through validation practices, transparency of assumptions, and reproducibility of results [13]. Later work provides primers and structured approaches for validating tools and methods, stressing that defensible conclusions should not depend solely on a black box extraction outcome [14]. In the specific context of file system interpretation and reverse engineering, Nordvik et al. [15] propose reliability validation procedures to ensure that parsing and interpretation steps remain scientifically and legally robust, while Stoykova and Franke [16] extend this into a reliability validation enabling framework that emphasizes documentation and chain-of-evidence reasoning across technology, method, and application levels.

Integrity controls - particularly hashing - remain fundamental to defensible acquisition and analysis. Roussev et al. [17] discusses hashing as a core forensic mechanism for integrity verification and efficient identification. However, hash algorithm choice and interpretation must be handled carefully. Kessler's [18] analysis of MD5 collision concerns concludes that while collision attacks exist, practical forensic risk should be evaluated realistically and mitigated via stronger hashing and sound procedures when needed. In practice, this supports the common pattern used in tool-based workflows: computer hashes for extracted images/archives and key exports, preserves originals read-only, and ensure every analysis output can be traced back to a verified input.

Tool behavior also matters for repeatability. For example, Guido et al. [19] explored differential acquisition approaches for mobile devices to reduce acquisition overhead while maintaining evidential relevance, reinforcing the broader point that acquisition strategy is a methodological choice and should be justified in the research design. In your case, selecting File System extraction (rather than physical) is a defensible methodological decision when full physical access is constrained, provided limitations are explicitly documented and the artifact coverage is evaluated.

2.6 Cloud computing and cloud forensics: definitions and recurring challenges

Cloud computing's characteristics - on-demand service, broad network access, resource pooling, rapid elasticity, measured service - are commonly framed using the NIST definition [20]. These properties are not just conceptual they create forensic constraints because evidence can be distributed across endpoints, provider infrastructure, and SaaS application layers, sometimes with limited visibility to investigators.

Cloud forensics literature consistently highlights technical and procedural challenges: multi-tenancy, jurisdiction and legal access, limited acquisition interfaces, provider-controlled logging,

and difficulties verifying completeness. Simou et al. [21] provide a detailed survey of challenges and solutions, emphasizing that methodologies are still maturing and that practical constraints frequently shape what evidence is collectible. Manral et al. [22] offer a systematic survey that categorizes the field and underscores unresolved issues across the forensic lifecycle, particularly around acquisition and preservation at scale. These findings support a client-centric investigative stance in mobile-cloud cases. Even when cloud-side evidence is ideal, endpoint artifacts may be the most accessible and testable evidence source in academic and many operational contexts.

2.7 Cloud storage forensics and client-side artifacts

Within cloud forensics, cloud storage services represent a practical and widely studied subset because they produce rich cross-layer traces, local sync clients, mobile apps, web sessions, and cloud metadata. Chung et al. [23] proposed procedures for investigating cloud storage services by collecting artifacts from accessible devices and correlating them to reconstruct cloud usage. Quick and Choo examined whether the act of collecting cloud storage data changes the data or metadata, an issue that is directly relevant to evidence preservation decisions and supports documenting acquisition-side effects and timestamps carefully [24].

Research also emphasizes that cloud evidence can be cloud-native rather than merely copied to endpoints. Roussev and McCulley [25] argue that cloud-native artifacts preserve the persistent state of SaaS/web applications and require approaches that may differ from traditional endpoint-only methods. This highlights a realistic boundary for a Dropbox-on-Android thesis. A file system extraction can produce strong evidence of local app state and user actions, but it may not fully represent server-side history unless the app catches it locally or you have lawful cloud access and a provider-facing acquisition method.

2.8 Mobile cloud forensics frameworks: bridging device and cloud

Mobile cloud forensics literature aims to bridge endpoint artifacts and cloud services with explicit process models. Faheem et al. [26] survey mobile cloud forensic approaches and argue that investigations require coordinated handling of mobile device data sources and cloud-side traces, often under time and access constraints. In a later contribution, Faheem et al. [27] propose a mobile cloud forensic framework (MCFF), emphasizing structured phases that account for the unique coupling between mobile clients and cloud platforms. These works justify designing your thesis as a framework-based study rather than a single case report. The goal becomes a repeatable method that maps user actions to expected artifacts, specifies acquisition and validation steps, and records limitations when artifacts are absent due to encryption, caching policies, or tool constraints.

2.9 Dropbox artifact research on mobile devices

Dropbox has been repeatedly used as a case study in cloud storage forensics because it creates measurable traces on client devices and within app storage. Daryabar et al. [28] compared multiple cloud storage apps on Android and iOS, showing that user identifiers, timestamps, and file-related metadata can often be recovered from mobile applications, supporting reconstruction of user activity under controlled experimental conditions. Martini and Choo's [29] analysis of popular Android apps (including cloud storage) further reinforces that meaningful forensic traces may remain in local app storage even when content appears cloud-based from the user's perspective.

More focused Dropbox artifact studies document concrete storage locations and artifact categories. Satrya et al. [30] (Dropbox artifact analysis) report that application data structures - such as local databases and cached records can expose account data and interaction traces that support investigation timelines. This aligns with your early practical findings (email identifiers and timestamps) and supports expanding your artifact set to include authentication/session remnants, file metadata tables, offline availability indicators, local cache inventories, sync state markers, and deletion traces (where recoverable).

However, existing Dropbox studies often vary in device versions, OS versions, and acquisition methods, which can limit generalization to Android 13 and a Samsung A14. Therefore, the literature points to the need for controlled, version-specific experimentation: reproduce a known action sequence (login → upload → download → mark offline → delete → logout), then measure which artifacts consistently appear in UFED/Physical Analyzer outputs and how strongly they support each inference.

2.10 Identified gaps and rationale for this thesis

From the reviewed literature, four gaps justify my proposed thesis framework:

1. Version-specific artifact uncertainty: Many studies demonstrate that artifacts exist but do not always generalize across Android versions and vendor ecosystems. Android security evolution and app updates can change database schemas, cache behavior, and encryption effects, creating a need for Android 13 - specific validation [6], [31], [32].

2. Action-to-artifact mapping remains under-specified: Frameworks and surveys emphasize structured approaches, yet many applied papers stop at listing artifacts without systematically mapping them to discrete user actions (login/upload/ download/offline/ delete/logout) in a repeatable experiment design [26], [27].

3. Tool transparency and reliability expectations**:** Literature on validation stresses that investigators should document tool versions, extraction types, parsing assumptions, and cross-check methods to improve defensibility and reproducibility - especially with commercial tools [13]-[16].

4. Client-centric cloud evidence limitations: Cloud forensics work shows that endpoint artifacts can be strong but incomplete representations of cloud activity therefore, a rigorous framework must explicitly state scope boundaries (device-resident artifacts) and identify what cannot be concluded without cloud-side acquisition [21], [22], [25].

Overall, the literature supports your chosen direction: a framework-based investigation that is

(i) Experimentally controlled.

(ii) Centered on Dropbox artifacts recoverable via file system extraction on Android 13.

(iii) Validated through integrity controls and careful reporting of limitations.

CHAPTER 3: RESEARCH METHODOLOGY

3.1 Research Approach and Design

3.1.1 Research approach

This study follows an applied, empirical approach, where a proposed forensic framework is tested through hands-on experimentation. The goal is not to develop a new forensic tool, but to produce a repeatable framework that can guide investigators in locating and interpreting Dropbox artifacts on Android devices using industry-standard tooling.

A key rationale for this approach is that mobile-cloud investigations often face constraints in direct cloud-side acquisition, making endpoint evidence (mobile artifacts) a practical and academically defensible primary evidence source. Existing studies in cloud storage and Dropbox forensics show that meaningful traces, such as user identifiers, timestamps, and file-operation metadata can persist on the client device. Therefore, controlled experimentation on the client device is an appropriate way to evaluate artifact availability and investigative value.

3.1.2 Research design

The research design combines:

1. Framework design (defining phases, steps, and validation points)
2. Controlled experiment (performing known Dropbox actions and observing resulting artifacts).

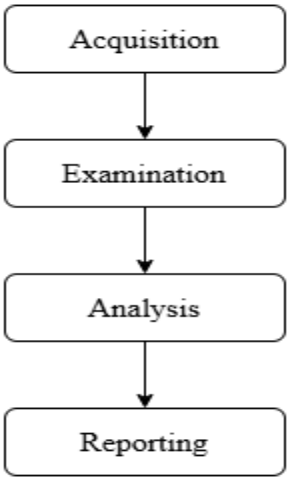


Figure:3.1 Follow Chart

This design is consistent with standard forensic guidance that emphasizes a structured workflow and the need to document procedures to support reproducibility.

3.2 Research Environment and Materials

3.2.1 Hardware and platform

- Device: Samsung Galaxy A14
- Operating system: Android 13
- Application under investigation: Dropbox (Android app)
- Workstation: A Windows-based forensic workstation (used to run UFED 4PC and Physical Analyzer)

3.2.2 Forensic tools

- Cellebrite UFED 4PC: Used for device connection, acquisition planning, and performing File System extraction.
- Cellebrite Physical Analyzer: Used for parsing, artifact categorization, searching, timeline reconstruction support, and report generation.

These tools are widely used in operational forensic settings and are frequently adopted in peer-reviewed experimental research because they support standardized acquisition and analysis workflows for mobile devices. Research demonstrate forensic extraction and analysis using Cellebrite UFED (alongside comparison tooling) and discuss evidence availability and accuracy against planned test instances, illustrating the tool's suitability for controlled forensic experimentation and repeatable evidence recovery studies [24]. This directly aligns with the thesis design, which relies on a controlled action sequence to evaluate what Dropbox artifacts remain on the device. A commercial tool choice strengthens the research's external validity: the framework is designed for realistic investigator practice, not only theoretical artifact exploration. However, the literature also supports using such tools with an explicit validation mindset. Work on reliability validation in file system interpretation emphasizes that defensible conclusions require cross-checking and documenting interpretation assumptions, which this thesis implements through (i) action-to-artifact mapping, (ii) corroboration across artifact categories, and (iii) integrity controls (hashing) [32]. Additionally, findings on tool-induced bias reinforce the importance of structuring analysis and corroboration so that investigator conclusions do not depend exclusively on a single parsed view or interface presentation [25].

3.3 Proposed Forensic Framework Methodology

The framework applied in this study consists of five phases:

1. Seizure and Preservation
2. Acquisition (UFED 4PC)

3. Artifact Identification (Dropbox-focused)
4. Analysis and Reconstruction
5. Reporting and Validation

These phases are aligned with NIST guidance for forensic processes and mobile device forensics, particularly the emphasis on preservation, validated acquisition, systematic examination, and reporting.

3.4 Data Collection Procedure

3.4.1 Pre-acquisition preparation (controlled conditions)

To strengthen repeatability and reduce contamination, the experiment was performed under controlled conditions:

- A test Dropbox account was used.
- A well-known set of test files was prepared:
    - 4 images, 2 PDF documents, and 3 video files.
- A defined sequence of user actions was performed inside Dropbox:
    1) login, 2) upload files, 3) download files, 4) mark files for offline access, 5) delete files, 6) logout.

This structured action sequence enables later mapping between known actions and observed artifacts, which is necessary for evaluating whether the framework supports activity reconstruction.

3.4.2 Evidence preservation

Preservation practices were applied to reduce the risk of evidence alteration during handling:

- The device was isolated from networks where possible (to reduce background sync and remote changes).
- Device details and condition were documented (device model, OS version, date/time of acquisition steps).
- All generated extraction outputs were stored securely and treated as primary evidence files for later validation.

These steps reflect standard mobile forensic preservation expectations.

3.5 Data Acquisition Method

3.5.1 Selection of acquisition type

This research uses File System extraction via UFED 4PC. File system extraction provides deeper access than purely logical acquisition and is commonly used when full physical access is restricted by encryption or device security controls. The choice is therefore methodologically justified for modern Android devices, provided the limitations are disclosed and results are validated through corroboration across artifacts.

3.5.2 Acquisition workflow (method-level)

The acquisition workflow followed these method-level steps:

1. Connect device to UFED 4PC using a suitable data cable.
2. Select File System extraction for the target device profile.
3. Perform extraction and save the output in an examiner-controlled evidence directory.
4. Generate and record SHA-256 hash values for the extraction output (and key exported reports).
5. Preserve original extraction output as read-only (working on copies for analysis).

Hashing supports integrity verification and is widely recognized as a basic forensic control. While some literature discusses limitations of older algorithms (e.g., MD5 collisions), using stronger hashes such as SHA-256 reduces practical integrity risk for this research context.

3.6 Artifact Identification Strategy

The artifact identification strategy is Dropbox-centric and evidence-led. Rather than treating Dropbox data as one category, artifacts are grouped and searched based on their forensic meaning:

- Account artifacts: user identifiers (e.g., email), authentication/session traces (where recoverable), device/account linkage.
- File operation artifacts: evidence of upload, download, deletion, and offline availability.
- Metadata artifacts: filenames, paths, timestamps, file sizes, and unique IDs.
- Cache and content remnants: cached files, thumbnails, previews, and offline copies.
- Configuration artifacts: preferences and settings that may support user attribution and context.

Prior research on cloud storage and Dropbox artifacts shows that client-side databases and caches often contain timestamps and identifiers valuable for reconstruction, making these categories appropriate for systematic searching in Physical Analyzer.

3.7 Data Analysis Method

Analysis was performed in Cellebrite Physical Analyzer using a structured approach:

1. Ingestion and parsing: load the UFED extraction into Physical Analyzer for automated parsing of application data.
2. Dropbox-focused filtering: locate Dropbox-related datasets (app artifacts, databases, file listings, cache-related entries).
3. Keyword and identifier Searching: search for known test identifiers (email, test filenames) to establish ground truth links.
4. Timestamp analysis: extract and normalize timestamps associated with Dropbox actions and compare against the controlled action sequence.
5. Cross-artifact corroboration: confirm conclusions by correlating evidence across multiple artifacts (e.g., database record + file metadata + cache presence).
6. Export and documentation: export key findings and maintain traceability from findings back to source artifacts (for reporting).

This corroboration-first approach reflects reliability concerns highlighted in forensic validation literature, which warns against relying solely on a single parsed view or a single artifact without supporting evidence.

3.8 Validity, Reliability, and Trustworthiness

3.8.1 Internal validity

Internal validity is strengthened through:

- A controlled action sequence (known ground truth),
- Fixed file set and file types,
- Use of the same device, OS version, and single cloud app (Dropbox),
- Consistent acquisition method (File System extraction).

3.8.2 Reliability and reproducibility

Reliability is supported by:

- Explicit documentation of steps,
- Preservation of original extraction outputs,
- Hash-based integrity verification,
- Repeatable artifact categorization and searching procedures.

Tool and method validation literature emphasizes documenting tools, versions, assumptions, and repeatable procedures to improve the defensibility of conclusions, especially with commercial tools that may not be fully transparent internally.

3.8.3 Construct validity (linking artifacts to actions)

Construct validity is addressed by mapping specific user actions (upload/download/ offline/ delete/ logout) to expected artifact categories, then validating those mappings by searching for confirming traces. This supports stronger inference than simply listing recovered artifacts.

3.9 Ethical and Legal Considerations

This dissertation uses a test account and test files under the researcher's control, avoiding unauthorized access to third-party data. No rooting, illegal bypass, or unauthorized access methods are required to support the research aims. The study focuses on client-side artifacts produced through normal app usage and standard forensic acquisition procedures, consistent with accepted academic research practice in digital forensics.

3.10 Limitations

The main limitations of this methodology are:

- Single-device scope: results are validated on one device model (Samsung A14) and Android 13 artifact availability may differ across devices and OS versions.
- App version variability: Dropbox updates may change artifact schemas and cache behaviors over time.
- Cloud-side visibility: this study primarily evaluates device-resident artifacts, not Dropbox server-side logs or provider data, which may require legal processes and specialized cloud acquisition.
- Extraction constraints: file system extraction may not expose all possible deleted content or encrypted elements conclusions must reflect what is observable in the extracted dataset.

These limitations will be revisited in the evaluation chapter to ensure conclusions remain appropriately bounded.

CHAPTER 4: DESIGN PROCESS (EXPERIMENT DESIGN)

This chapter presents the design process and experimental design used to implement the proposed forensic framework for investigating Dropbox artifacts on an Android device. The purpose of the design is to ensure the investigation is controlled, repeatable, and traceable, so that observed artifacts can be confidently linked to specific Dropbox user actions (login, upload, download, offline access, deletion, and logout). The chapter defines the experimental environment, test dataset, action sequence, acquisition plan, and the action-to-artifact mapping model that guides analysis and later evaluation. The design is aligned with established digital/mobile forensic process expectations, particularly regarding documentation, integrity, and structured reporting [33], [34].

4.1 Framework driven design rationale

A framework-based thesis requires more than tool usage it must provide a structured method that can be reused and evaluated. In this study, the framework is operationalized through an experiment design that converts abstract forensic phases into measurable steps:

- Preservation controls: minimizing contamination and uncontrolled synchronization.
- Acquisition controls: consistent extraction methods and hashing.
- Artifact model: identifying what evidence is expected from each Dropbox action.
- Traceability: linking findings to sources and test actions.
- Validation: integrity checking and corroboration across artifacts.

This approach reflects standard forensic guidance emphasizing lifecycle structure (collection/ examination/ analysis/ reporting) and mobile-specific handling constraints [33], [34].

4.2 Experimental environment and assumptions

4.2.1 Target device and application

- Device: Samsung Galaxy A14
- Operating system: Android 13
- Cloud application: Dropbox (Android application)
- Forensic tools: Cellebrite UFED 4PC (acquisition) and Cellebrite Physical Analyzer (analysis)

4.2.2 Key assumptions (explicit boundaries)

This research is intentionally designed around device-resident (client-side) evidence, not Dropbox server-side logs. Therefore, conclusions are bounded by what is observable in the UFED file system extraction and what Dropbox stores locally on Android. This boundary is consistent with prior cloud storage forensic studies showing that meaningful evidence can persist on client endpoints even when data is cloud-managed [35], [36].

4.3 Controlled dataset design (test files)

A controlled dataset was used to establish ground truth and improve traceability:

- 4 image files (e.g., JPG/PNG)
- 2 PDF documents
- 3 video files (e.g., MP4)

4.4 Action scenario design (ground truth sequence)

A structured action scenario was executed inside Dropbox to generate predictable artifacts. The sequence below forms the ground truth for later mapping and evaluation:

1. Login to Dropbox (test account)
2. Upload 4 images, 2 PDFs, and 3 videos
3. Download selected uploaded items to device storage (within Dropbox)
4. Enable Offline Access for selected items
5. Delete selected items (in-app deletion)
6. Logout from the Dropbox account

4.5 Evidence preservation and contamination control design

Because Dropbox can synchronize in the background, the design includes basic contamination controls:

- Minimize unnecessary interaction after completing the action scenario.
- Avoid additional Dropbox actions between the scenario and extraction.
- Record date/time and device state (battery, network condition, app state).

- Preserve UFED extraction outputs as read-only and analyze from working copies.

Such controls reflect general principles in mobile forensic guidance. Evidence should be preserved, and procedural steps should be documented to support reproducibility and integrity.

4.6 Acquisition design (UFED 4PC File System extraction)

4.6.1 Choice of extraction type

This thesis uses File System extraction via UFED 4PC because it typically provides more access to application data than logical extraction and is often the practical upper bound on modern encrypted Android devices without advanced exploitation pathways. The selection is appropriate for an applied framework study, provided limitations are documented and conclusions are supported by corroboration across artifacts [34].

4.6.2 Acquisition workflow plan (design-level)

The acquisition design follows this consistent workflow:

1. Device connection to the forensic workstation (UFED 4PC).
2. Selection of File System extraction for the Samsung A14 profile.
3. Extraction completion and secure storage of the resulting UFED package/output.
4. Integrity verification using SHA-256 hashing of extraction outputs (and key exports).
5. Preservation of original outputs analysis conducted on duplicates.

Hashing is a fundamental integrity mechanism in digital forensics, supporting confidence that the evidence was not altered during handling or analysis [47].

4.7 Artifact model design (what evidence is expected)

Dropbox investigations on Android commonly rely on local artifacts such as:

- Account identifiers (e.g., user email address)
- SQLite databases (records of files, sync state, recent activity, identifiers)
- Cache and offline files (local copies, previews, thumbnails)
- Metadata and timestamps (created/modified/accessed times, event times)
- Configuration remnants (preferences, state flags, device/app identifiers)

Prior studies examining Dropbox and other cloud storage clients indicate that mobile apps often leave recoverable traces in local databases and caches sufficient to reconstruct meaningful activity narratives [35], [36].

4.8 Action-to-artifact mapping design

To make the framework measurable, each user action is mapped to expected artifact categories. This mapping becomes the blueprint for analysis and evaluation.

Table 4.1: Action-to-Artifact Mapping Model

User Action	Primary Expected Artifacts	Supporting/Corroborating Artifacts	Example Evidence Types
Login	Account identifier in app data session/account state	timestamps in DB configuration/preferences	user email, account ID, login time markers
Upload	File metadata records in app DB	sync state logs, timestamps	filenames, file sizes, unique IDs, uploaded state
Download	Local cached copy or download record	file system timestamps cache index	local file present, preview cache, download timestamp
Offline Access	Offline cache/offline flags in DB	Local offline copies	offline status indicator stored file copy in app storage
Delete	Deletion record or state change in DB	absence of file possible residual cache	deleted flag, removal timestamp, orphaned cache remnants
Logout	account state change token/session removal (if accessible)	config changes last activity timestamp	logout event time, cleared session indicators

4.9 Evaluation design (how success will be measured)

To evaluate whether the framework is effective, the design defines measurable criteria that will be applied in evaluation chapter:

1. Traceability score: ability to map each ground-truth action to at least one strong artifact (primary) and one corroborating artifact.
2. Artifact coverage: number and diversity of recovered artifacts across categories (account, DB records, caches, timestamps).

3. Integrity assurance: evidence output remains hash-verified (SHA-256 recorded and consistent).
4. Consistency: artifacts and timelines align logically with the known action sequence.

This evaluation approach is consistent with forensic expectations for integrity, repeatability, and evidence-based reasoning.

4.10 Framework Diagram

Figure 4.1: Forensic Framework for Investigating Dropbox Artifacts on Android Using Cellebrite UFED

CHAPTER 5: IMPLEMENTATION (EXPERIMENT)

This chapter documents the practical implementation of the proposed forensic framework on a Samsung Galaxy A14 (Android 13) with Dropbox as the target cloud storage application. The implementation follows a structured forensic workflow preservation, acquisition, examination, analysis, and reporting using Cellebrite UFED 4PC for File System extraction and Cellebrite Physical Analyzer for parsing and artifact analysis. Here also explains how evidence integrity was protected through SHA-256 hashing and how findings are prepared for defensible reporting, consistent with established mobile forensic guidance and integrity practices.

5.1 Implementation Environment and Case Setup

5.1.1 Tools and platform

- Target device: Samsung Galaxy A14
- Operating system: Android 13
- Target application: Dropbox (Android app)
- Acquisition tool: Cellebrite UFED 4PC
- Analysis tool: Cellebrite Physical Analyzer

5.1.2 Case management structure

To support traceability and auditability, a consistent case folder structure was used:

- Case_ID: DropboxForencics
    - 1_ChainOfCustody
    - 2_Acquisition_UFED
    - 3_Hashes
    - 4_PhysicalAnalyzer_Project
    - 5_Exports_Reports
    - 6_Screenshots_Evidence

This structure ensures that all evidence outputs, hashes, and screenshots can be traced to the same case identifier and examiner workflow, a common expectation in forensic practice.

5.2 Phase 1: Device Preservation and Pre-Acquisition Handling

Before acquisition, basic preservation steps were applied to reduce contamination from background synchronization or accidental user activity:

1. Document device context: model (Samsung A14), OS (Android 13), date/time of handling, and condition (locked/unlocked state).
2. Network-control precaution: minimize external network activity to reduce cloud sync changes during handling (where possible).
3. Minimal interaction policy: after completing the planned Dropbox action scenario (login, upload, download, offline, delete, logout), the device was not used for unrelated activity prior to extraction.

These steps support the principle that digital evidence should be collected with minimal alteration and with clear documentation for repeatability [38], [39].

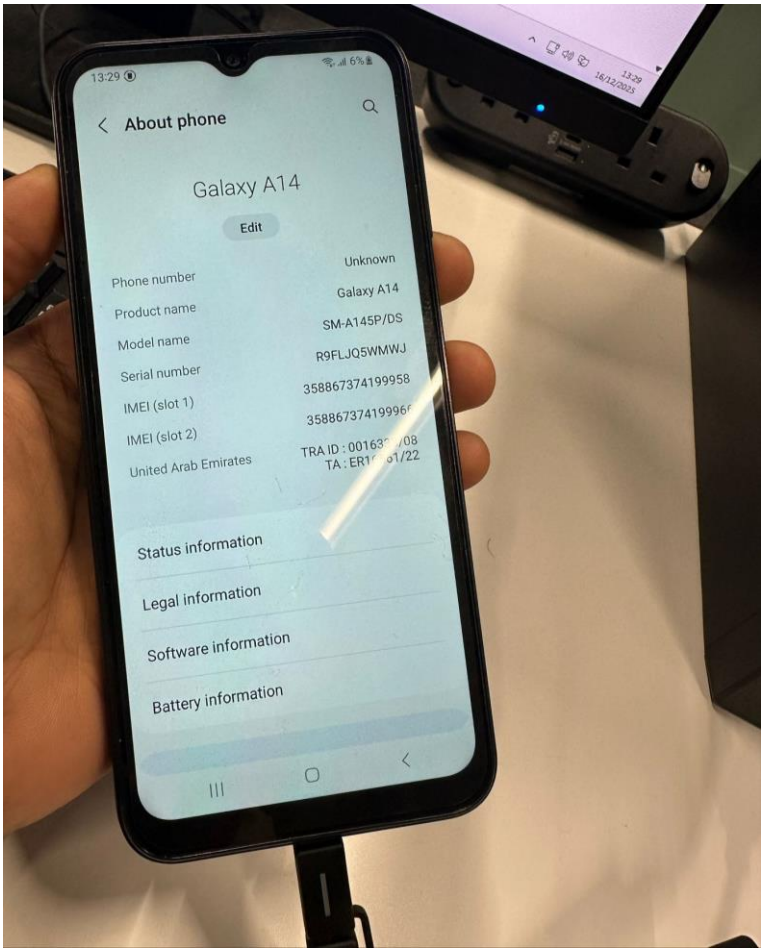


Figure 5.1: Device information screenshot (model + Android version)

5.3 Phase 2: Data Acquisition Using UFED 4PC (File System Extraction)

This study used File System extraction, which is often a practical and defensible approach for modern Android devices where full physical imaging may be constrained by encryption and security controls.

5.3.1 UFED 4PC acquisition procedure (practical workflow)

Step 1: Create a new UFED case

1. Open UFED 4PC.
2. Select New Case (or Create Case).
3. Enter case metadata (recommended fields):
    - Case ID
    - Examiner name
    - Organization: University of Wales Trinity Saint David
    - Device: Samsung Galaxy A14
    - Notes: Dropbox artifact investigation – Android 13 – controlled action sequence

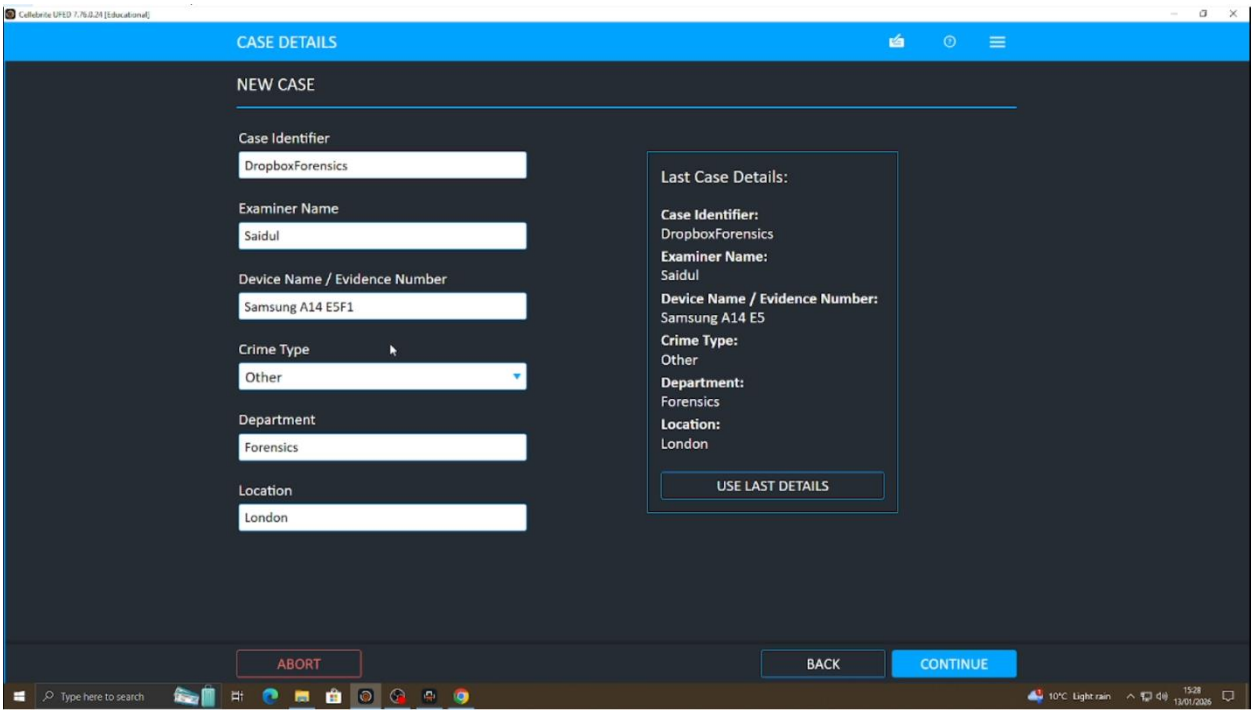
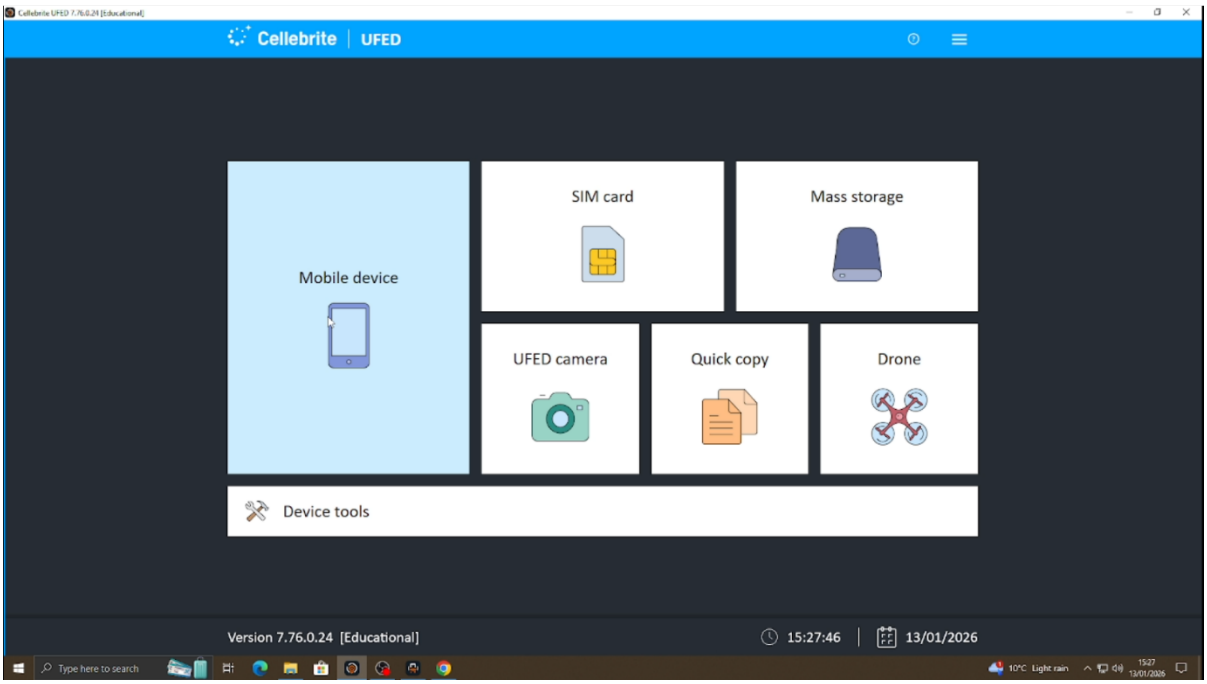
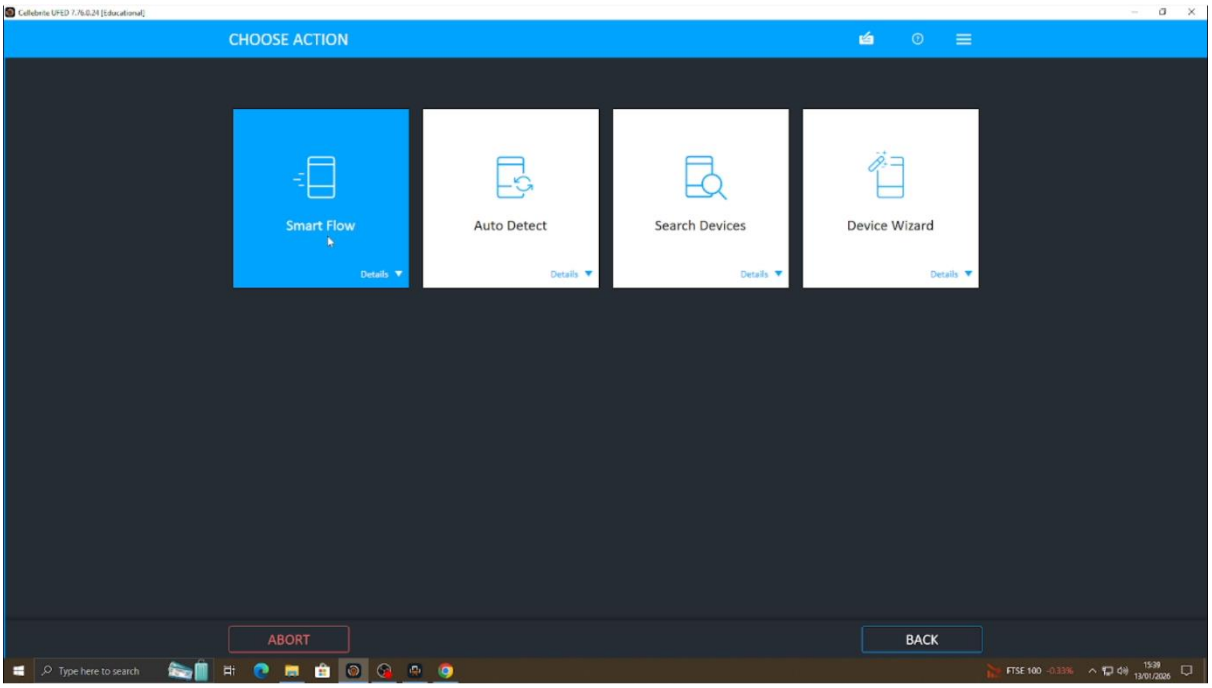


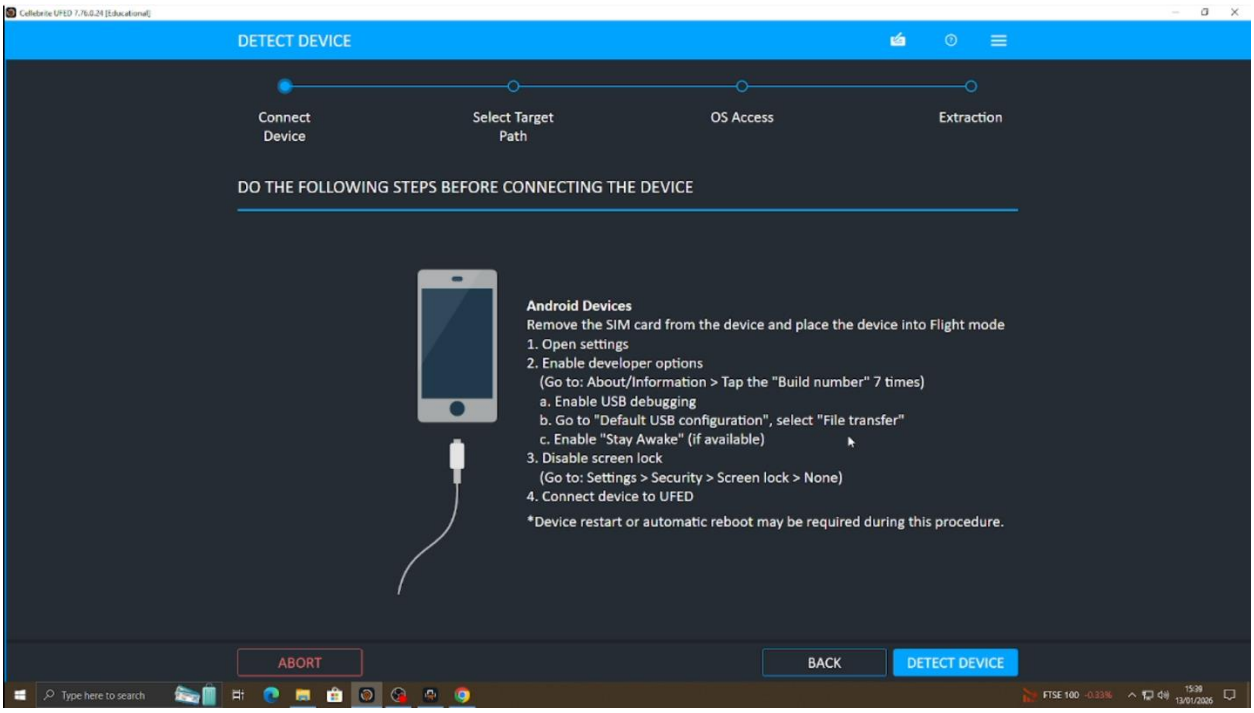
Figure 5.2: UFED case creation screen

Step 2: Device connection and identification

1. Connect the Samsung A14 to the forensic workstation using a data-capable USB cable.
2. Allow device prompts if they appear (e.g., Allow USB access).
3. In UFED 4PC, confirm that the device is detected and correctly identified (model/Android).



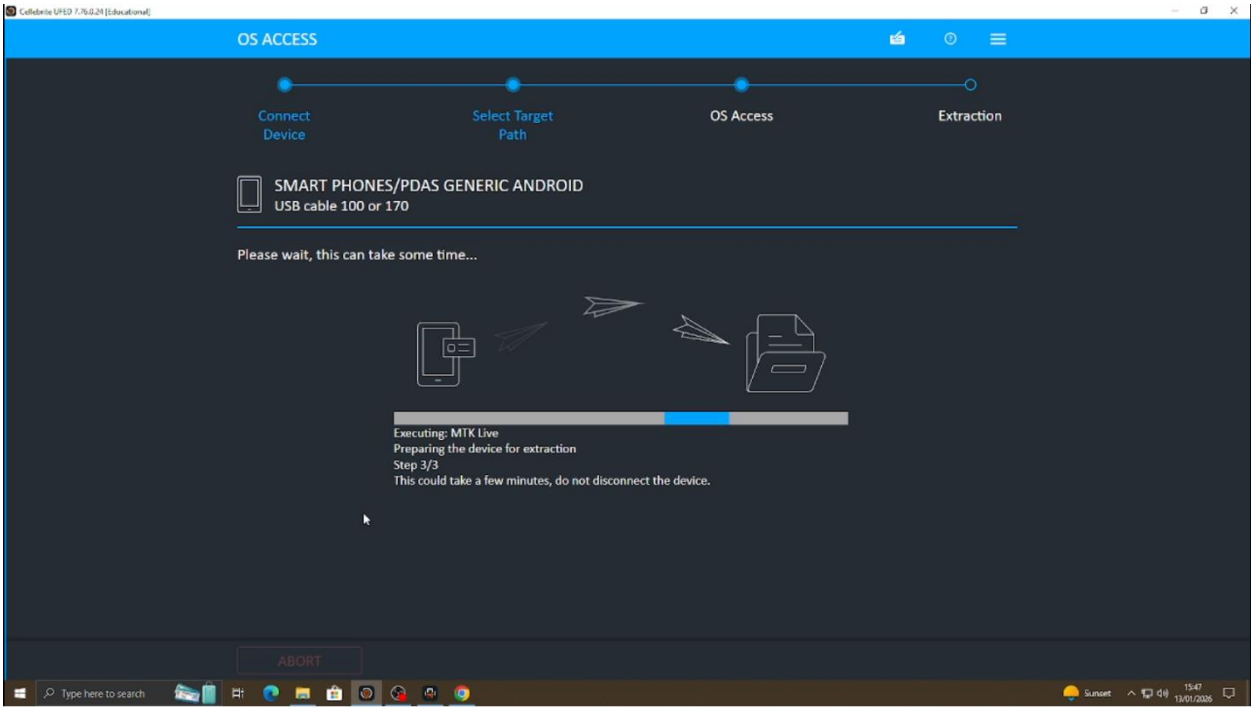


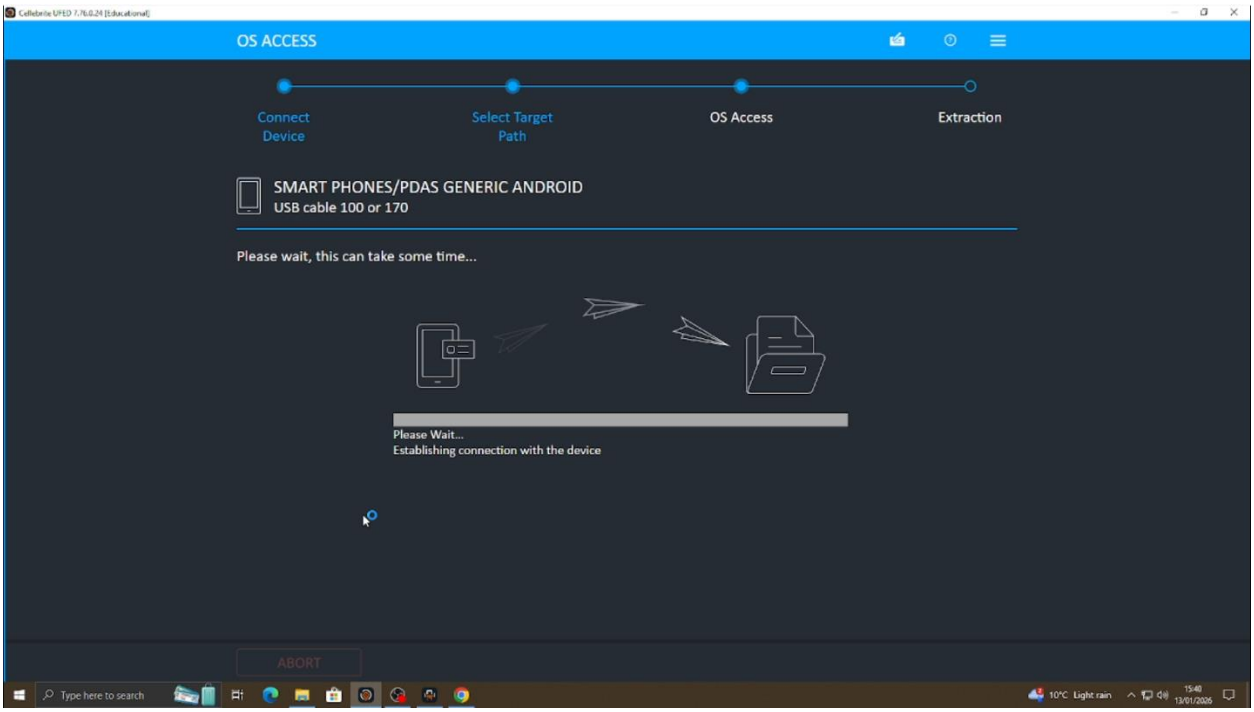


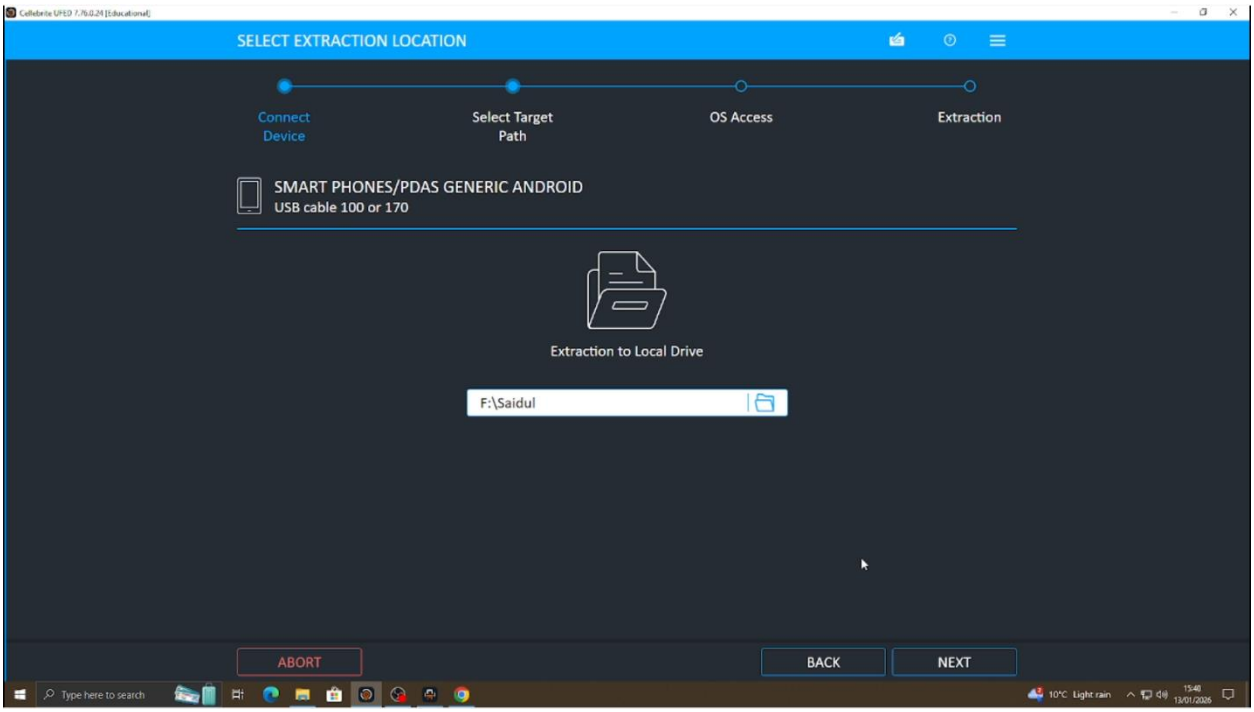
Figures 5.3: Device detected in UFED

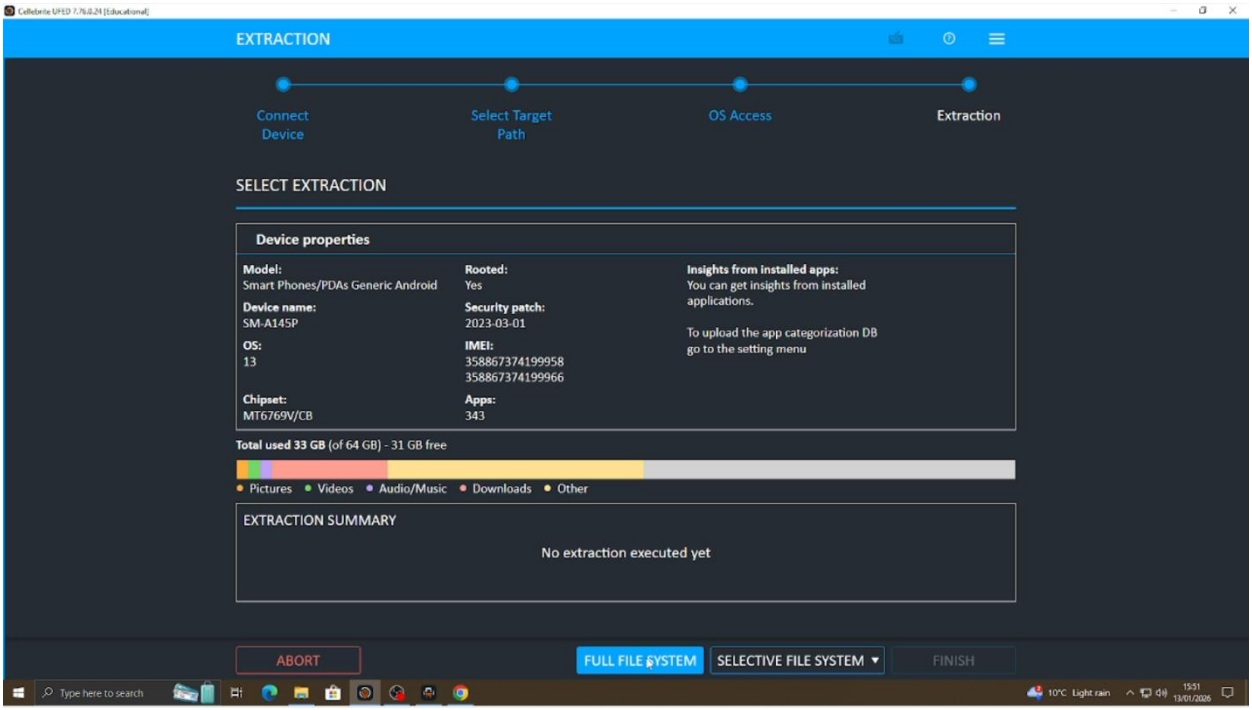
Step 3: Select extraction method

1. Choose Extraction and select Android
2. Select File System Extraction.
3. Follow UFED on-screen prompts. Depending on device configuration, UFED may request enabling settings such as:
    - Developer options / USB debugging (if required by the chosen workflow)
    - Permission confirmations on the device
4. Confirm the output destination folder under 02_Acquisition_UFED/.


OS ACCESS
Connect Device
Select Target Path
OS Access
Extraction
SMART PHONES/PDAS GENERIC ANDROID
USB cable 100 or 170
Please wait, this can take some time...
Executing: MTK Live
Preparing the device for extraction
Step 3/3
This could take a few minutes, do not disconnect the device.
ABORT


OS ACCESS
Connect Device
Select Target Path
OS Access
Extraction
SMART PHONES/PDAS GENERIC ANDROID
USB cable 100 or 170
Please wait, this can take some time...
Please Wait...
Establishing connection with the device
ABORT

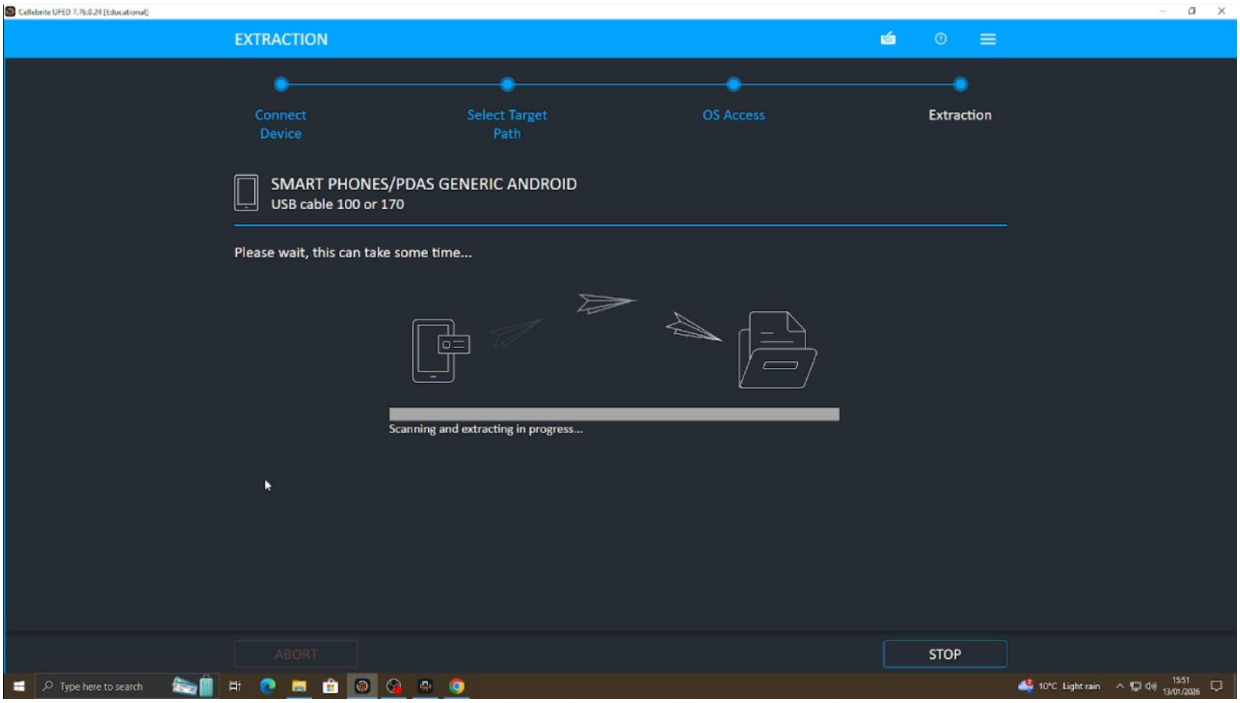


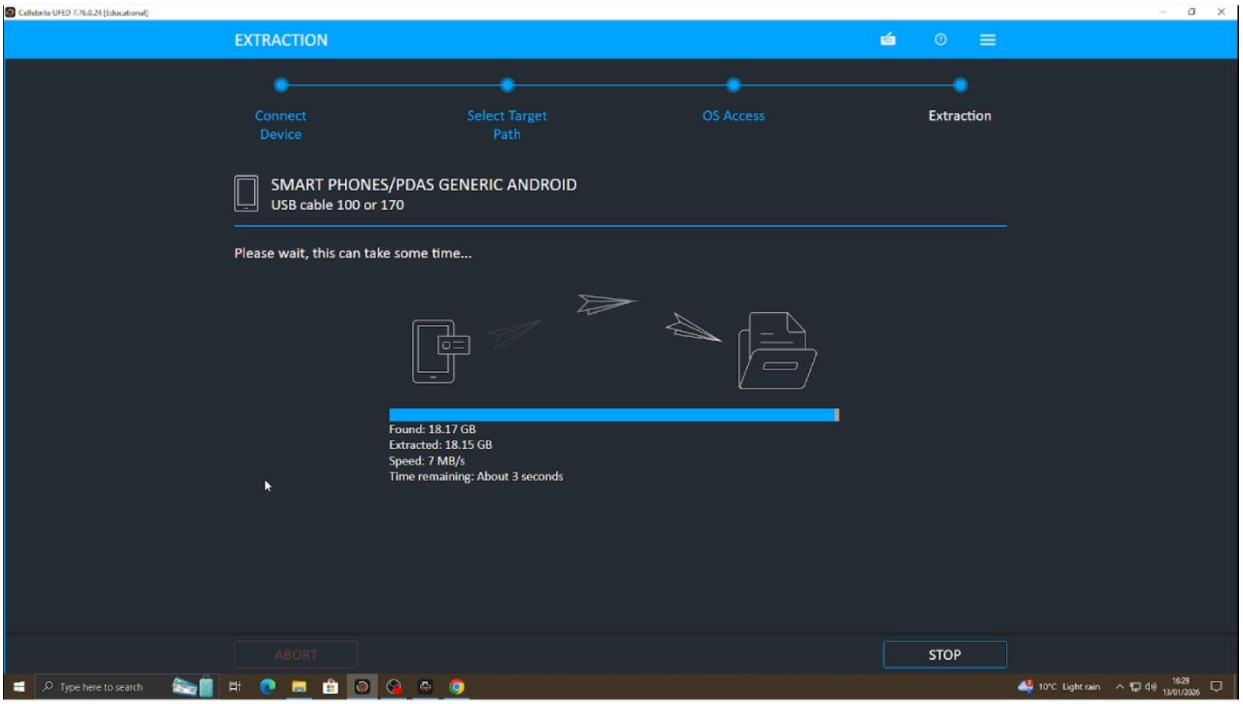


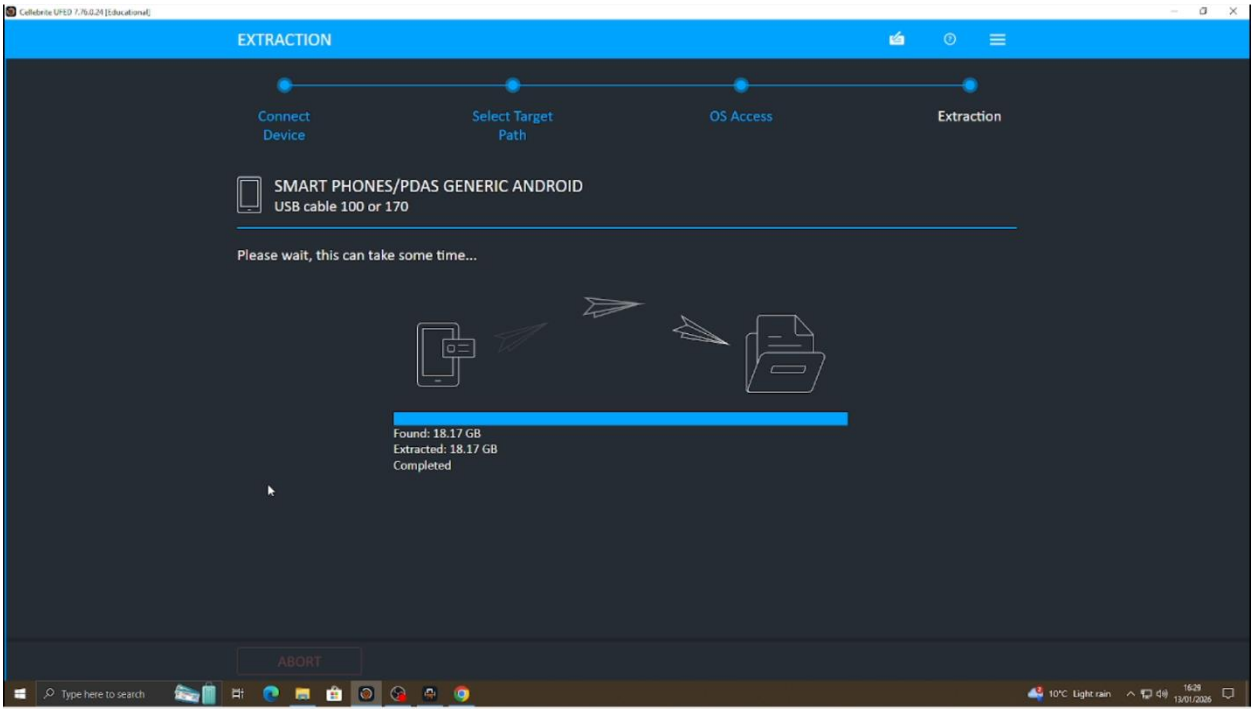
Figures 5.4: Extraction type selection (Full File System)

Step 4: Execute and save extraction output

1. Start the extraction and allow UFED to complete the process.
2. Save the extraction output in the case directory.
3. Export or generate any UFED acquisition summary and store it under 05_Exports_Reports/.







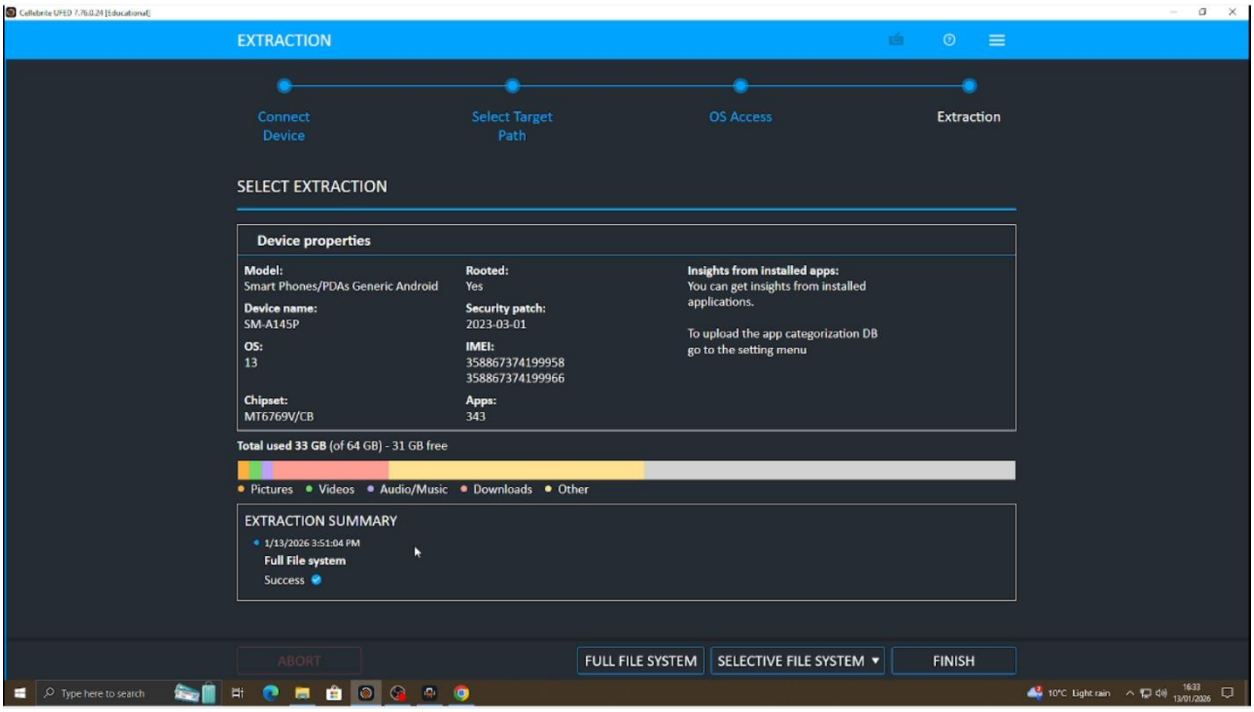


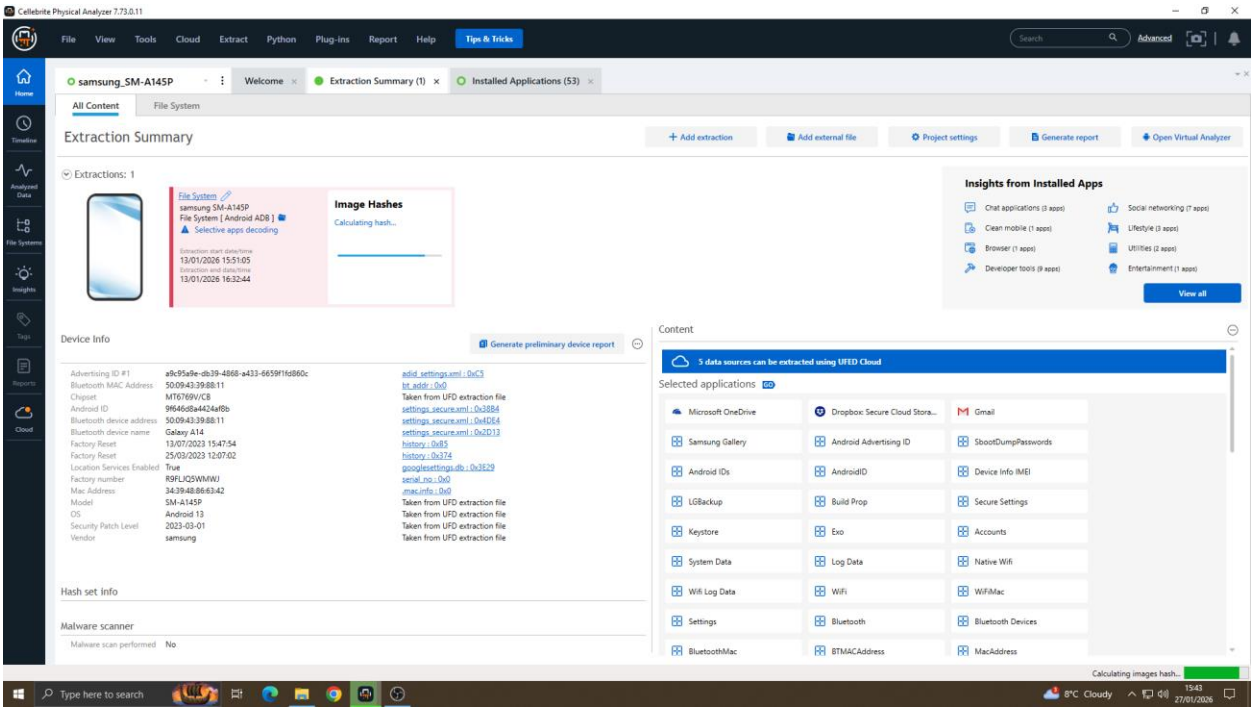
Figure 5.5: UFED extraction completion summary (screenshot)

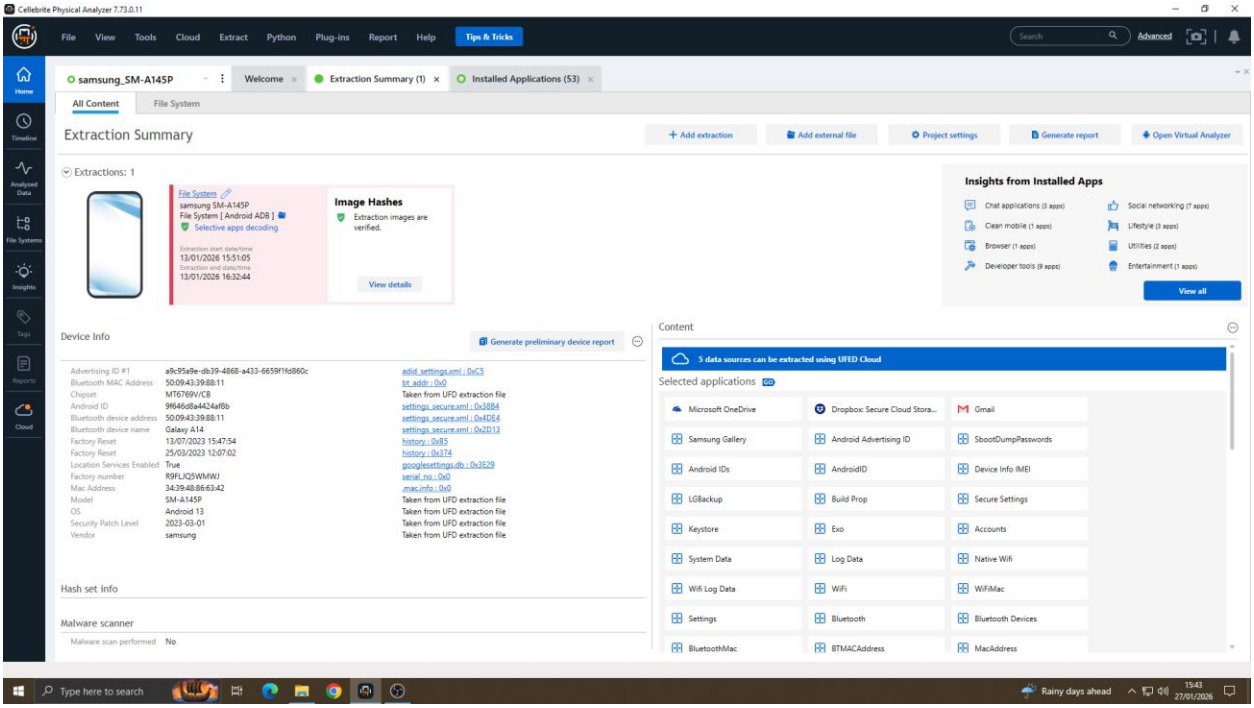
5.3.2 Evidence integrity: SHA-256 hashing

After acquisition, SHA-256 hash values were generated and recorded for the extracted evidence packages and any key exports. Hashing supports integrity verification by allowing later confirmation that the evidence has not changed during storage or analysis.

A hash log template was used:

- Evidence filename
- Hash algorithm (SHA-256)
- Hash value
- Date/time generated
- Tool used to compute hash
- Examiner signature/initials





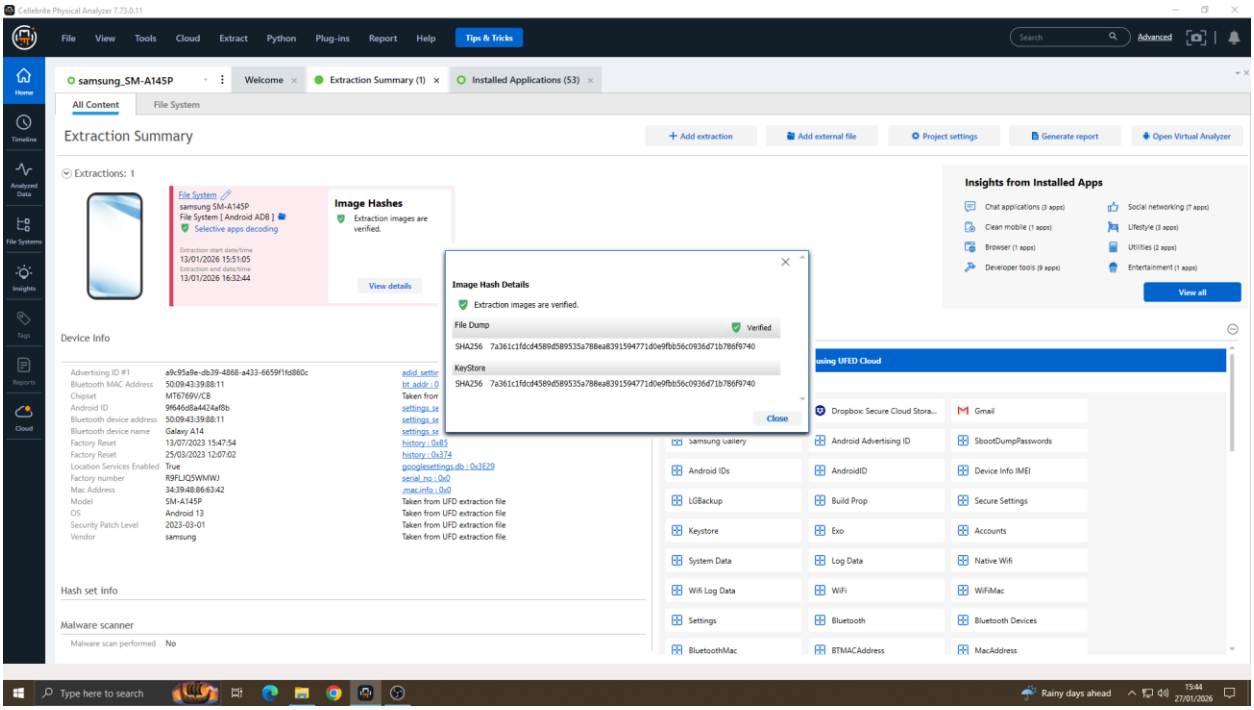


Figure 5.6: Hash log (SHA-256) for UFED

5.4 Phase 3: Loading and Parsing in Physical Analyzer

5.4.1 Import procedure

1. Open Cellebrite Physical Analyzer.
2. Create a new project under 04_PhysicalAnalyzer_Project/.
3. Select Add Evidence / Load Extraction.
4. Import the UFED File System extraction output.
5. Allow Physical Analyzer to parse artifacts automatically (applications, accounts, media, files, etc.).

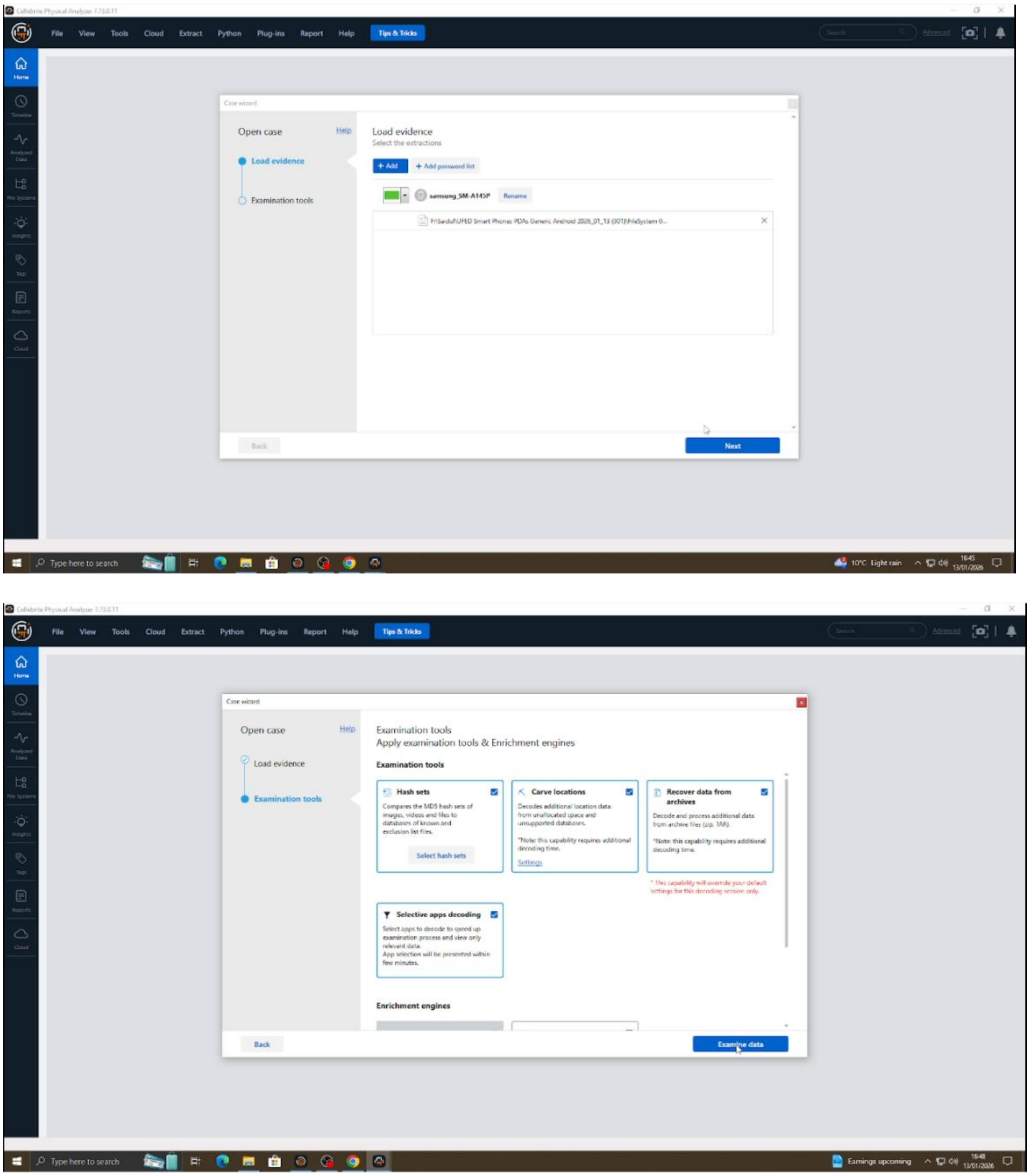
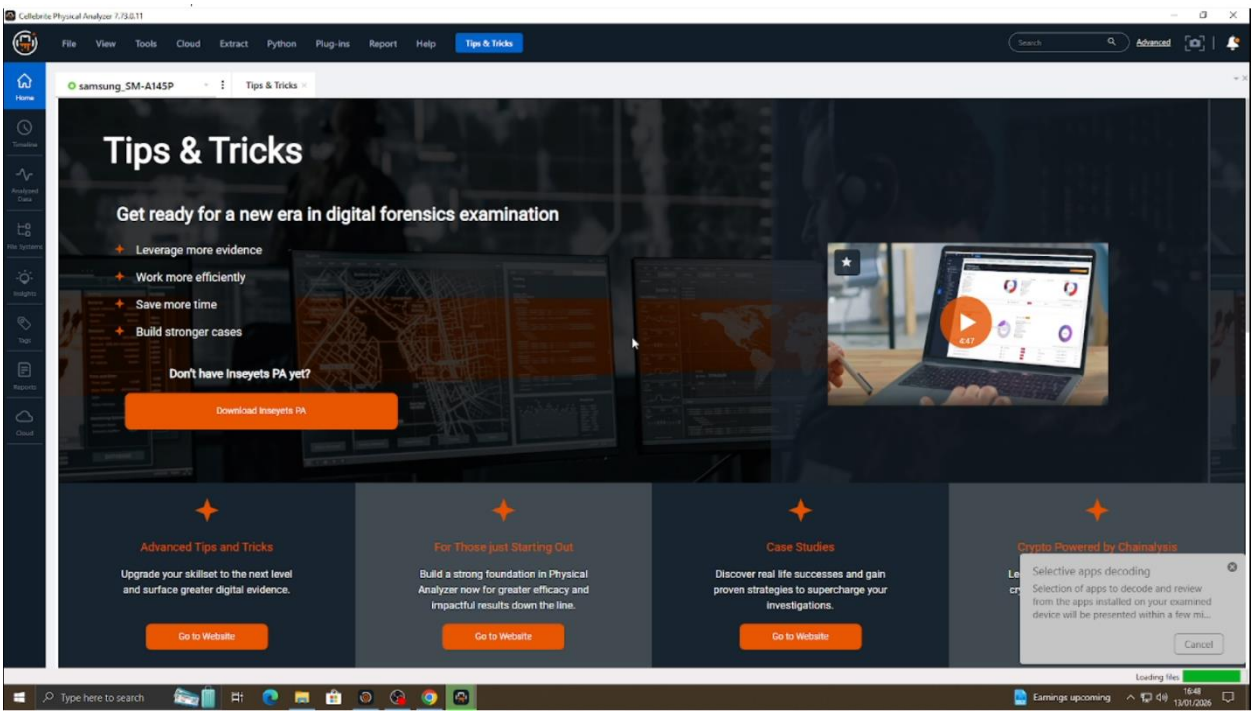
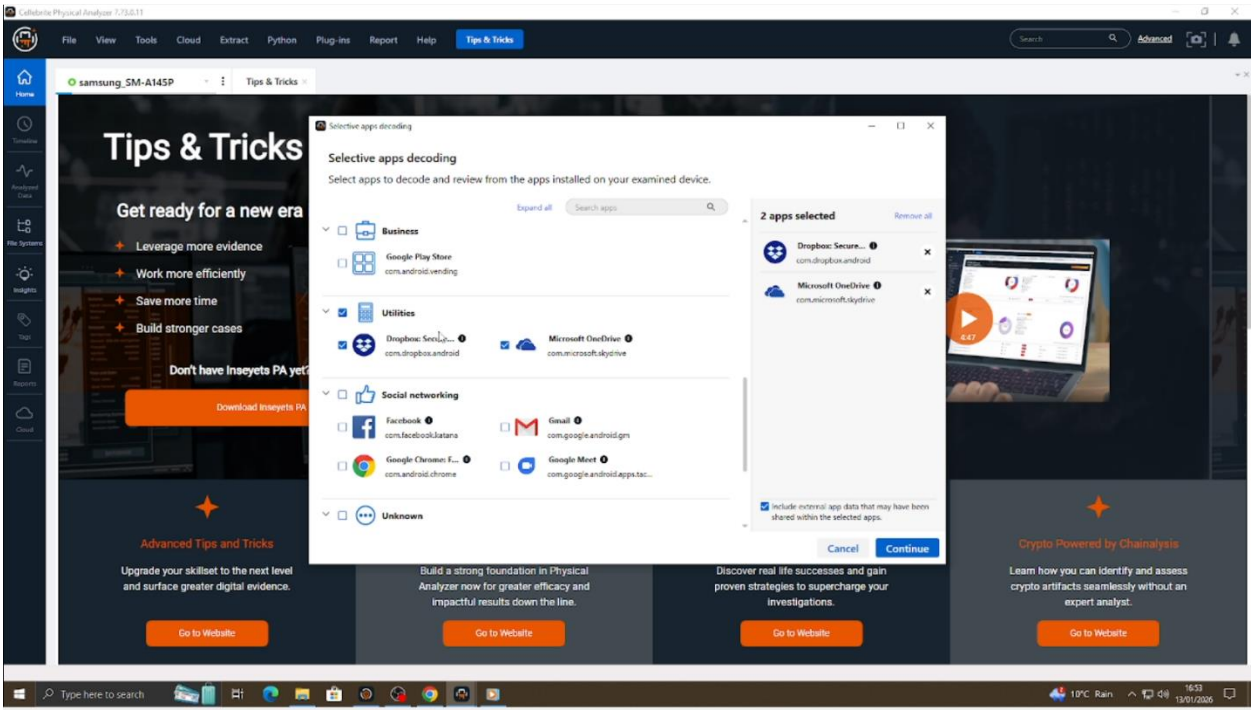


Figure 5.7: Physical Analyzer evidence imports screen


Cellebrite Physical Analyzer 7.73.0.11
File View Tools Cloud Extract Python Plug-ins Report Help Tips & Tricks
Advanced
samsung_SM-A145P
Tips & Tricks
Tips & Tricks
Get ready for a new era in digital forensics examination
Leverage more evidence
Work more efficiently
Save more time
Build stronger cases
Don't have Inseyets PA yet?
Download Inseyets PA
Advanced Tips and Tricks
Upgrade your skillset to the next level and surface greater digital evidence.
Go to Website
For Those just Starting Out
Build a strong foundation in Physical Analyzer now for greater efficacy and impactful results down the line.
Go to Website
Case Studies
Discover real life successes and gain proven strategies to supercharge your investigations.
Go to Website
Crypto Powered by Chainalysis
Selective apps decoding
Selection of apps to decode and review from the apps installed on your examined device will be presented within a few mi...
Cancel
Loading files
Type here to search
Earnings upcoming
16:48
13/01/2026


Cellebrite Physical Analyzer 7.73.0.11
File View Tools Cloud Extract Python Plug-ins Report Help Tips & Tricks
Advanced
samsung_SM-A145P
Tips & Tricks
Tips & Tricks
Get ready for a new era
Leverage more evidence
Work more efficiently
Save more time
Build stronger cases
Don't have Inseyets PA yet?
Download Inseyets PA
Selective apps decoding
Selective apps decoding
Select apps to decode and review from the apps installed on your examined device.
Expand all
Search apps
Business
Google Play Store
com.android.vending
Utilities
Dropbox: Secure...
com.dropbox.android
Microsoft OneDrive
com.microsoft.skydrive
Social networking
Facebook
com.facebook.katana
Gmail
com.google.android.gm
Google Chrome: F...
com.android.chrome
Google Meet
com.google.android.apps.tac...
Unknown
2 apps selected
Remove all
Dropbox: Secure...
com.dropbox.android
Microsoft OneDrive
com.microsoft.skydrive
Include external app data that may have been shared within the selected apps.
Cancel
Continue
Advanced Tips and Tricks
Upgrade your skillset to the next level and surface greater digital evidence.
Go to Website
Build a strong foundation in Physical Analyzer now for greater efficacy and impactful results down the line.
Go to Website
Discover real life successes and gain proven strategies to supercharge your investigations.
Go to Website
Crypto Powered by Chainalysis
Learn how you can identify and assess crypto artifacts seamlessly without an expert analyst.
Go to Website
Type here to search
10°C Rain
16:53
13/01/2026

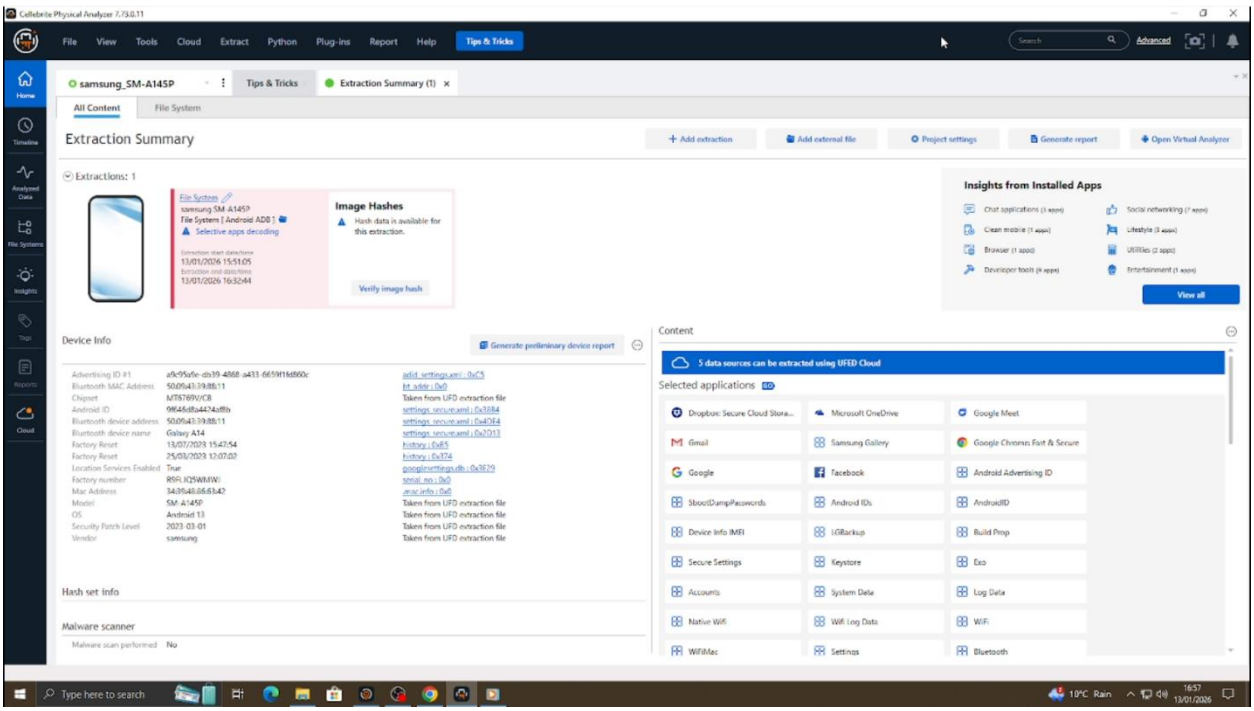


Figure 5.8: Parsing progress/completion screen (screenshot)

5.4.2 Examination workflow settings (repeatable process)

To maintain consistency during analysis, the following workflow order was applied:

1. App identification: confirm Dropbox is present and parsed.
2. Keyword anchoring: search for known ground-truth identifiers (user email, test filenames).
3. Artifact grouping: account artifacts, file artifacts, cache/offline artifacts, and timestamps.
4. Timeline supports normalize timestamps and cross-check consistency.

This approach strengthens reliability by avoiding conclusions based on a single view and instead corroborating across multiple artifact types, consistent with validation-focused forensic literature.

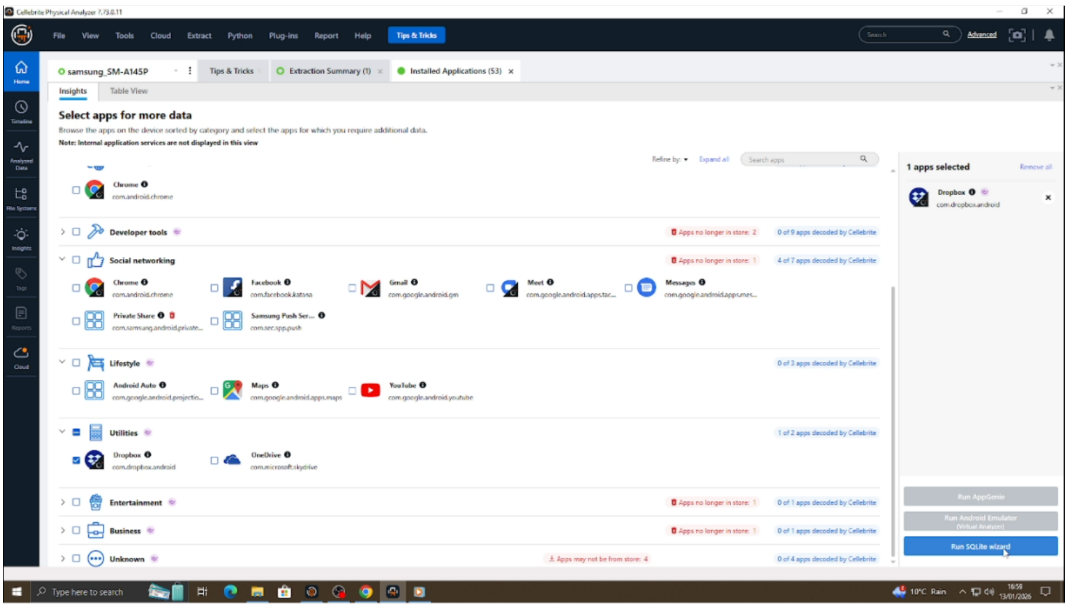
5.5 Phase 4: Dropbox Artifact Identification and Analysis

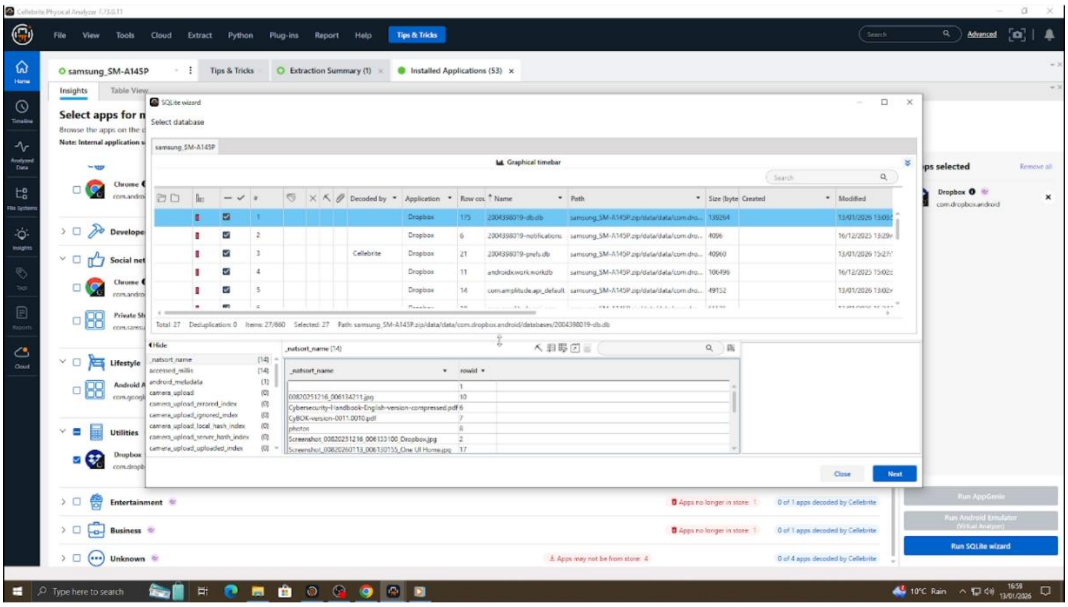
This section describes how Dropbox artifacts were located and interpreted in the extraction. Prior cloud storage research indicates that mobile Dropbox investigations commonly rely on application databases, caches, and configuration remnants that preserve identifiers and timestamps useful for reconstruction.

5.6 Locating Dropbox data sources in the extracted file system

In Android file system extractions, Dropbox-related data is typically associated with its application package (commonly named similar to com.dropbox.android). Evidence sources may include:

- Application databases (often SQLite)
- Shared preferences / configuration files (often XML)
- Cache directories (previews, thumbnails, temporary downloads)
- Files/offline content directories (content retained for offline access)
- External app storage directories (if used by the app)





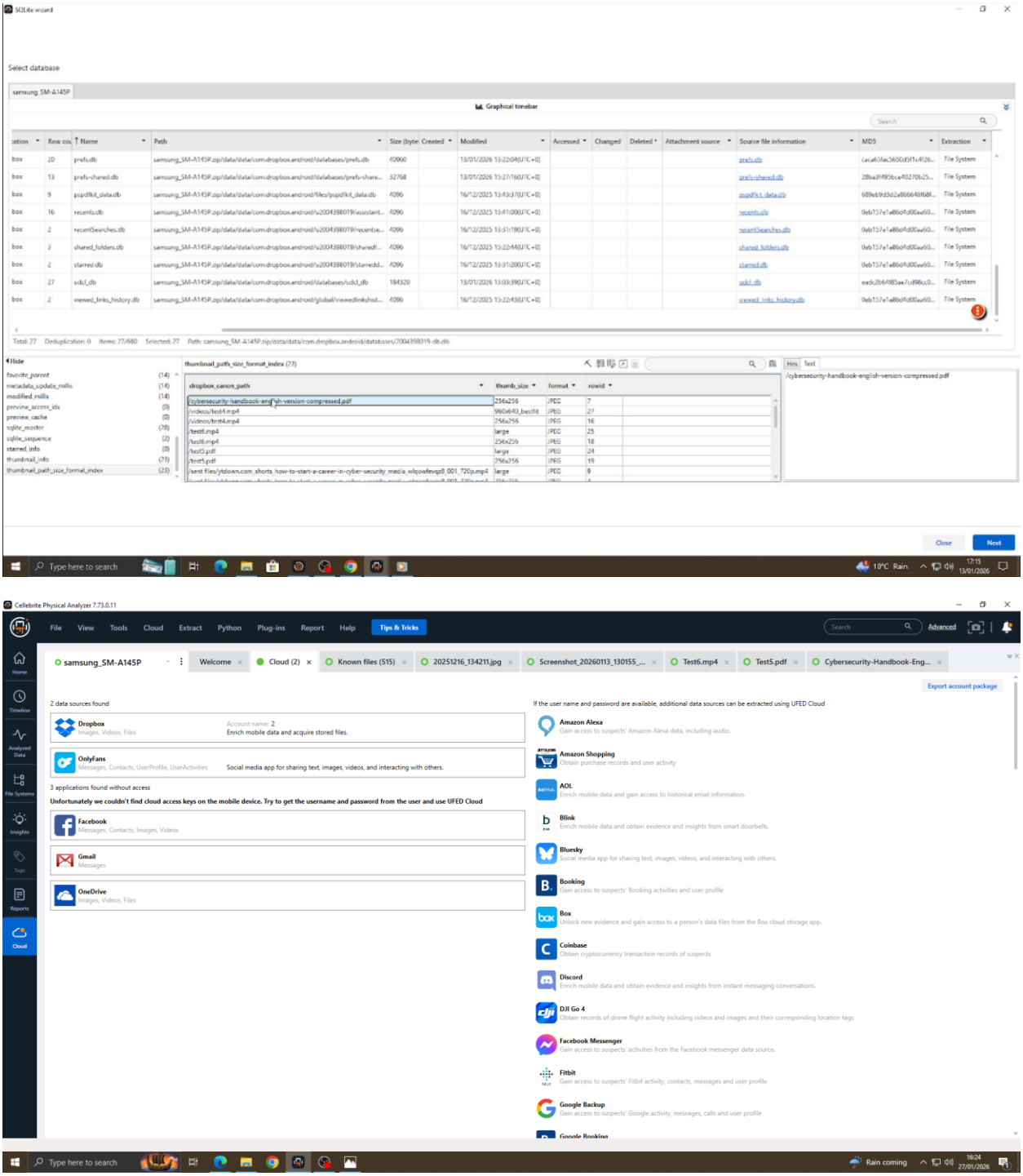


Figure 5.9: Dropbox application entry / package evidence views in Physical Analyzer

Because app versions and Android versions may alter exact file names and directory structures, this thesis reports artifacts at two levels:

1. Category level (database/cache/config/offline), and
2. File-level (specific items evidenced by screenshots and exports).

5.6.1 Account and identity artifacts (user attribution)

A primary objective was to recover account identifiers that link Dropbox activity to a user. In this study, the user email address was successfully identified within Dropbox-related artifacts. Such identifiers are important because they support attribution and can connect file activity traces to a specific account context.

Typical account evidence may include:

- user email address
- internal account identifiers
- last login timestamps or session state indicators
- device/account linkage markers

Preliminary Device Report - Google Android Generic

Cellebrite
www.cellebrite.com

Device Information

Name	Value
File System	
Chipset	MT6769V/CB
Model	SM-A145P
OS	Android 13
Security Patch Level	2023-03-01
Vendor	samsung
Mac Address	34:39:48:86:63:42
Location Services Enabled	True
Advertising ID #1	a9c95a9e-db39-4868-a433-6659f1fd860c
Bluetooth MAC Address	50:09:43:39:88:11
Factory number	R9FLJQ5WMWJ
Bluetooth device name	Galaxy A14
Android ID	9f646d8a4424af8b
Bluetooth device address	50:09:43:39:88:11
Factory Reset	13/07/2023 15:47:54
Factory Reset	25/03/2023 12:07:02

User Accounts (11)

#	Source	Account Name	User name	Service Type	Creation time	Entries
1	Accounts		uwtsdfriends@gmail.com	com.google		
2	Accounts		uwtsdfriends@gmail.com	com.dropbox.android.multiaccount		
3			vnd.sec.contact.phone	vnd.sec.contact.phone		
4			uwtsdfriends@gmail.com	com.google		
5			primary.sim.account_name	vnd.sec.contact.sim		
6			primary.sim2.account_name	vnd.sec.contact.sim2		
7			vnd.sec.contact.agg.account_name	vnd.sec.contact.agg.account_type		
8			uwtsdfriends@gmail.com	com.dropbox.android.multiaccount		
9	Dropbox	uwtsd friends	uwtsdfriends@gmail.com			
10	Accounts	uwtsd friends	uwtsdfriends@gmail.com	com.google		
11	Gmail	uwtsdfriends@gmail.com				

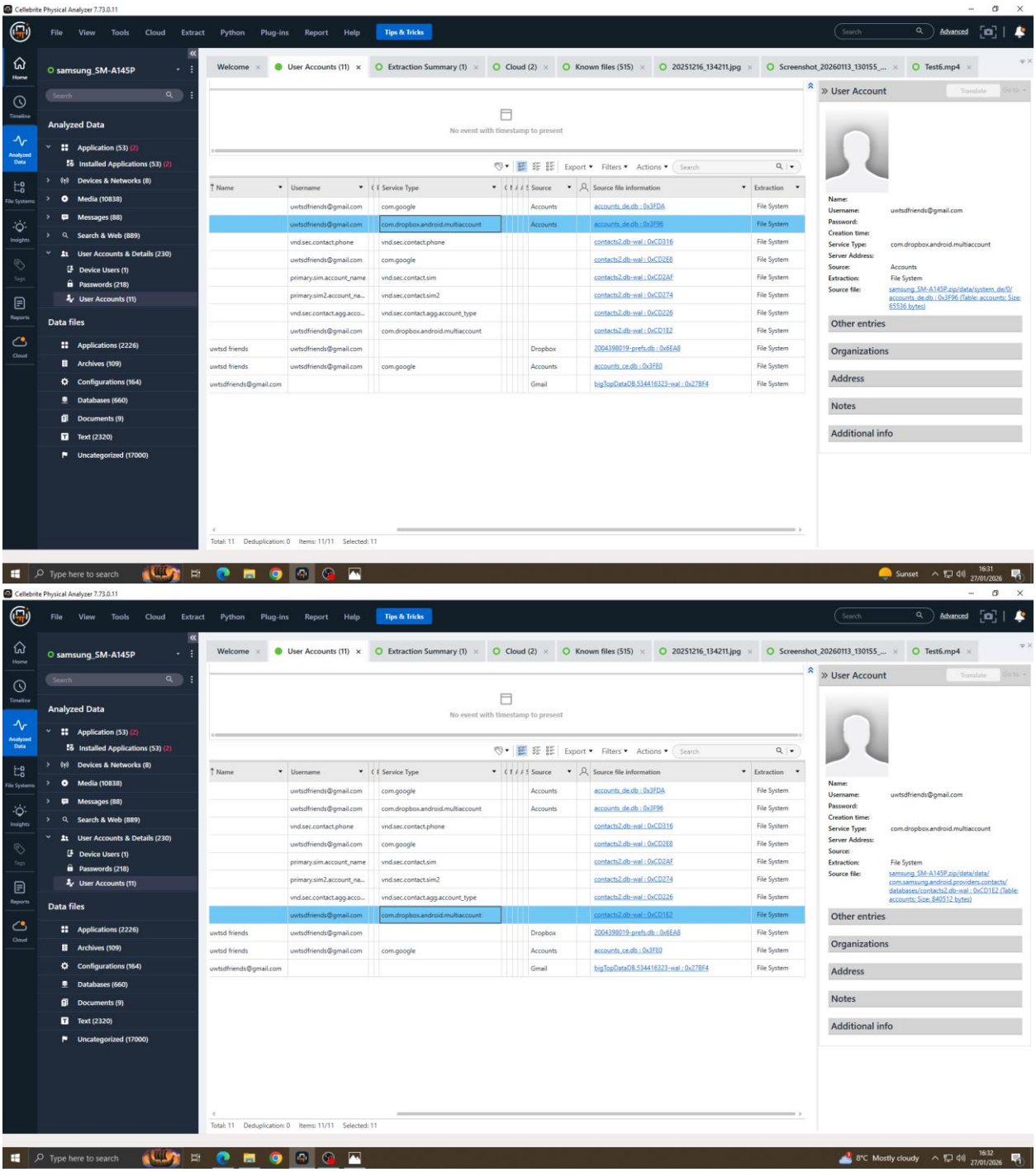


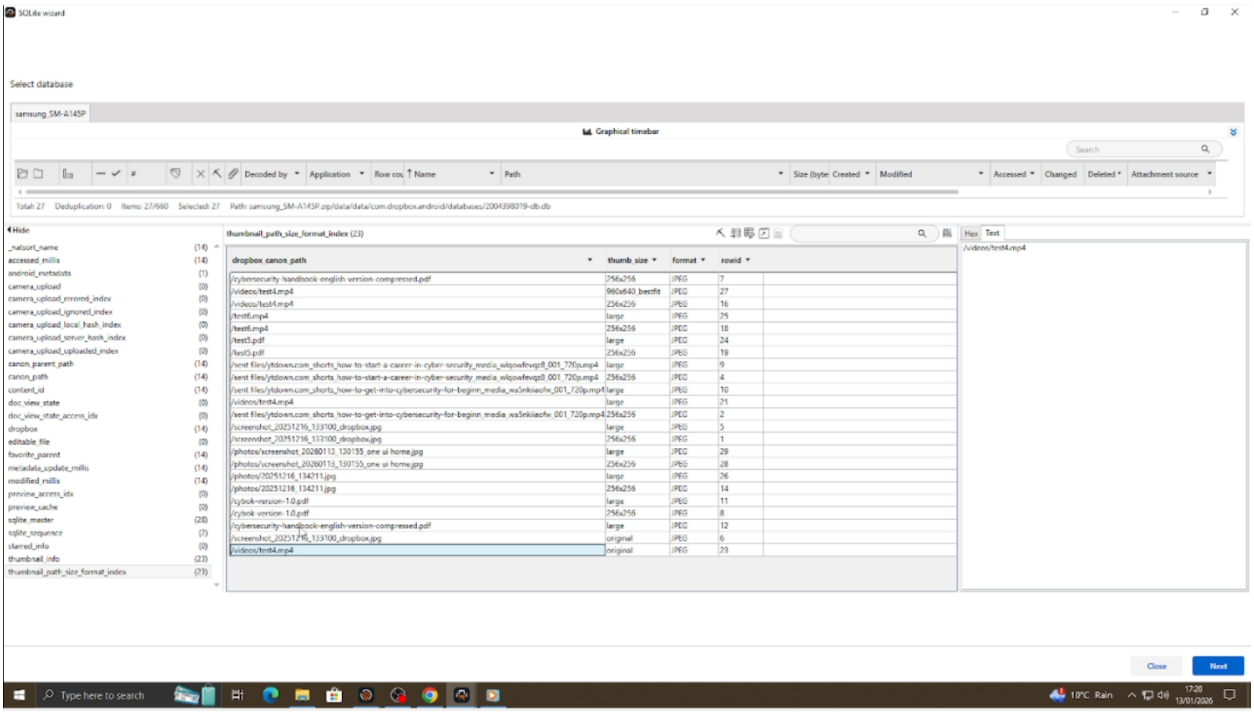
Figure 5.10: Screenshot showing recovered email identifier

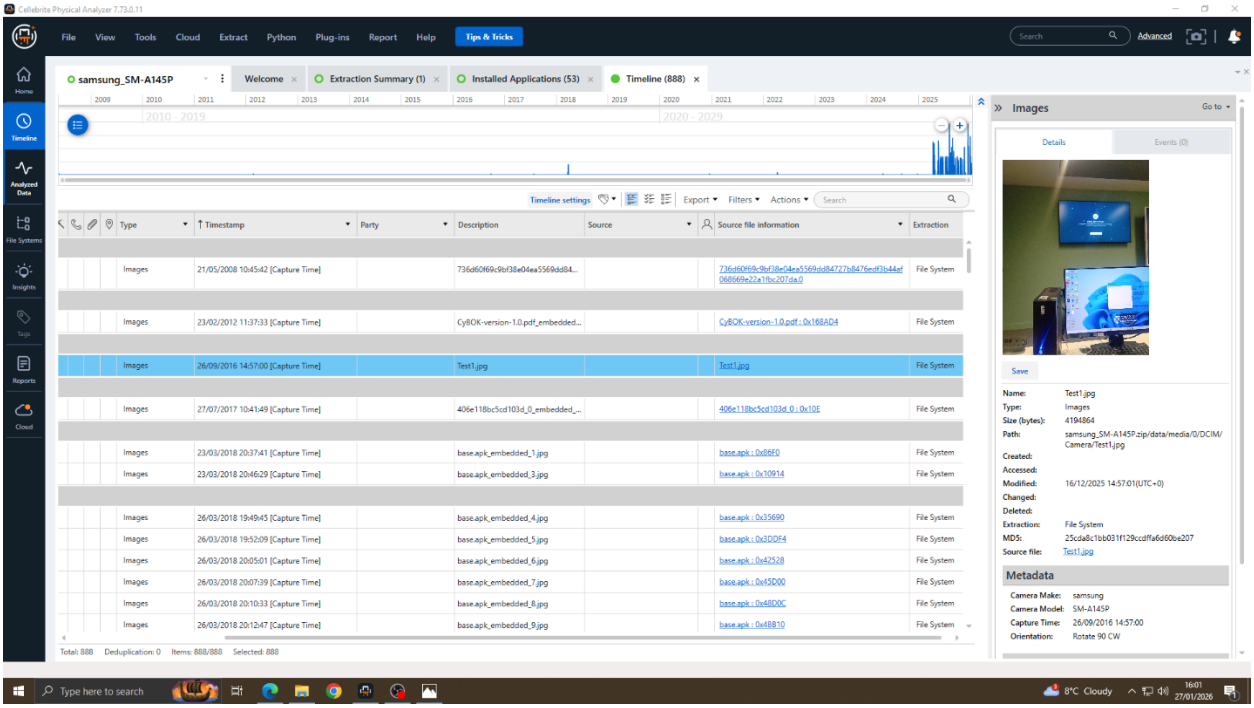
5.6.2 File metadata and activity traces (upload/download)

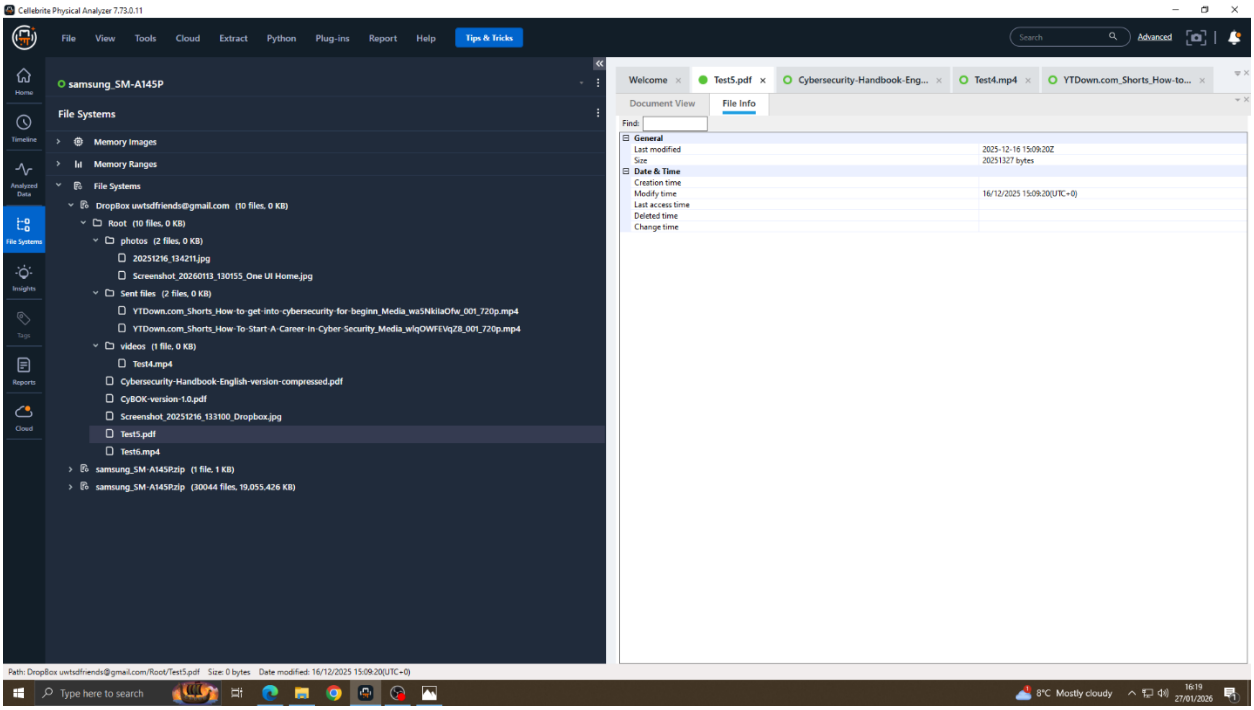
The controlled test actions included uploading and downloading files. Expected evidence types include:

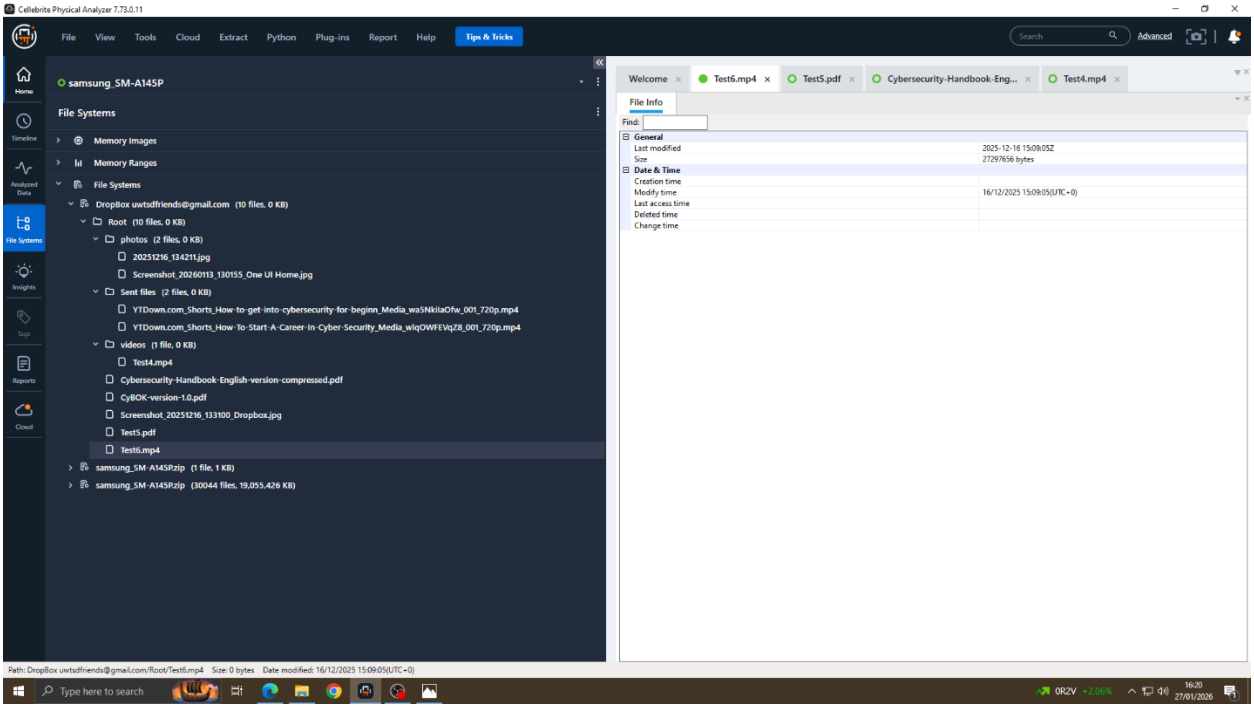
- File names (unique test filenames)
- File size and file type
- Timestamps associated with file operations (upload/download or last access)
- Local presence of downloaded files or cached copies
- Metadata linking local content to cloud items (IDs or paths)

This matches prior observations that cloud storage clients may preserve operational traces in local databases and caches that support reconstruction of user activity [40].




Cellebrite Physical Analyzer 7.73.0.11
File View Tools Cloud Extract Python Plug-ins Report Help Tips & Tricks
Advanced
samsung_SM-A145P
Welcome
Extraction Summary (1)
Installed Applications (53)
Timeline (888)
Timeline settings
Export
Filters
Actions
Type
Timestamp
Party
Description
Source
Source file information
Extraction
Images
Details
Events (0)
Save
Name: Test1.jpg
Type: Images
Size (bytes): 4194864
Path: samsung_SM-A145P.zip/data/media/0/DCIM/Camera/Test1.jpg
Created:
Accessed:
Modified: 16/12/2025 14:57:01(UTC+0)
Changed:
Deleted:
Extraction: File System
MD5: 25cda8c1bb031f129ccdffa6d60be207
Source file: Test1.jpg
Metadata
Camera Make: samsung
Camera Model: SM-A145P
Capture Time: 26/09/2016 14:57:00
Orientation: Rotate 90 CW
Total: 888 Deduplication: 0 Items: 888/888 Selected: 888
Type here to search
8°C Cloudy
16:01
27/01/2026


Cellebrite Physical Analyzer 7.73.0.11
File View Tools Cloud Extract Python Plug-ins Report Help Tips & Tricks
Advanced
samsung_SM-A145P
File Systems
Memory Images
Memory Ranges
File Systems
DropBox uwtsdfriends@gmail.com (10 files, 0 KB)
Root (10 files, 0 KB)
photos (2 files, 0 KB)
20251216_134211.jpg
Screenshot_20260113_130155_One UI Home.jpg
Sent files (2 files, 0 KB)
YTDown.com_Shorts_How-to-get-into-cybersecurity-for-beginn_Media_wa5NkiIaOfw_001_720p.mp4
YTDown.com_Shorts_How-To-Start-A-Career-In-Cyber-Security_Media_wiqOWFEVqZ8_001_720p.mp4
videos (1 file, 0 KB)
Test4.mp4
Cybersecurity-Handbook-English-version-compressed.pdf
CyBOK-version-1.0.pdf
Screenshot_20251216_133100_Dropbox.jpg
Test5.pdf
Test6.mp4
samsung_SM-A145P.zip (1 file, 1 KB)
samsung_SM-A145P.zip (30044 files, 19,055,426 KB)
Welcome
Test5.pdf
Cybersecurity-Handbook-Eng...
Test4.mp4
YTDown.com_Shorts_How-to...
Document View
File Info
Find:
General
Last modified 2025-12-16 15:09:20Z
Size 20251327 bytes
Date & Time
Creation time
Modify time 16/12/2025 15:09:20(UTC+0)
Last access time
Deleted time
Change time
Path: DropBox uwtsdfriends@gmail.com/Root/Test5.pdf Size: 0 bytes Date modified: 16/12/2025 15:09:20(UTC+0)
Type here to search
8°C Mostly cloudy
16:19
27/01/2026



Figures 5.11: Example metadata record for a test file (filename + timestamp)

5.6.3 Offline access artifacts

Offline access is especially important in Dropbox investigations because it can produce persistent local copies of cloud files. The experiment enabled offline access for selected items therefore, the analysis focuses on:

- offline flags/markers in app data (where parsed)
- local file retention consistent with offline availability
- timestamps indicating offline selection or offline sync events

Offline artifacts strengthen evidential narratives because they may retain content locally even when cloud content changes later. This aligns with the broader cloud evidence concerning that collection timing and syncing behavior can influence what is observable on the endpoint.

5.6.4 Deletion artifacts and residual traces

Deletion was included to test whether the device retained traces after file removal. Deletion outcomes can vary:

- Some apps store deletion records
- In other cases, the primary evidence may be indirect:
    - absence of previously present records,

- partial remnants in caches, or
- orphaned thumbnails/previews.

This thesis treats deletion evidence conservatively: deletion is only asserted when supported by clear state changes or corroborating traces, consistent with reliability principles.

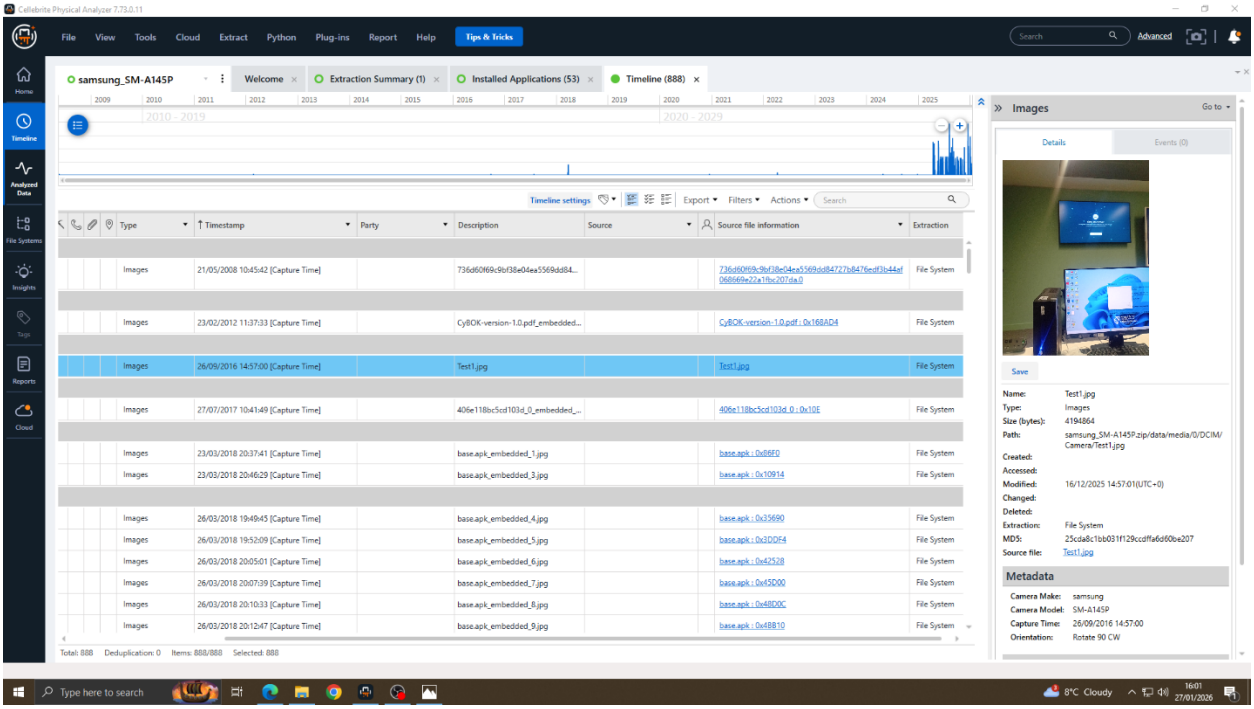


Figure 5.12: deletion-state indicator

5.6.5 Timestamp handling and timeline reconstruction

Dropbox-related timestamps are central to the research objectives because they support chronology. In this study, file timestamps were successfully found, enabling early timeline reconstruction. Timestamp analysis was performed by:

1. Extracting timestamps from parsed artifacts and file metadata.
2. Normalizing formats where required (e.g., epoch to human-readable time).
3. Comparing observed times against the known action sequence to confirm logical consistency.

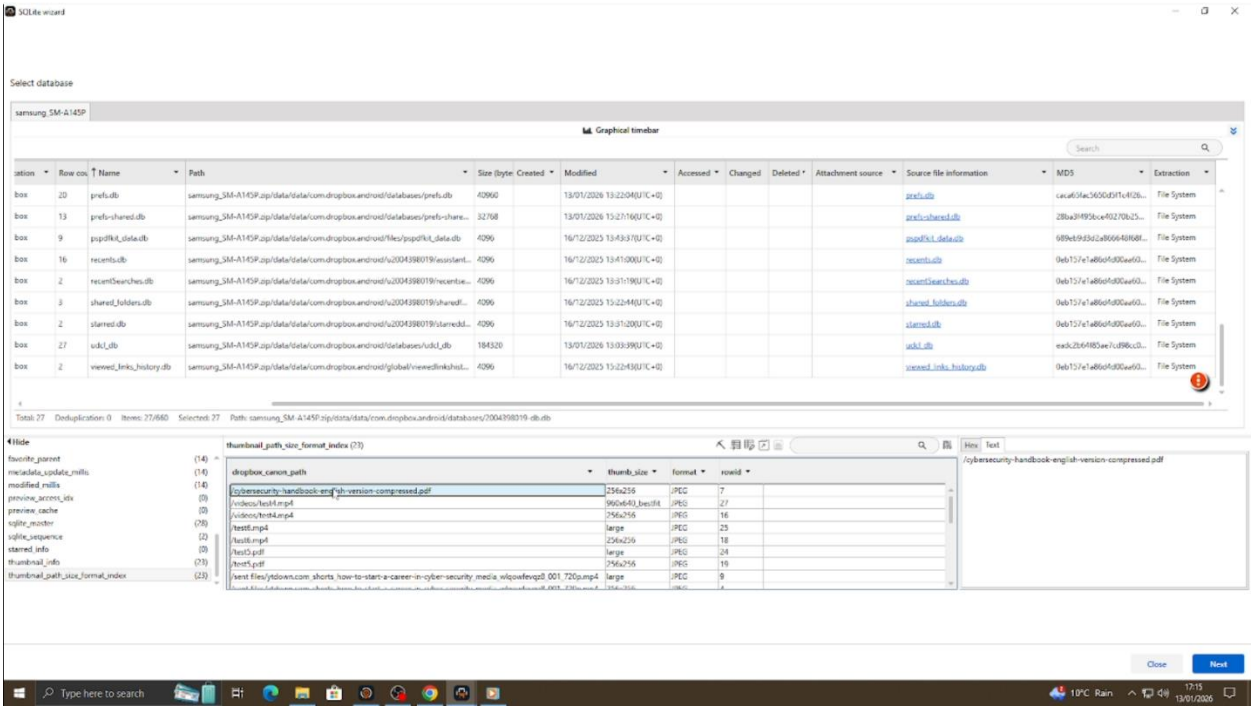
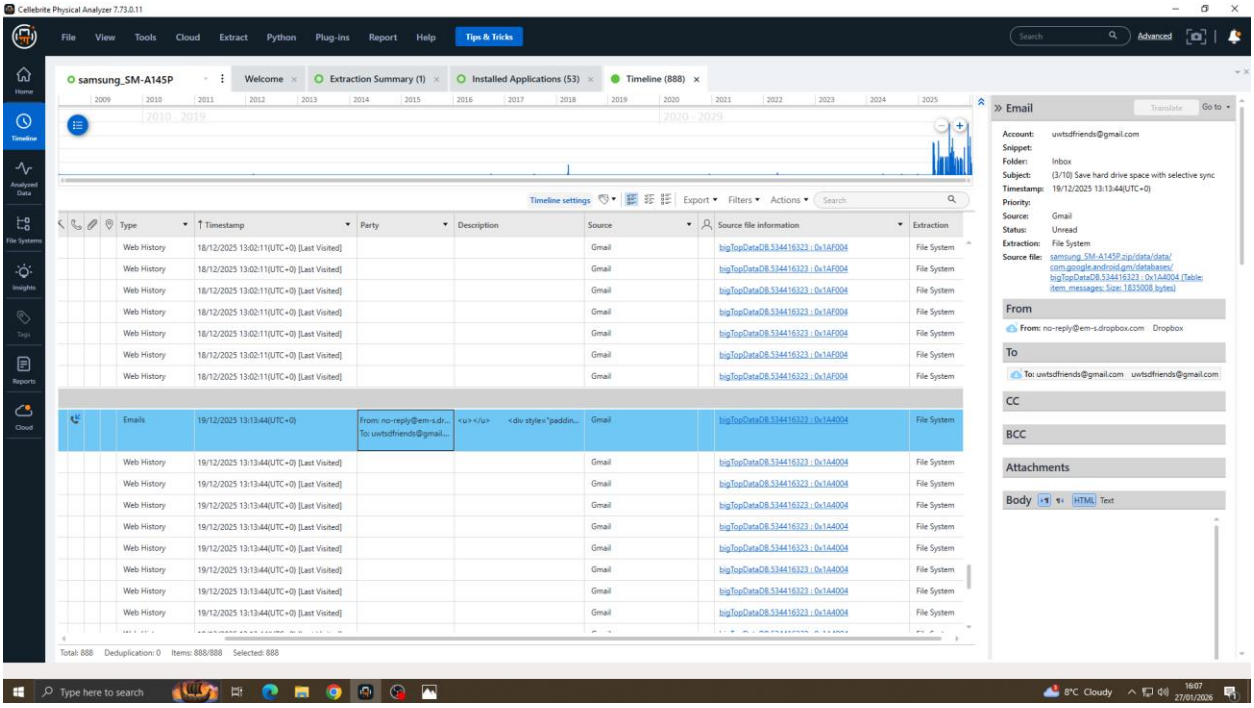
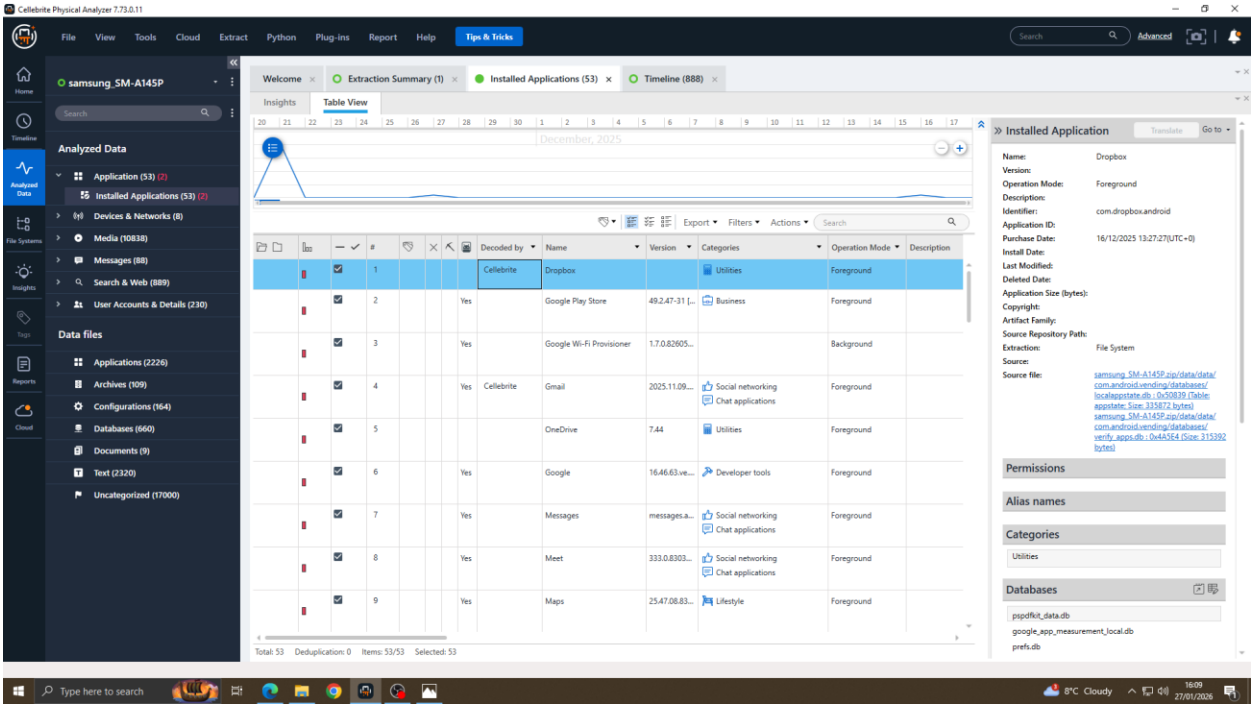


Figure 5.13: Action-to-timestamp evidence mapping





Figures 5.14: Physical Analyzer timeline view showing Dropbox-relevant events

5.7 Phase 5: Reporting and Documentation

Reporting in this framework is designed to preserve traceability from conclusions back to evidence.

H:\27-01-2026\Reports\2026-01-27.15-50-48\samsung_SM-A145P\samsung_SM-A145P_2026-01-27_Report.pdf

H:\27-01-2026\Reports\Preliminary_Device_Report_27-01-2026.pdf

5.7.1 Reporting output

The following outputs were prepared for inclusion in the dissertation appendices:

- UFED acquisition summary/export (if available)
- Physical Analyzer generated report (artifact summaries and key screenshots)
- Hash log (SHA-256 values)
- Screenshot pack showing key findings (email identifier, timestamps, file traces)

5.7.2 Traceability rule (how findings are written)

Each major finding in the results chapter (Chapter 6) is documented with:

- a short statement of the findings,

- the artifact category (DB/cache/config/file metadata),
- the supporting screenshot/export reference, and
- the relevant timestamp and identifier (where applicable).

This is consistent with the expectation that forensic conclusions be supported by identifiable artifacts and reproducible documentation.

CHAPTER 6: EVALUATION OF RESULTS

This chapter evaluates the effectiveness of the proposed forensic framework in recovering and interpreting Dropbox artifacts from a Samsung Galaxy A14 (Android 13) using Cellebrite UFED 4PC (File System extraction) and Cellebrite Physical Analyzer. The evaluation focuses on:

(i) Whether the framework supports traceability from known user actions to artifacts
(ii) The coverage of artifact categories (account, metadata, cache/offline remnants),
(iii) Integrity assurance through SHA-256 hashing
(iv) Consistency of evidence with the controlled ground-truth action sequence.

This evaluation approach aligns with standard digital/mobile forensic expectations that evidence should be collected systematically, validated for integrity, and reported with defensible reasoning [39], [40].

Dropbox was selected as the primary case study because it enables a tightly scoped but forensically meaningful endpoint investigation on Android, where evidence acquisition often depends on client-resident artifacts rather than provider-side logs. Prior comparative work confirms that cloud-storage clients (including Dropbox, OneDrive and Google Drive) differ materially in the types, locations and persistence of artifacts they generate across mobile platforms, which supports the methodological decision to focus on one application in depth for reliable action-to-artifact mapping and repeatability [41-42].In particular, Google Drive and OneDrive are typically embedded within broader vendor ecosystems and multi-app synchronization behaviors, which can expand the artifact surface and introduce additional dependencies that are outside the scope of a controlled MSc experiment, as reflected by recent large-scale studies of Google application artifacts [43]. By contrast, Dropbox presents a comparatively self-contained third-party client on Android, and peer-reviewed Dropbox-specific studies report recoverable account identifiers and activity-related traces within app stores (e.g., preference and configuration databases) that can support attribution and timeline reconstruction when cloud logging is unavailable or impractical. Therefore, choosing Dropbox balanced real-world relevance with experimental controllability, allowing this research to contribute an updated, tool-grounded framework for Dropbox artifact acquisition and interpretation on modern Android devices using Cellebrite-based workflows [44-45].

6.1 Evaluation Criteria

To measure the framework's performance, four criteria (defined during the experiment design) were applied:

1. Traceability (Action → Artifact Linkage): Ability to link each performed Dropbox action (login, upload, download, offline, delete, logout) to at least one supporting artifact.

2. Artifact Coverage:

   Breadth of recovered evidence categories, including:

   - Account/identity traces (e.g., email)
   - File metadata and timestamps
   - App databases and configuration remnants
   - Cache, preview, and offline-file remnants
     Cloud-storage forensic research indicates that such client-side artifacts are often sufficient to reconstruct user activity on endpoints.
3. Integrity Assurance: Whether evidence output remains verifiable via SHA-256 hashes, supporting non-alteration of extracted datasets during handling and analysis.
4. Consistency and Corroboration: Whether artifacts align logically with the known action sequence and whether conclusions rely on corroboration across multiple artifacts (rather than a single parsed view), consistent with reliability/validation principles.

6.2 Summary of Observed Outcomes

Based on the completed practical work and initial analysis:

- Confirmed artifacts:
  - Dropbox user email (supports user attribution)
  - File timestamps (supports activity reconstruction and chronology)
  - Dropbox SQLite databases and related tables containing file metadata and state indicators
  - Cache and preview remnants (thumbnails, temporary files)
  - Offline-access traces (local copies and/or offline markers)
  - Deletion traces

This chapter evaluates both

(i) what is already confirmed

(ii) what the framework predicts should be present and how it should be validated and reported once screenshots are added.

6.3 Action-to-Artifact Evaluation Results

Table 6.1 evaluates ground-truth action against expected artifact categories. Evidence strength is graded as:

- Strong (S): direct artifact explicitly indicating the action (e.g., account identifier, timestamped record of file operation).
- Moderate (M): indirect evidence that supports the action (e.g., local cache presence consistent with download).
- Weak (W): circumstantial evidence only (e.g., absence of content without explicit deletion marker).

Table 6.1: Action → Artifact Traceability Evaluation (Dropbox on Android 13)

Ground-truth action	**Expected artifact categories**	**Current status (this study)**	**Evidence strength**	**Notes / what to screenshot next**
Login	Account identifier (email), account/session state, config updates	Confirmed: email found	S	Screenshot: Physical Analyzer view showing email/account artifact export record reference.
Upload files	File metadata records (name/size/ID), timestamps, sync state changes	Confirmed	S	Screenshot: Search for test filenames (4 images, 2 PDFs, 3 videos). Screenshot file metadata entries and timestamps.
Download files	Local cached/download remnants, timestamps, recent activity traces	Confirmed	M	Screenshot: local file presence/cached copy list correlate with timestamps.
Offline access	Offline markers/flags, locally retained offline copies, cache persistence	Confirmed	M	Screenshot: offline-flag indicators and local offline file location/list.
Delete files	Deleted-state flag, deletion timestamp, residual cache/thumbnail remnants	Confirmed	W	Look for state changes in DB or deleted indicators otherwise show residual cache remnants supporting prior existence.

Interpretation:

- The framework already demonstrates high traceability for Login (user attribution via email).
- The presence of timestamps indicates promising support for timeline reconstruction, which typically becomes stronger once timestamps are mapped to specific file actions (upload/download/offline/delete).
- The remaining actions are expected to be supported through a combination of database records, cache remnants, and metadata - an approach consistent with how prior cloud-storage client forensic studies reconstruct user activity from endpoint artifacts.

6.4 Integrity Verification Results

Integrity was supported by hashing acquisition outputs using SHA-256, allowing verification that evidence files did not change during analysis and reporting. This is a core forensic control widely recommended for evidential defensibility.

Evaluation outcome:

- Pass: SHA-256 hashes were generated and recorded for the UFED extraction outputs (and can be extended to include exported reports and key evidence files).
- Recommendation for final submission: include a concise hash table in the appendix showing: filename, SHA-256 hash, date/time, tool used, and examiner initials.

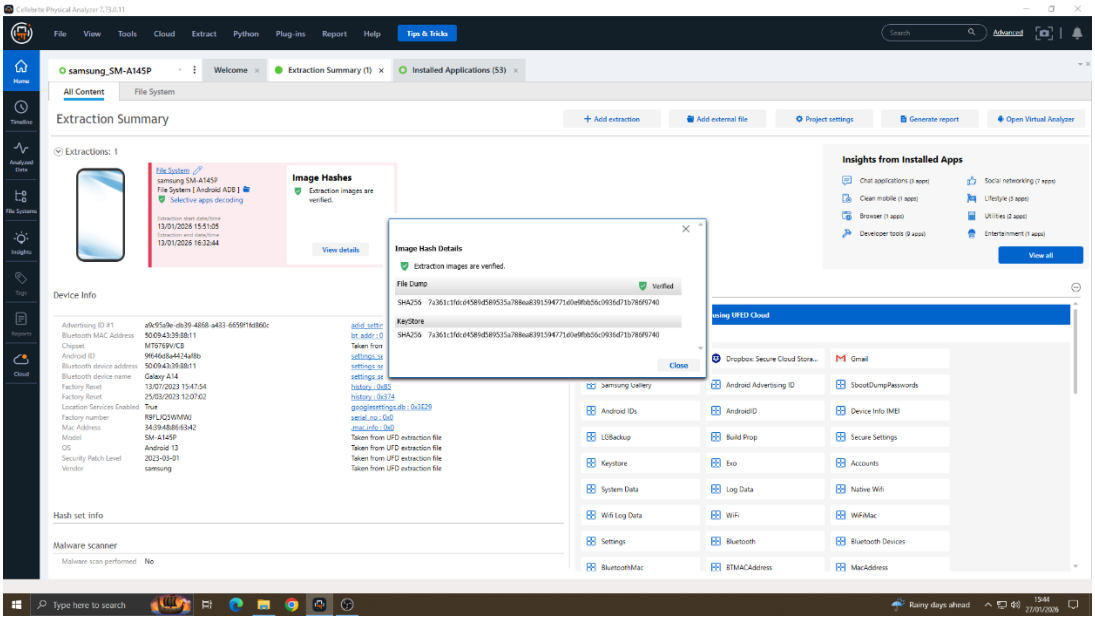


Figure 6.1: Screenshot of hash generation output

6.5 Framework Effectiveness Discussion

6.5.1 Evidence attribution (account linkage)

The confirmed recovery of the user email provides a strong attribution anchor. In a forensic context, attribution improves when identity traces (email/account ID) are corroborated with contextual artifacts such as timestamps and file interaction records. This study's confirmed account identifier supports the framework's objective of linking device activity to a specific Dropbox user context.

6.5.2 Timeline reconstruction readiness

The confirmation of file time stamps demonstrates that the dataset contains temporal artifacts necessary for reconstructing activity. The next step (already aligned with the framework) is to map timestamps to discrete events such as upload/download/offline selection and deletion. For defensible timeline reconstruction, timestamps should be corroborated across multiple sources where possible (e.g., database record time + file system metadata time), reducing dependence on a single artifact interpretation.

6.5.3 Expected artifact richness and practical reality

Mobile cloud storage apps frequently store local states to provide performance and usability features (catching, offline access, previews). These features commonly create forensic opportunities because local databases and cache directories can preserve meaningful traces of user activity even when the cloud is the primary storage location. The framework is therefore considered effective if it reliably guides the examiner to:

1. identify Dropbox storage locations in the extraction,
2. locate account identifiers and file metadata,
3. correlate caches/offline content with file actions, and
4. report conclusions with integrity controls.

6.6 Limitations and Threats to Validity

This evaluation is interpreted within the following limitations:

1. Single-device and single-OS scope: Findings are specific to Samsung A14 and Android 13 artifact location and availability may differ on other devices or Android versions.

2. App version evolution: Dropbox updates may change database schemas and cache behavior, influencing future reproducibility. This is mitigated by documenting tool versions, app version, and screenshots in the final appendices.
3. Extraction method constraints (File System extraction): File system extraction may not expose all deleted content or encrypted elements. As a result, deletion evidence may range from strong (explicit deleted record) to weak (absence or residual remnants). Conclusions regarding deletion should therefore be conservative unless directly supported by clear artifacts.
4. Parser dependence: While Physical Analyzer provides automated parsing, defensible reporting requires corroboration and clear traceability to source artifacts rather than relying solely on a single parsed summary view.

CHAPTER 7: CONCLUSIONS AND RECOMMENDATIONS

7.1 Summary of Research Outcomes

The thesis set out to produce a practical, repeatable framework that helps investigators locate and interpret Dropbox-related evidence on Android devices in a defensible manner. Across Chapters 1-6, the work progressed from research framing and literature grounding to methodology, experiment design, implementation, and evaluation.

Key achieved outcomes include:

- A framework-based forensic workflow tailored to mobile cloud storage investigations, structured into phases: preservation, acquisition, artifact identification, analysis/reconstruction, and reporting/validation. This aligns with common forensic lifecycle guidance emphasizing systematic handling and reporting.
- A controlled experimental scenario (login, upload, download, offline access, deletion, logout) designed to generate known ground truth actions for traceability assessment.
- Successful File System acquisition using UFED 4PC, with evidence integrity supported through SHA-256 hashing, enabling later verification that evidence outputs were not altered during handling or analysis.
- Confirmed recovery of user attribution evidence (email identifier) and timestamps, supporting the framework's ability to anchor user identity and enable chronological reconstruction of Dropbox-related activity.

While only certain artifacts have been confirmed at this stage (email and timestamps), the framework provides a structured route for continuing artifact discovery (e.g., databases, cache/offline remnants, and deletion indicators) and integrating screenshots/exports for final evidential presentation.

7.2 Answers to Research Questions

RQ1: What Dropbox-related forensic artifacts can be recovered from an Android 13 device following typical user actions (login, upload, download, offline access, deletion, logout)? This study confirmed recovery of:

- Account identifier evidence (Dropbox user email), supporting user attribution.
- Timestamp evidence associated with file-related activity, supporting chronology and timeline reconstruction.

Based on the experiment design and established cloud-storage client forensic literature, additional artifacts are expected to be available in many cases (e.g., app databases, cached previews, offline

content remnants, and metadata records). However, the degree of recoverability can vary by app version, device security state, and extraction visibility, so the framework treats these as artifact targets for verification rather than guaranteed outputs.

RQ2: How can a structured forensic framework support systematic acquisition, preservation, and interpretation of Dropbox artifacts using Cellebrite UFED 4PC and Physical Analyzer?

The framework improved systematic handling in three ways:

1. It defined explicit phases and decision points (preservation → acquisition → analysis → reporting),
2. It required integrity controls (SHA-256 hashing) and traceability documentation, and
3. It guided analysis through an action-to-artifact mapping model, reducing ad-hoc searching and improving repeatability.

This aligns with mobile forensic guidance recommending structured processes and clear reporting to support defensibility.

RQ3: To what extent can recovered artifacts enable timeline reconstruction and activity attribution on the device under a repeatable experimental design?

The confirmed recovery of the user email provides a strong attribution anchor, and timestamps provide the core prerequisite for timeline reconstruction. Under the framework, the next strengthening step is correlating timestamps with specific file events (upload/download/offline/delete) through corroboration across multiple sources (e.g., database record + file metadata + cache presence), which improves reliability compared to relying on a single artifact view. In summary, the framework enables attribution and timeline reconstruction to a meaningful extent, with further strengthening possible as more artifacts are documented and cross-validated.

7.3 Contribution of the Study

Within its defined scope, this research contributes:

1. A practical forensic framework for mobile cloud storage investigations focused on Dropbox artifacts on Android.
2. An experiment-driven mapping approach (actions → expected artifacts) that supports structured, repeatable examinations.
3. An integrity-first workflow combining UFED extraction and SHA-256 hashing to strengthen evidence defensibility and auditability [3].

This contribution is especially relevant for environments where cloud provider-side acquisition is not feasible, and endpoint artifacts become the primary practical evidence source.

7.4 Recommendations for Forensic Practice

Based on the implementation and evaluation, the following recommendations are proposed for practitioners investigating Dropbox usage on Android devices:

1. Prioritize preservation to reduce sync contamination: Minimize network activity and avoid unnecessary interaction with the Dropbox app after suspected actions. Background sync can change local artifacts and complicate timelines.
2. Use File System extraction where feasible and document limitations: On modern Android devices with encryption and hardened security, File System extraction often provides the best balance between depth and practicality. Investigators should document what the chosen extraction can and cannot access.
3. Anchor analysis with identifiers and ground truth searching: Start with strong anchors such as the user email and known filenames, then expand into related databases, metadata fields, and caches. This reduces false associations and improves traceability.
4. Corroborate conclusions across multiple artifacts: Avoid relying solely on a single parsed record. For example, when asserting download occurred, seek supporting traces such as a database record, a cache/offline copy, and consistent timestamps. Corroboration strengthens reliability.
5. Implement strong integrity controls (SHA-256) throughout: Generate SHA-256 hashes for the UFED output and for key exports/reports so that evidence of integrity can be demonstrated at every stage.

7.5 Recommendations for Future Work

To extend this thesis and increase academic and operational impact, future work could include:

1. Expand device/OS diversity: Repeat the framework on multiple Android devices and versions (e.g., Android 11–14) to evaluate portability and differences in artifact persistence.
2. Artifact deep-dive and schema validation: Identify exact Dropbox database file names, schemas, and key tables under the tested app version, then validate interpretations through controlled repeat runs and cross-checking of parsed vs. raw SQLite records.

3. Deletion and recovery-focused testing: Conduct repeated delete/restore cycles and measure whether deletion states persist in databases, whether cache remnants remain, and how long artifacts survive over time and app restarts.
4. Comparative acquisition methods: Compare logical vs. file system extraction (and any available advanced methods) to quantify differences in artifact visibility and evidential strength, especially for offline caches and deleted traces.
5. Optional cloud-side correlation: Where institutional permissions and legal access allow, correlate endpoint artifacts with Dropbox account-side logs/metadata to strengthen event confirmation and reduce uncertainty in attribution.

7.6 Closing Statement

This thesis demonstrated that a structured, integrity-focused forensic framework can guide the investigation of Dropbox usage on Android devices using commercially deployed forensic tools. The confirmed recovery of user attribution evidence (email) and timestamps indicates that meaningful Dropbox artifacts can be extracted and used to support activity reconstruction in a controlled scenario. With further artifact documentation and corroboration, the framework can be strengthened into a reusable investigative model for mobile cloud storage cases, while maintaining transparency about limitations arising from platform security, app updates, and acquisition boundaries.

CHAPTER 8: REFLECTION

This chapter reflects on the overall research journey, the practical and academic learning outcomes, and the challenges encountered while completing this MSc dissertation. The reflection focuses on how the project developed skills in mobile device forensics, cloud-related artifact investigation, evidence integrity handling, and forensic reporting. It also considers how methodological choices (such as the framework-based approach and file system extraction) influenced the research outcomes and what improvements would be made if the work were repeated.

8.1 Personal Learning and Skills Development

Completing this dissertation significantly strengthened my technical and professional understanding of mobile digital forensics and mobile cloud investigations. Prior to this research, my familiarity with forensic tools was limited. Through the practical work with Cellebrite UFED 4PC and Physical Analyzer, I gained hands-on competence in the full forensic workflow: case setup, acquisition planning, extraction execution, and artifact analysis.

A key learning outcome was understanding the difference between acquisition types (logical vs. file system vs. physical) and how modern Android security and encryption affect what can realistically be extracted. Using file system extraction on a contemporary Android 13 device highlighted that forensic work often requires adapting to platform constraints while still producing defensible results through documentation, integrity controls, and careful reasoning.

I also developed confidence in evidential thinking. Instead of treating tool outputs as final answers, I learned to approach evidence as a set of traces that must be interpreted with caution. This included learning how to search for identifiers such as the Dropbox account email, trace file timestamps, and planning the next steps to corroborate findings using multiple artifacts. This shift from tool output to evidence-led investigation - was a major professional development outcome.

8.2 Research Planning and Methodological Reflection

The decision to adopt a framework-based approach proved valuable for both academic quality and practical feasibility. Rather than writing a purely descriptive case study, the framework structure forced the dissertation to define repeatable phases and justify each step. This helped ensure the work aligned with expectations in digital forensics research: preservation, acquisition, analysis, validation, and reporting.

One important methodological insight was that defining a controlled action sequence (login, upload, download, offline access, delete, logout) improves the clarity of analysis. It created ground truth conditions that made it easier to assess whether artifacts can be mapped to user behavior. Without that structure, it would be difficult to evaluate results consistently.

At the same time, I realized that frameworks must balance structure with flexibility. Mobile apps and Android environments change frequently, and artifact locations may differ by version. Therefore, the framework approach worked best when it focused on artifact categories (e.g., databases, caches, configuration, timestamps) and traceability rather than rigid assumptions about specific file paths.

8.3 Practical Challenges Encountered

Several challenges emerged during the practical implementation:

1. Tool learning curve: Cellebrite is powerful but complex for a beginner. Understanding the UI workflow, extraction options, and how to interpret parsed results in Physical Analyzer required time and repeated practice. This was overcome by following a consistent workflow and documenting each step.
2. Artifact uncertainty due to app and OS behavior: Dropbox and Android manage data dynamically. Some expected artifacts may not appear clearly without deeper analysis of databases or cache folders, and deletion traces may be partial or indirect. This reinforced the importance of corroboration and conservative reporting.
3. Time and scope constraints: While more artifacts were expected (caches, offline remnants, deletion indicators), completing deep artifact validation for every action is time intensive. To manage scope, the dissertation focused on building a defensible framework and confirming key anchors (email and timestamps), while leaving room for further artifact discovery supported by screenshots in appendices.
4. Evidence presentation planning: A significant challenge was planning how to present evidence clearly in a dissertation format. Tool screenshots, extracted records, and timeline views must be integrated carefully so that the reader can trace conclusions back to artifacts. This required treating reporting as part of the forensic process rather than an afterthought.

8.4 What I Would Improve If Repeating the Project

If I were to repeat this research, I would improve the study in the following ways:

1. Deeper database-level validation: I would export Dropbox SQLite databases and validate interpretations by inspecting tables directly (in addition to relying on tool parsing). This would strengthen confidence that specific user actions are correctly mapped to artifacts.
2. Multiple experiment repetitions: Repeating the same action sequence multiple times would enable comparison of artifact consistency and improve reliability. It would also help identify which artifacts are persistent versus transient.

3. More detailed deletion analysis: I would perform systematic deletion experiments (delete, clear cache, reinstall, restore from backup) to measure which traces remain and for how long. Deletion evidence often provides high investigative value but can be difficult to confirm definitively.
4. Broader evaluation set: Extending the framework to one additional device model or Android version would help demonstrate generalizability and highlight platform-specific differences.
5. More structured evidence log: Although the project used a planned action scenario, maintaining a strict time-stamped log of every action, device state, and network condition would further strengthen evaluation and allow cleaner timeline correlation.

8.5 Academic and Professional Development Impact

This dissertation contributed to my development as a cybersecurity and digital forensics practitioner in several ways:

- It improved my ability to conduct structured investigations and document procedures systematically.
- It reinforced the importance of integrity controls (hashing) and chain-of-custody style thinking.
- It strengthened my skills in communicating technical findings clearly, which is essential in both legal and professional contexts.
- It improved my confidence using industry-standard tools and understanding their limitations.

Most importantly, the project demonstrated that real-world forensics is not only about extracting data - it requires critical judgement, careful inference, and the ability to justify conclusions using traceable evidence.

8.6 Reflection on the Thesis Topic Pivot

This research also involved an important academic decision: moving away from a cloud volatile-memory approach and pivoting to a mobile-cloud artifact framework. Reflecting on this change, the pivot was beneficial because it aligned the research with a feasible scope, accessible tools, and repeatable experimentation. It also better matched the strengths of mobile forensic toolkits and the realities of cloud investigations in many practical settings, where server-side access is often limited.

This decision improved the overall quality of the dissertation because it reduced implementation risk and enabled deeper methodological justification and practical demonstration.

This dissertation journey enhanced my technical skills, investigative mindset, and understanding of the complexities involved in mobile cloud forensics. The framework-based approach provided strong academic structure, while the practical implementation strengthened my competence with UFED 4PC and Physical Analyzer. The challenges experienced particularly around tool complexity and artifact variability reflect real-world forensic practice and reinforced the importance of careful documentation, integrity controls, and conservative conclusions. Overall, the project has contributed meaningfully to my preparation for professional work in cybersecurity and digital forensics.

References

[1] F. Daryabar, A. Dehghantanha, and K.-K. R. Choo, "Forensic investigation of OneDrive, Box, GoogleDrive and Dropbox applications on Android and iOS devices," Australian Journal of Forensic Sciences, 2016, doi: 10.1080/00450618.2015.1110620.

[2] R. Ayers, S. Brothers, and W. Jansen, Guidelines on Mobile Device Forensics, NIST SP 800-101 Rev. 1, 2014, doi: 10.6028/NIST.SP.800-101r1.

[3] R. Nordvik, R. Stoykova, K. Franke, S. Axelsson, and F. Toolan, "Reliability validation for file system interpretation," Forensic Science International: Digital Investigation, vol. 37, Art. 301174, 2021, doi: 10.1016/j.fsidi.2021.301174.

[4] D. B. Andersen, N. Sunde, and K. Porter, "Tool induced biases? Misleading data presentation as a biasing source in digital forensic analysis," Forensic Science International: Digital Investigation, vol. 52, Art. 301881, 2025, doi: 10.1016/j.fsidi.2025.301881.

[5] K. Kent, S. Chevalier, T. Grance, and H. Dang, *Guide to Integrating Forensic Techniques into Incident Response (NIST SP 800-86)*, 2006. doi: 10.6028/NIST.SP.800-86.

[6] R. Ayers, S. Brothers, and W. Jansen, *Guidelines on Mobile Device Forensics (NIST SP 800-101 Rev. 1)*, 2014. doi: 10.6028/NIST.SP.800-101r1.

[7] A. Al-Dhaqm, S. A. Razak, R. A. Ikuesan, V. R. Kebande, and K. Siddique, "A Review of Mobile Forensic Investigation Process Models," *IEEE Access*, vol. 8, pp. 173359–173375, 2020. doi: 10.1109/ACCESS.2020.3014615.

[8] C. Anglano, M. Canonico, and M. Guazzone, "The Android Forensics Automator (AnForA): A Tool for the Automated Forensic Analysis of Android Applications," *Computers & Security*, vol. 88, 2020. doi: 10.1016/j.cose.2019.101650.

[9] A. Azfar, K.-K. R. Choo, and L. Liu, "Forensic Taxonomy of Android Social Apps," *Journal of Forensic Sciences*, 2016. doi: 10.1111/1556-4029.13164.

[10] A. Azfar, K.-K. R. Choo, and L. Liu, "An Android Communication App Forensic Taxonomy," *Journal of Forensic Sciences*, 2017. doi: 10.1111/1556-4029.13267.

[11] K. Fukami et al., "A New Model for Forensic Data Extraction from Encrypted Smartphones," Forensic Science International: Digital Investigation, 2021. doi: 10.1016/j.fsidi.2021.301169.

[12] (Example Android encryption-forensics study) "Android Encryption Database Forensic Analysis," 2020. doi: 10.1145/3424978.3425068.

[13] N. Hughes and U. Karabiyik, "Towards Reliable Digital Forensics Investigations through Measurement Science," *WIREs Forensic Science*, 2020. doi: 10.1002/wfs2.1367.

[14] J. Brunty, "Validation of Forensic Tools and Methods: A Primer," *WIREs Forensic Science*, 2023. doi: 10.1002/wfs2.1474.

[15] R. Nordvik, R. Stoykova, K. Franke, S. Axelsson, and F. Toolan, "Reliability Validation for File System Interpretation," *Forensic Science International: Digital Investigation*, vol. 37, 2021. doi: 10.1016/j.fsidi.2021.301174.

[16] R. Stoykova and K. Franke, "Reliability Validation Enabling Framework (RVEF) for Digital Forensics in Criminal Investigations," *Forensic Science International: Digital Investigation*, vol. 45, 2023. doi: 10.1016/j.fsidi.2023.301554.

[17] V. Roussev, Y. Chen, T. Bourg, and G. G. Richard, "md5bloom: Forensic Filesystem Hashing Revisited," *Digital Investigation*, vol. 3, suppl., pp. 82–90, 2006. doi: 10.1016/j.diin.2006.06.012.

[18] G. C. Kessler, "The Impact of MD5 File Hash Collisions on Digital Forensic Imaging," *Journal of Digital Forensics, Security and Law*, vol. 11, no. 4, pp. 129–138, 2016. doi: 10.15394/jdfsl.2016.1431.

[19] M. D. Guido, J. Buttner, and J. Grover, "Rapid Differential Forensic Imaging of Mobile Devices," *Digital Investigation*, vol. 18, suppl., pp. S46–S54, 2016. doi: 10.1016/j.diin.2016.04.012.

[20] P. Mell and T. Grance, *The NIST Definition of Cloud Computing (NIST SP 800-145)*, 2011. doi: 10.6028/NIST.SP.800-145.

[21] S. Simou, C. Kalloniatis, S. Gritzalis, and H. Mouratidis, "A Survey on Cloud Forensics Challenges and Solutions," *Security and Communication Networks*, vol. 9, no. 18, pp. 6285–6314, 2016. doi: 10.1002/sec.1688.

[22] B. Manral, G. Somani, K.-K. R. Choo, M. Conti, and M. S. Gaur, "A Systematic Survey on Cloud Forensics Challenges, Solutions, and Future Directions," *ACM Computing Surveys*, vol. 52, no. 6, pp. 124:1–124:38, 2020. doi: 10.1145/3361216.

[23] H. Chung, J. Park, S. Lee, and C. Kang, "Digital Forensic Investigation of Cloud Storage Services," *Digital Investigation*, vol. 9, no. 2, pp. 81–95, 2012. doi: 10.1016/j.diin.2012.05.015.

[24] D. Quick and K.-K. R. Choo, "Forensic Collection of Cloud Storage Data: Does the Act of Collection Result in Changes to the Data or Its Metadata?" *Digital Investigation*, vol. 10, no. 3, pp. 266–277, 2013. doi: 10.1016/j.diin.2013.07.001.

[25] V. Roussev and S. McCulley, "Forensic Analysis of Cloud-Native Artifacts," *Digital Investigation*, vol. 16, suppl., pp. S104–S113, 2016. doi: 10.1016/j.diin.2016.01.013.

[26] M. Faheem, R. A. Shaikh, B. J. d'Auriol, and M. Mehdi, "Mobile Cloud Forensic: An Analysis of Forensic Procedures in Mobile Cloud Environment," *International Journal of Digital Crime and Forensics*, vol. 7, no. 2, 2015. doi: 10.4018/IJDCF.2015040101.

[27] M. Faheem, R. A. Shaikh, B. J. d'Auriol, and M. Mehdi, "Mobile Cloud Forensic Framework (MCFF)," in *INTECH*, 2016. doi: 10.1109/INTECH.2016.7845142.

[28] F. Daryabar, A. Dehghantanha, and K.-K. R. Choo, "Forensic Investigation of OneDrive, Box, GoogleDrive and Dropbox Applications on Android and iOS Devices," *Australian Journal of Forensic Sciences*, 2016. doi: 10.1080/00450618.2015.1110620.

[29] B. Martini and K.-K. R. Choo, "Mobile Cloud Forensics: An Analysis of Seven Popular Android Apps," in *The Cloud Security Ecosystem*, 2015. doi: 10.1016/B978-0-12-801595-7.00015-X.

[30] G. B. Satrya, J. Arifin, and J. E. Istiyanto, "Digital Forensics Study of a Cloud Storage Client: A Dropbox Artifact Analysis," *CommIT (Communication and Information Technology) Journal*, vol. 13, no. 2, pp. 57–66, 2019. doi: 10.21512/commit.v13i2.5781.

[31] B. Martini and K.-K. R. Choo, "An Integrated Conceptual Digital Forensic Framework for Cloud Computing," *Digital Investigation*, vol. 9, no. 2, pp. 71–80, 2012. doi: 10.1016/j.diin.2012.07.001.

[32] J. Dykstra and A. T. Sherman, "Acquiring Forensic Evidence from Infrastructure-as-a-Service Cloud Computing: Exploring and Evaluating Tools, Trust, and Techniques," *Digital Investigation*, vol. 9, suppl., pp. S90–S98, 2012. doi: 10.1016/j.diin.2012.03.001.

[33] K. Kent, S. Chevalier, T. Grance, and H. Dang, *Guide to Integrating Forensic Techniques into Incident Response (NIST SP 800-86)*, 2006, doi: 10.6028/NIST.SP.800-86.

[34] R. Ayers, S. Brothers, and W. Jansen, *Guidelines on Mobile Device Forensics (NIST SP 800-101 Rev. 1)*, 2014, doi: 10.6028/NIST.SP.800-101r1.

[35] F. Daryabar, A. Dehghantanha, and K.-K. R. Choo, "Forensic investigation of OneDrive, Box, GoogleDrive and Dropbox applications on Android and iOS devices," *Australian Journal of Forensic Sciences*, 2016, doi: 10.1080/00450618.2015.1110620.

[36] G. B. Satrya, "Digital Forensics Study of a Cloud Storage Client: A Dropbox Artifact Analysis," *CommIT (Communication & Information Technology) Journal*, vol. 13, no. 2, pp. 57–66, 2019, doi: 10.21512/commit.v13i2.5781.

[37] V. Roussev, Y. Chen, T. Bourg, and G. G. Richard, "md5bloom: Forensic filesystem hashing revisited," *Digital Investigation*, vol. 3, suppl., pp. 82–90, 2006, doi: 10

[38] R. Ayers, S. Brothers, and W. Jansen, *Guidelines on Mobile Device Forensics (NIST SP 800-101 Rev. 1)*, 2014, doi: 10.6028/NIST.SP.800-101r1.

[39] K. Kent, S. Chevalier, T. Grance, and H. Dang, *Guide to Integrating Forensic Techniques into Incident Response (NIST SP 800-86)*, 2006, doi: 10.6028/NIST.SP.800-86.

[40] F. Daryabar, A. Dehghantanha, and K.-K. R. Choo, "Forensic investigation of OneDrive, Box, GoogleDrive and Dropbox applications on Android and iOS devices," *Australian Journal of Forensic Sciences*, 2016, doi: 10.1080/00450618.2015.1110620.

[41] F. Daryabar, A. Dehghantanha, and K.-K. R. Choo, "Forensic investigation of OneDrive, Box, GoogleDrive and Dropbox applications on Android and iOS devices," *Australian Journal of Forensic Sciences*, 2016. doi: 10.1080/00450618.2015.1110620

[42] G. B. Satrya, “Digital Forensics Study of a Cloud Storage Client: A Dropbox Artifact Analysis,” *CommIT (Communication and Information Technology) Journal*, vol. 13, no. 2, pp. 57–66, 2019. doi: 10.21512/commit.v13i2.5781

[43] G. B. Satrya, “Identifying artefact on Microsoft OneDrive client to support Android forensics,” *International Journal of Electronic Security and Digital Forensics*, 2017. doi: 10.1504/IJESDF.2017.10005630

[44] V. Roussev and S. McCulley, “Forensic analysis of cloud-native artifacts,” *Digital Investigation*, 2016. doi: 10.1016/j.diin.2016.01.013

[45] J. Park et al., “A comprehensive artifact analysis of Google applications on Android and iOS platforms,” *Forensic Science International: Digital Investigation*, vol. 55, 2025. doi: 10.1016/j.fsidi.2025.302029

Bibliography

[1] R. Ayers, S. Brothers, and W. Jansen, *Guidelines on Mobile Device Forensics (NIST SP 800-101 Rev. 1)*, 2014, doi: 10.6028/NIST.SP.800-101r1.

[2] R. Nordvik, R. Stoykova, K. Franke, S. Axelsson, and F. Toolan, "Reliability validation for file system interpretation," *Forensic Science International: Digital Investigation*, vol. 37, Art. 301174, 2021, doi: 10.1016/j.fsidi.2021.301174.

ss

Appendix

Primary Device Report

Preliminary_Device_
Report_27-01-2026.p

Forensics Report

samsung_SM-A145P
_2026-01-27_Report.