

University of Wales Trinity Saint David

MSc Cybersecurity and Digital Forensics

ACFA7003 – Master's Project

**Mobile Device Forensics on Modern Android Devices: A Controlled
Comparative Evaluation of Cellebrite UFED and Autopsy for the
Acquisition and Recovery of Deleted Messages, Media and
Application Data**

Student Name: Vaishnavi Riteshkumar Patel

Enrolment Number: 2503493

Submission Date: 25th September 2026

Contents

Chapter 1: Introduction	2
1.1 Background and context	2
1.2 Problem statement	2
1.3 Research gap	3
1.4 Aim	3
1.5 Objectives	3
1.6 Research questions	4
1.7 Scope and delimitations	4
1.8 Significance and contribution	4
1.9 Structure of the dissertation	5
Chapter 2: Literature Review.....	5
2.1 Review strategy	5
2.2 Digital evidence: principles and standards.....	6
2.3 Android architecture from a forensic perspective	7
2.4 Android security evolution and forensic impact	7
2.5 Deletion semantics and data persistence	8
2.6 Acquisition techniques: taxonomy and trade-offs.....	9
2.7 Application artefact research.....	10
2.8 Forensic tool validation literature	10
2.9 Anti-forensics and data concealment	11
2.10 Synthesis and research gap	11
Chapter 3: Research Methodology	12
3.1 Philosophy and approach	12
3.2 Variables	13
3.3 Test environment	13
3.4 Device matrix.....	14
3.5 Ground-truth dataset construction.....	14
3.6 Deletion protocol	15
3.7 Acquisition protocol.....	15
3.8 Analysis protocol	16
3.9 Evaluation framework.....	16
3.10 Validity, reliability and repeatability	17

3.11 Legal, ethical and data protection considerations	17
3.12 Methodological limitations	18
Chapter 4: Experimental Work and Acquisition Outcome.....	18
4.1 Overview	18
4.2 Device, shared acquisition and test environment	19
4.3 Ground-truth population and deletion	20
4.4 Commercial chain.....	20
4.5 Open-source chain	22
4.6 Causes of the acquisition failure	24
4.7 Steps taken to resolve the failure.....	25
4.8 Deviation log.....	25
Chapter 5: Results	26
5.1 Acquisition outcome matrix	26
5.2 Commercial chain output.....	26
5.3 Open-source chain output	27
5.4 Messaging application artefacts.....	27
5.5 Scoring against the ground truth.....	28
5.6 Metadata fidelity	28
5.7 Integrity and efficiency	29
Chapter 6: Discussion	29
6.1 Answering the research questions.....	29
6.2 Why access, not tool capability, set the result.....	30
6.3 The two chains compared at the logical level	30
6.4 Comparison with prior studies	30
6.5 Evidential implications	31
6.6 Limitations	31
Chapter 7: Conclusion	32
7.1 Objectives revisited.....	32
7.2 Contributions	32
7.3 Recommendations for practitioners	32
7.4 Future work	32
7.5 Closing remark	33
References.....	33

Mobile Device Forensics on Modern Android Devices: A Controlled Comparative Evaluation of Cellebrite UFED and Autopsy for the Acquisition and Recovery of Deleted Messages, Media and Application Data

Chapter 1: Introduction

1.1 Background and context

A single handset now holds a person's messages, their movements, their payments and every photograph they have taken since they last changed phone. That is why it is seized ahead of the laptop. The concentration of all of it into one encrypted volume is what makes mobile forensics a distinct problem rather than a subset of disk forensics. Vinayagam's review of the mobile forensic process describes the same consolidation and notes that the built-in encryption of modern handsets poses significant challenges for investigators [1]. The tool landscape has grown in response. Ntshangase et al. survey the platforms available for Android and iOS and show a field split between a small number of commercial suites and a wider set of open-source utilities of uneven maturity [2].

Two of those options dominate practice in the United Kingdom. Cellebrite UFED, with UFED Physical Analyzer as its analysis component, is the commercial extraction platform most police units and most university forensic laboratories hold under licence. Autopsy, built on The Sleuth Kit and extended for Android through ALEAPP, is the open-source alternative that appears in academic work and in resource-limited investigations. The two are not equivalent. One is a closed system whose parsing logic cannot be inspected, the other is auditable but depends on the examiner to supply an image it can read. Investigators frequently have access to only one of them, and the question of what a case loses in that situation has not been answered with measured evidence.

The specific problem this study addresses is deleted data. A recovered live message is easy to verify and rarely disputed. A recovered deleted message is the item that changes a case and the item most open to challenge, because its recovery depends on storage behaviour that neither the examiner nor the tool controls.

1.2 Problem statement

Android has hardened in ways that were designed to protect users and that incidentally reduce what a forensic examiner can obtain. Four changes account for most of that effect:

- File-based encryption replaced full-disk encryption and split user data into credential-encrypted and device-encrypted classes, so the state of the device at seizure now determines whether most application data is readable at all.
- Verified boot and hardware-backed key storage restrict the modifications that once made physical acquisition routine.
- Scoped storage narrowed the paths an application may write to, which changes where media artefacts appear.
- Flash translation layers issue TRIM commands that release deleted blocks for erasure before any examiner reaches the device.

The evidence that these changes bite is empirical rather than theoretical. Blankesteijn, Fukami and Geradts tested retail Android 11 and 12 handsets after a factory reset and measured what remained,

working on real hardware rather than an emulated target [3]. Their results show that persistence is now conditional on device, version and reset behaviour rather than being a general property of flash storage.

Two mechanisms carry most deleted user content, and they fail in different ways:

- Messages live in SQLite databases, where deletion usually means unlinking a record rather than overwriting it, so recovery depends on freelist pages, unvacuumed regions and write-ahead log contents. Lee et al. classify the available techniques as metadata-based, carving-based and write-ahead-log-based, benchmark four public tools against each other, and identify false positives produced when a database is restructured after deletion [4].
- Media files depend instead on the file system. Oh and Hwang show that existing tools recover comparatively little deleted file data from F2FS, which is the file system used by a large share of current Android handsets, and propose a metadata-carving algorithm to close part of that gap [5].

The consequence for an examiner is that a missing artefact has at least two possible causes, and the tool cannot distinguish between them on its own. When a tool reports nothing, an examiner cannot currently say whether the data was destroyed by the platform or missed by the software. Without a known answer to compare against, no report can separate the two, and that ambiguity is the practical problem this study addresses.

1.3 Research gap

Comparative tool studies exist, but they are built in a way that prevents them from answering the question above. Mehta et al. compare Autopsy against Belkasoft X and Magnet Axiom on Android and iOS images and report which artefacts each platform surfaced [6]. Farnan et al. benchmark Autopsy and The Sleuth Kit against a commercial platform on artefact identification [8]. Wangchuk et al. compare commercial and open-source tools on deleted file recovery from a Linux image [15]. In each case the comparison is between tools, and the more capable tool sets the reference. Nothing in the design allows a reader to know what both tools missed, and no precision figure is reported, because a false positive cannot be identified without an independent record of what was actually there.

Adjacent work goes part of the way. Ismail and Zainol Ariffin ran triplicate experiments, computed error rates for Autopsy and ProDiscover against two commercial tools used as control references, and interpreted those rates against the Daubert admissibility factors [7]. Their reference was still a commercial tool, however, so an item missed by both the control and the tested tool could not be detected. Geus, Ottmann and Freiling took the same position for acquisition, testing whether a local backup yields a correct copy of the original storage by systematic comparison, and flagging corner cases such as database files with pending changes [12]. Both studies show that error can be measured once a reference exists, but neither applies a fixed ground truth to deleted data on Android.

What has not been done is to apply that design to deleted data on current Android hardware, comparing a commercial chain against an open-source chain across several acquisition methods, with the ground truth fixed before deletion. Heath, MacDermott and Akinbi provide a useful expectation baseline for messaging recovery across WhatsApp, Snapchat and Telegram [9], and Sudiana et al. show that meaningful results are obtainable on an unrooted handset under NIST SP 800-101r1 [14], but neither study scores both tool chains against a manifest. That is the gap this dissertation occupies.

1.4 Aim

To evaluate, under controlled and repeatable laboratory conditions, the extent to which deleted messages, multimedia and application data can be recovered from current Android devices, and to compare a

commercial forensic suite against an open-source workflow using a ground-truth dataset that permits error rates to be computed.

1.5 Objectives

- **O1.** Review Android security mechanisms, deletion semantics and forensic standards, and evidence the research gap. Addressed in Chapter 2.
- **O2.** Design a repeatable methodology built on a hashed ground-truth dataset, with evaluation metrics defined before any data is collected. Addressed in Chapter 3.
- **O3.** Acquire the device matrix by logical, file system, full file system and, where feasible, physical methods. Addressed in Chapter 4.
- **O4.** Analyse each image with both tool chains and score recovery against the ground-truth manifest. Addressed in Chapters 4 and 5.
- **O5.** Quantify the effect of Android version, encryption state and root status on recovery. Addressed in Chapter 5.
- **O6.** Derive practitioner recommendations and a tool-corroboration protocol. Addressed in Chapters 6 and 7.

1.6 Research questions

- **RQ1.** Which classes of deleted data remain recoverable on current Android devices, and in what proportion? (O3, O4)
- **RQ2.** How does acquisition method affect the volume and fidelity of recovered deleted data? (O3)
- **RQ3.** How do the commercial and open-source chains compare on recovery, precision, metadata fidelity, integrity and time? (O4)
- **RQ4.** How far do Android version, encryption state and root status explain variation in outcomes? (O5)
- **RQ5.** What do the measured error rates imply for evidential weight and admissibility? (O6)

1.7 Scope and delimitations

The study covers Android handsets only. Ntshangase et al. show that the Android and iOS tool landscapes diverge enough that a single comparison would have to hold neither platform steady [2], and the number of devices available does not support two platforms. iOS is therefore excluded and named as future work.

Acquisition is limited to methods a laboratory examiner can perform without destroying the device. The following are excluded, each for a stated reason:

- **Chip-off and JTAG.** destructive, and outside the equipment available.
- **Passcode bypass, exploit-based unlocking and cloud account acquisition.** outside the ethical approval for this project, and each would introduce a variable the study cannot control.
- **A third tool platform.** adding one would improve coverage at the cost of the replicate runs that make error estimates meaningful. Only two chains are compared: Cellebrite UFED with UFED Physical Analyzer, and Autopsy with The Sleuth Kit and ALEAPP.
- **Real user data.** none is acquired, examined or stored at any point. All content on the test devices is synthetic and created for this study.

- **Root-level and secure deletion techniques.** deletion is performed through the normal user interface only, because the aim is to reproduce what a device user would do rather than what an examiner could do.

Finally, the study measures recovery under laboratory conditions. It does not model an adversary who actively destroys evidence, and it does not claim that its figures transfer to devices with heavy long-term use.

1.8 Significance and contribution

The empirical contribution is a set of measured recovery and precision figures for deleted messages, media and application data on current Android hardware, produced with both tool chains against the same images and scored against the same manifest. Precision in particular is absent from the comparative literature reviewed above, and it is the figure that matters most when a report is challenged.

The methodological contribution is a scoring design that treats content and metadata separately, uses a control device to detect false positives, and runs each acquisition in triplicate. Yoo, Park and Jeong measured hash changes across repeated full file system acquisitions of the same Android handset and showed that bit-for-bit repeatability cannot simply be assumed on a live device [11]. Replicate runs are the honest response to that finding. Gruber, Hargreaves and Freiling define contamination as the inadvertent transfer of traits at any stage of the forensic and show that error can be introduced at the analysis stage as well as during acquisition [10]. Their work informs the isolation controls and the deviation log used here.

The practical contribution is a corroboration protocol. If the two chains recover different subsets of the same deleted data, then using one alone leaves evidence on the device, and the protocol tells a practitioner when the second chain is worth the additional time.

1.9 Structure of the dissertation

Chapter 2 reviews the standards, the Android security and storage mechanisms that govern persistence, the acquisition taxonomy, and the tool validation literature, closing with the comparison matrix that establishes the gap. Chapter 3 sets out the methodology, including the device matrix, the ground-truth dataset, the deletion and acquisition protocols and the evaluation framework. Chapter 4 records the experimental work as performed, including deviations. Chapter 5 presents the results. Chapter 6 interprets them against the research questions and the prior literature. Chapter 7 revisits the objectives, states the contributions and identifies future work.

Chapter 2: Literature Review

2.1 Review strategy

The review was conducted as a structured search rather than an exhaustive systematic review, on the grounds that the question is narrow and the relevant primary literature is small. Four databases were searched: IEEE Xplore, ScienceDirect, the ACM Digital Library and Google Scholar. Two repositories were searched directly for grey literature and preprints, namely arXiv and the DFRWS proceedings archive.

Search strings combined a platform term, a data term and a tool or method term. Representative strings were “Android AND deleted AND (recovery OR carving)”, “SQLite AND deleted record AND forensic”, “mobile forensic tool AND (validation OR comparison)”, “F2FS AND forensic”, and “file-based

encryption AND Android AND acquisition”. Standards documents were retrieved directly from the issuing bodies rather than through the databases.

Inclusion required that a study be published from 2019 onward, address Android or a storage mechanism used by Android, and report either an empirical result or a validation framework. Exclusion removed studies testing only emulated devices where the conclusion depended on the emulation, studies on Android 9 or earlier where the finding is superseded by file-based encryption, and studies reporting tool output with no description of method. Three exceptions to the date rule were retained because they define terminology still in use, and these are identified where they appear.

Screening followed a PRISMA-style sequence recorded in Appendix H. Titles and abstracts were screened first, full texts were then read, and studies were retained only where the method section was detailed enough for the result to be interpreted. The final set comprises fifteen research sources and a set of standards and vendor documentation. The number is small, and that scarcity is itself part of the argument in section 2.10.

One limitation of the strategy should be stated at the outset. Commercial tool vendors do not publish their parsing logic, so no peer-reviewed account of how UFED Physical Analyzer decodes a given artefact exists. The review therefore treats the commercial chain as a black box whose behaviour can be characterised only by measuring its output, which is the position taken by every study in this field and the reason experimental comparison is necessary.

2.2 Digital evidence: principles and standards

Four instruments govern the handling of digital evidence in the United Kingdom, and together they set the constraints this methodology must satisfy.

The ACPO Good Practice Guide for Digital Evidence, now maintained under NPCC, states four principles [16]:

1. No action should change data held on a device that may later be relied on in court.
2. Where a person finds it necessary to access original data, that person must be competent to do so and must be able to explain the implications of their actions.
3. An audit trail should be created and preserved such that an independent third party can repeat the process and reach the same result.
4. The case officer holds overall responsibility for compliance.

The second principle is the one that matters most for mobile work. A live Android handset cannot be write-blocked in the way a hard disk can, because acquisition requires the device to run and respond. Every extraction therefore proceeds under principle two rather than principle one, and the burden shifts to documentation and to the examiner’s ability to account for changes.

ISO/IEC 27037 provides the international counterpart, covering identification, collection, acquisition and preservation of digital evidence, and introducing the roles of first responder and specialist [17]. ISO/IEC 17025 governs the competence of testing laboratories and is the standard against which United Kingdom digital forensic units are accredited [18]. The Forensic Science Regulator’s statutory Code of Practice makes method validation a requirement rather than good practice, and requires that a method be shown fit for purpose before it is used in casework [19]. The distinction the Code draws between validating a method and trusting a tool is central to this dissertation and is developed in section 2.8.

NIST SP 800-101 Revision 1 remains the most widely cited procedural reference for mobile device forensics, setting out preservation, acquisition, examination, analysis and reporting phases and describing the acquisition levels used in section 2.6 [20]. Sudiana et al. apply it directly to an unrooted Android

handset, which demonstrates that the framework still structures current work [14]. Its age is a limitation, since it predates file-based encryption and scoped storage, and the review treats it as a process model rather than a technical reference.

The NIST Computer Forensics Tool Testing programme supplies the model this study adapts [21]. CFTT tests a tool against a specification using a reference dataset with known contents, reports the results as measured deviations, and does not rank products. The value of that approach is that it produces a statement about what a tool did rather than a statement about which tool is better. Three features of the CFTT model are carried into Chapter 3: a reference dataset whose contents are known in advance, a test procedure fixed before execution, and reporting of anomalies whether or not they favour the tool.

2.3 Android architecture from a forensic perspective

Android is layered, and each layer both holds evidence and constrains how that evidence can be reached.

At the application layer, an application's private data lives under `/data/data/<package>` or `/data/user/0/<package>`, typically as SQLite databases, XML shared preferences and cache files. This is where deleted messages persist or fail to persist. Access to this path is the single most important determinant of what an examination can produce, and it is unavailable to a logical acquisition on an unrooted device.

The framework layer exposes content providers, of which the SMS/MMS provider and MediaStore are the two that matter here. A logical acquisition works through these interfaces, which means it receives what the framework is willing to report rather than what is on disk. Deleted records are generally invisible at this layer, which is the structural reason logical acquisition is expected to underperform in this study.

The native library and Android Runtime layer contributes evidence indirectly, through cached artefacts and through the behaviour of the SQLite library that most applications use for storage. The library's write-ahead logging mode, discussed in 2.5, is a property of this layer rather than of the individual application.

The hardware abstraction layer and the Linux kernel expose the block devices and the file system. This is the layer a physical or full file system acquisition targets, and it is also where verified boot and the flash translation layer sit. Vinayagam's review of mobile forensic tools notes that Autopsy alone is not sufficient for deleted item recovery on mobile targets, an observation this study tests directly [1].

The partition layout matters for a practical reason. The `/data` partition holds user and application data and is the target of interest. The `/system` and `/vendor` partitions are largely static and are useful mainly for establishing device state and build identity. Newer devices place a substantial part of user-visible media in emulated storage backed by `/data/media`, so media recovery and application data recovery are not as separable as older literature assumes.

2.4 Android security evolution and forensic impact

The move from full-disk encryption to file-based encryption is the single change with the largest forensic consequence. Under full-disk encryption, one key protected the whole user partition and the device was readable once booted and unlocked. File-based encryption encrypts files individually under keys tied to different protection classes [22]. Credential-encrypted storage is protected by a key derived from the user's credential and is unavailable until the user has unlocked the device at least once since boot. Device-encrypted storage is available after boot without the credential and holds only what the system needs to function before unlock.

This produces the distinction between the two states an examiner encounters. Before First Unlock, credential-encrypted data is cryptographically inaccessible regardless of the tool used, and no amount of parsing sophistication changes that. After First Unlock, the keys are resident in memory and credential-encrypted data is readable to a process with sufficient privilege. Almost all application content, including messaging databases, is credential-encrypted. The practical result is that device state at seizure, not tool capability, sets the ceiling for the examination. This study controls for state explicitly by acquiring in a defined state and recording it for every run.

Verified boot enforces cryptographic verification of the boot chain and refuses to load a modified boot image without an explicit unlock that triggers a data wipe [23]. This closes the route by which examiners previously loaded a custom recovery to obtain a physical image, and it is why physical acquisition is conditional rather than routine in the device matrix of section 3.4. Hardware-backed key storage, including StrongBox where a discrete secure element is present, keeps key material off the application processor and outside the reach of a software-based extraction.

Scoped storage changed where applications may write. From Android 10 and enforced from Android 11, an application has unrestricted access only to its own directory and to media it created, with other access mediated by the framework [24]. For the examiner this narrows the set of locations where an application's media artefacts appear and increases the importance of MediaStore entries and thumbnails as secondary evidence when the original file is gone.

Blankesteijn, Fukami and Geradts provide the empirical measure of what these mechanisms leave behind, testing retail Android 11 and 12 devices after a factory reset [3]. Their finding, that remnant survival depends on the specific device and its reset implementation rather than on a general property of the platform, supports the decision in this study to vary Android version across the device matrix rather than treating Android as a single condition.

2.5 Deletion semantics and data persistence

Deletion on Android is not one operation. It is at least four, and each leaves a different trace.

The first is application-level soft deletion. Many messaging and gallery applications mark a record as deleted or move it to a recycle location while leaving the underlying row intact. Recovery here is trivial and requires only that a tool parse the correct column. Heath, MacDermott and Akinbi found recoverable content for ephemeral messages across WhatsApp, Snapchat and Telegram on mobile, desktop and cloud, which indicates how often the application layer retains what the user believes has gone [9].

The second is SQLite record deletion. When a row is deleted, SQLite normally unlinks the record and adds the page or cell to a freelist rather than zeroing the bytes. The record content frequently remains until the space is reused or the database is vacuumed. Lee et al. survey the recovery techniques built on this behaviour and group them into three families [4]. Metadata-based recovery reads the freelist and unallocated cell areas and reconstructs records using the table schema, which yields high-fidelity output including field boundaries. Carving-based recovery searches for serial-type patterns without relying on schema and recovers fragments that metadata methods miss, at the cost of precision. Write-ahead-log recovery reads the -wal and -shm files, which under the default journaling mode of many Android applications hold committed and uncommitted pages that may include pre-deletion images of a row [25]. Their benchmark of Undark, SQLite Deleted Record Parser, Bring2Lite and FQLite shows that no single tool dominates, and their analysis of false positives arising from database restructuring is the direct justification for measuring precision in this study rather than recovery alone.

The third is file system deletion. Android devices use ext4 or F2FS for the data partition. Under ext4, deletion typically zeroes the extent tree in the inode, which removes the map from file to blocks and

makes metadata-based recovery difficult even though the data blocks may persist. F2FS is log-structured and writes updates to new locations, which in principle preserves older versions, but Oh and Hwang demonstrate that existing tools recover comparatively little deleted file data from F2FS in practice and propose a metadata-carving algorithm to address the shortfall [5]. Their result is important for interpretation in this study, because it means a media recovery failure on an F2FS device is more plausibly a tool limitation than a proof of data destruction.

The fourth is physical erasure by the storage controller. Flash memory cannot be overwritten in place, so a flash translation layer maps logical to physical blocks and garbage-collects invalid blocks. The TRIM command tells the controller which blocks the file system no longer needs, after which the controller may erase them at a time it chooses. From an examiner's position this is non-deterministic. Data may survive for hours or may be gone before the device is seized, and the elapsed time between deletion and acquisition becomes a variable. This study therefore includes an ageing condition in the deletion protocol described in section 3.6.

Two further residues deserve mention because they are frequently the only surviving evidence of a deleted image. Thumbnail caches maintained by the gallery application or by MediaStore may persist after the original file is removed, and EXIF metadata embedded in a carved fragment may establish origin and time even when the image itself is incomplete. The scoring design in Chapter 3 records recovery source for exactly this reason, because a thumbnail and a full file are not evidentially equivalent and should not be counted as the same success.

2.6 Acquisition techniques: taxonomy and trade-offs

NIST SP 800-101r1 orders acquisition methods by increasing technical difficulty and increasing risk to the device [20]. The order below follows that model and states what each method can be expected to yield for deleted data.

Manual examination means operating the device and photographing the screen. It produces no deleted data at all and is used here only to verify device state before acquisition begins.

Logical acquisition queries the device through supported interfaces such as the backup service, ADB or the content providers. It is fast, needs no privilege escalation and works on an unrooted device. It returns live records only. Geus, Ottmann and Freiling evaluated whether local backup acquisition returns a correct copy of original storage and identified corner cases including database files with pending changes [12]. Even the live output, in other words, can be incomplete if a write-ahead log has not been checkpointed. Their finding shapes the handling of -wal files in this study's analysis protocol.

File system acquisition copies the accessible file system, which on an unrooted device is a subset of /data. It sits between logical and full file system extraction and yields deleted data only where an application has retained it at the application level.

Full file system acquisition obtains the complete /data partition, including application private directories, journals and write-ahead logs. On a current device it normally requires the After First Unlock state and a vendor-specific agent or exploit. It also produces the bulk of deleted messaging evidence, which makes it the primary acquisition target in this study.

Physical acquisition produces a bit-for-bit image of the storage. On modern hardware it is conditional on verified boot state and on a bootloader-level method existing. It is also worth less than it once was: where the device uses file-based encryption, an image of an encrypted partition obtained without the keys has limited value. It is attempted where feasible in the device matrix and reported as unavailable where it is not.

Chip-off and JTAG remove or bypass the controller entirely. Both are excluded from this study on grounds of destructiveness and equipment access, and both face the same encryption problem as physical acquisition.

The trade-off structure is consistent. Methods that require less privilege return less deleted data and disturb the device less. Methods that return more deleted data require more privilege and carry higher risk of altering the exhibit. This is why the acquisition protocol in section 3.7 proceeds in ascending order of intrusiveness rather than starting with the most productive method.

2.7 Application artefact research

Messaging evidence concentrates in a small number of predictable locations. The SMS/MMS content provider stores messages in `mmssms.db` and continues to be a source of deleted records because the provider marks and unlinks rather than overwrites. Third-party messaging applications maintain their own databases within their private directories, and several apply an additional layer of local encryption to those databases, which means that obtaining the file is necessary but not sufficient. Heath, MacDermott and Akinbi document recovery of ephemeral messages across three platforms and three device classes and show that application behaviour, rather than platform policy, determines what survives [9]. Sudiana et al. examine WhatsApp disappearing messages specifically on an unrooted handset and report recovery under a NIST-structured process, which provides a directly comparable published result for the unrooted condition in this study's device matrix [14].

Media evidence has three layers. The original file in emulated storage or in an application directory is the strongest form. The MediaStore database entry persists independently of the file and can establish that an image existed, its filename, its size and its insertion time, even after the file is removed. Thumbnail caches provide a low-resolution copy that is often the last surviving image content. EXIF data, where the file or fragment retains its header, carries capture time, device model and sometimes location, and its survival is a separate question from the survival of the pixel data. This study scores these layers separately rather than recording a binary recovered or not recovered outcome.

Application data beyond messages and media includes shared preferences in XML, account records, cache directories and log files. These artefacts are rarely the object of a request but frequently establish context, such as whether an application was installed at the relevant time or which account was signed in. They are included in the ground-truth dataset so that the comparison covers the artefact classes an examiner would actually be asked about.

2.8 Forensic tool validation literature

The Forensic Science Regulator's Code requires validation of a method, not endorsement of a product [19]. The distinction is practical. A method is a defined sequence performed by a competent examiner on a defined class of exhibit, and validating it means demonstrating that the sequence produces correct results within stated limits. Treating a commercial tool as pre-validated because it is widely used substitutes reputation for evidence. That is common practice, and it does not withstand cross-examination.

Ismail and Zainol Ariffin's conceptual study of open-source forensic tools proposes a four-factor framework of capability, reliability, transparency and standards compliance, and that framework organises the remainder of this section [13]. Capability is what a tool can extract and parse. Reliability is whether it returns the same result on repeated execution. Transparency is whether its processing can be inspected and explained. Standards compliance is whether its use fits within an accredited method.

On capability, the comparative studies do not give a consistent picture. Mehta et al. compare Autopsy with Belkasoft X and Magnet Axion on Android 10, iOS 13 and iOS 14 images and report the artefact

classes each surfaced [6]. Farnan et al. compare Autopsy and The Sleuth Kit with the commercial Forensic Explorer on a disk image and find that both tools retrieve most of the available evidence [8]. Wangchuk et al. compare commercial and open-source tools, including on deleted file recovery, using a Linux image [15]. The consistent limitation across all three is that the comparison uses the more capable tool's output as the implicit reference, so a class of data that both tools missed cannot appear in the results and precision cannot be computed at all.

On reliability, Yoo, Park and Jeong measured hash value changes across repeated full file system acquisitions of the same Android handset and found inconsistencies [11]. This is not a defect in the tool so much as a property of acquiring from a live system, but it has a direct methodological consequence: a single acquisition cannot be assumed to be reproducible, and a study that acquires once cannot distinguish tool variation from device variation. Triplicate acquisition in this study follows from that result.

On transparency, the asymmetry is structural. Autopsy and The Sleuth Kit are open source, so a disputed result can be traced to the code that produced it. UFED Physical Analyzer cannot be examined in the same way, and its parsing decisions must be accepted or independently verified. Gruber, Hargreaves and Freiling make the risk concrete by defining contamination as inadvertent transfer of traits at any point in the forensic process and showing through worked examples how error can enter during analysis [10]. Their argument is that error can enter after acquisition, in the interpretation stage, where it is harder to detect than an acquisition failure.

On standards compliance, Ismail and Zainol Ariffin provide the fullest treatment, running triplicate experiments against control references, computing error rates for Autopsy against commercial tools, and mapping those rates onto the Daubert factors [7]. Their design is the closest published parallel to the one used here and is the direct model for the triplicate structure and the control reference in Chapter 3.

The convergent conclusion of this literature is dual-tool verification. If the transparent tool can be audited but recovers less, and the opaque tool recovers more but cannot be audited, then corroboration between the two is the mechanism that produces both coverage and defensibility. What the literature does not supply is a measurement of how much corroboration actually adds on current Android hardware, which is what RQ3 asks.

2.9 Anti-forensics and data concealment

This section is framed as investigative awareness. It describes categories of behaviour an examiner should recognise, and it deliberately omits operational detail.

Encryption is the most common obstacle, and on Android it is now the default rather than a user choice. Its forensic effect is described in 2.4 and does not require deliberate action by the user.

Secure deletion and wiping utilities attempt to overwrite data rather than unlink it. Their effectiveness on flash storage is limited by the flash translation layer, since the utility writes to logical addresses that the controller may map elsewhere. Blankesteijn, Fukami and Geradts measured remnants after a factory reset, which is the most common wiping action, and found that survival varied by device [3].

Application-level concealment includes ephemeral messaging, vault applications and steganography. Heath, MacDermott and Akinbi show that ephemeral messages frequently remain recoverable, so the presence of a concealment feature does not imply the absence of evidence [9].

Anti-forensic activity itself leaves traces, including installation records, timestamps clustered around a deletion event, and gaps in otherwise continuous logs. This study does not simulate an adversarial user, and section 1.7 records that as a delimitation. The relevance here is interpretive: a low recovery rate in the

results should not be read as evidence that a user acted deliberately, and the discussion in Chapter 6 keeps those explanations separate.

2.10 Synthesis and research gap

The literature reviewed above supports five positions:

1. Deleted content on Android persists conditionally rather than reliably, and persistence depends on the deletion mechanism, the file system and elapsed time [3], [4], [5].
2. Acquisition method determines what is available for analysis more strongly than tool selection does [12], [20].
3. Published tool comparisons measure tools against each other rather than against a known reference, and they do not agree on how far commercial tools lead [6], [8], [15].
4. Repeated acquisition of the same live handset does not guarantee identical output [11].
5. Validation requires a known reference, and error rates computed against that reference are what admissibility arguments rely on [7], [13], [19].

Setting the tool comparison studies side by side exposes the shared limitation. Mehta et al. cover Android and iOS with three platforms but report no ground truth and no precision figures [6]. Farnan et al. compare Autopsy and The Sleuth Kit against a commercial platform on a non-mobile target [8]. Wangchuk et al. compare commercial and open-source tools on a Linux image, so the storage stack and the artefact classes both differ from Android [15]. Ismail and Zainol Ariffin do compute error rates, but their control references are commercial tools and their targets are USB drives and laptop disks rather than deleted data on a current handset [7]. Blankesteijn, Fukami and Geradts do work on real Android hardware with a measured outcome, but they test the effect of factory reset rather than comparing tool chains [3]. Lee et al. and Oh and Hwang characterise the mechanisms in depth but evaluate specialised utilities rather than the platforms an examiner actually uses [4], [5].

No study identified in this review scores both a commercial and an open-source chain against a hashed ground-truth manifest, across multiple acquisition methods, on current Android devices, with deleted messages, media and application data treated as separate artefact classes and with precision reported alongside recovery.

That absence has a practical cost. Without precision figures, an examiner presenting a recovered deleted message cannot state a measured false positive rate for the method that produced it. Without a common reference, no one can say what both tools failed to find. Without replicate runs, variation between acquisitions is indistinguishable from variation between tools. The methodology in Chapter 3 is constructed to remove all three of those unknowns.

Chapter 3: Research Methodology

3.1 Philosophy and approach

The study adopts a post-positivist position. It assumes that a determinate fact exists about whether a given deleted item is present on a device, while accepting that measurement of that fact is imperfect and that the imperfection must be quantified rather than ignored. This position is appropriate because the object of study is not an interpretation but a physical state of storage, and because the tools used to observe that state are known to introduce error [4], [10].

The approach is deductive. Chapter 2 establishes expectations from theory and prior work, principally that acquisition method constrains recovery more than tool choice does, that SQLite-resident content survives deletion more often than file system-resident content, and that elapsed time reduces recovery through TRIM. Chapter 3 designs experiments capable of contradicting those expectations, and Chapters 4 and 5 report what was measured whether or not it matches.

The design is a controlled laboratory experiment with a quantitative-dominant strategy. Recovery rate, precision, metadata fidelity and processing time are numeric. Examiner effort and the interpretation of anomalies are qualitative and are reported as structured observations rather than converted into scores. Ismail and Zainol Ariffin adopt the same combination, running replicated experiments against control references and then interpreting error rates against a legal standard, and their study is the methodological model followed here [7].

An experiment is the correct design for this question because the alternatives cannot answer it. A survey of practitioners would report belief about tool performance rather than performance. A case study would provide depth without a ground truth. Only a controlled experiment with a known reference set permits false positives to be identified, and precision is the metric the literature is missing.

3.2 Variables

The independent variables are the four factors the study manipulates. Acquisition method takes four levels: logical, file system, full file system and physical where feasible. Tool chain takes two levels: the Cellebrite chain and the open-source chain. Android version and encryption state vary across the device matrix. Root status takes two levels, unrooted and rooted, on the one device where rooting is permissible.

The dependent variables are the measured outcomes. Recovery rate is the proportion of deleted ground-truth items recovered. Precision is the proportion of reported items that correspond to a real ground-truth item. Metadata fidelity is the accuracy of timestamp, path and attribution against the manifest. Integrity is the agreement of hash values across stages. Processing time is wall-clock duration per acquisition and per analysis. Examiner effort is recorded as the number of manual interventions required.

Controlled variables are held constant so that variation in outcome can be attributed. The ground-truth dataset content is identical across devices. Deletion is performed through the user interface only, using a fixed selection rule. The interval between deletion and acquisition is fixed per condition. Tool versions are frozen for the duration of the study and recorded in Appendix E. The workstation, cable set and connection order are the same for every run. Analysis is performed by the same person throughout; the acquisition is a shared dataset (section 4.2).

Two confounds are acknowledged rather than controlled. Device manufacturers implement storage and reset behaviour differently, so a difference between two handsets cannot be attributed to Android version alone. Flash translation layer garbage collection is not observable from outside the device, so the ageing condition measures elapsed time rather than a controlled amount of erasure.

3.3 Test environment

Acquisition and analysis are performed on a dedicated forensic workstation used for no other purpose during the study. The specification, operating system build and installed tool versions are recorded in Appendix E. The workstation is not connected to any network during acquisition.

Isolation controls follow from Gruber, Hargreaves and Freiling's account of contamination as inadvertent transfer at any stage of the process [10]. Devices are placed in flight mode and, where the acquisition method permits, in a shielded enclosure, to prevent remote wipe and to prevent inbound messages from

altering the databases under test. Each device is handled with a dedicated cable and is connected only to the forensic workstation.

The write-protection position is stated openly because it cannot be resolved. A live Android handset cannot be write-blocked, since acquisition requires the device to boot, run an agent and respond to requests. Every acquisition in this study therefore operates under ACPO principle two rather than principle one [16]. The compensating controls are a contemporaneous log of every action taken on the device, a hash of every output at the point of creation, and replicate acquisitions that make any resulting variation visible. Yoo, Park and Jeong showed that repeated full file system acquisition of the same handset can produce differing hash values, so the study reports acquisition variation as a finding rather than treating a mismatch as a procedural failure [11].

Analysis is performed on copies. Working images are hashed on creation and re-hashed after each analysis pass, and the original acquisition output is retained unmodified.

3.4 Device matrix

Four devices are used, three test devices and one control. The exact models are recorded in Appendix E once procurement is confirmed, and the matrix is designed around conditions rather than around specific handsets so that substitution does not invalidate the design.

Device A is a current-generation handset running the most recent Android version available, unrooted, with file-based encryption active and acquisition performed in the After First Unlock state. This is the baseline condition. Sudiana et al. examined unrooted Android 12 handsets [14], which gives a published point of comparison for the unrooted condition.

Device B runs an earlier Android version, unrooted, with file-based encryption active. It isolates the effect of platform version while holding root status constant.

Device C is rooted, which permits full file system and physical acquisition without vendor-specific agents. It provides an upper bound on what is recoverable when access is not the limiting factor, and the difference between Device C and Device A separates tool limitation from access limitation.

The control device is loaded with a ground-truth dataset from which nothing is deleted. Any item that either tool reports as deleted on this device is a false positive by construction. This device is what makes precision measurable, and it addresses the specific weakness identified in the comparison studies in section 2.10.

Three test devices is a small sample and the study does not claim statistical generalisation across the Android population. The matrix is designed to isolate mechanisms, and the replicate structure in section 3.10 supports claims about measurement stability rather than about population-level frequencies.

3.5 Ground-truth dataset construction

The dataset is the scoring key for the entire study, and it is constructed before any device is used.

All content is synthetic and produced for this project. No real user data, no real photographs of identifiable people and no real communications are included. This satisfies the ethical position in section 3.11 and removes any question about the provenance of recovered material.

Each item carries an embedded identifier token, formatted as a short alphanumeric string unique to the item, placed in the message body for text items and in a metadata field or an overlaid region for media items. The token makes matching deterministic. A recovered fragment either contains a known token or it

does not, which removes examiner judgement from the primary scoring decision and makes the process repeatable by a third party.

The dataset spans three artefact classes. Messages comprise SMS and MMS records and third-party messaging application content, in both directions, with a defined distribution of lengths. Media comprises images and video in more than one format, some with EXIF metadata present and some with it stripped, so that metadata survival can be assessed independently of file survival. Application data comprises shared preference entries, account records and cache items with known values.

Every item is hashed with SHA-256 before it is placed on a device, and the hashes are recorded in a manifest that also records item identifier, class, format, size, intended location, creation timestamp and whether the item is designated for deletion. The manifest is the reference against which all recovery is scored and is reproduced in Appendix B.

Population is verified before deletion. Each device is checked to confirm that every item is present and that the on-device copy hashes to the manifest value where the application has not re-encoded it. Where an application re-encodes content on receipt, which is normal for image attachments, the re-encoded version is hashed on the device and recorded as a second reference value, and the scoring rules in section 3.8 state which value applies.

3.6 Deletion protocol

Deletion is performed through the ordinary user interface of the relevant application. No root shell, no database editor and no secure deletion utility is used. The rationale is that the study models what a device user does, not what an examiner or an adversary with elevated access could do.

The items designated for deletion are selected by a fixed rule applied identically on every device, taking a stratified subset across artefact class, size and age so that the deleted set is not concentrated in one class. The deleted set is a subset of the populated set rather than an additional group of items, so live and deleted counts are not additive.

Every deletion is timestamped at the moment it is performed and the timestamp is recorded in the manifest. Deletion order is recorded, since order affects which pages are reused.

Two ageing conditions are used. In the immediate condition, acquisition begins within one hour of deletion. In the aged condition, the device remains powered and idle for a defined interval, currently set at seven days, before acquisition. The comparison between the two conditions is the study's measure of time-dependent loss and is the closest observable proxy for TRIM and garbage collection activity, which cannot be inspected directly [5].

Devices are not used for any other activity between deletion and acquisition, so that page reuse results from system behaviour rather than from new user content.

3.7 Acquisition protocol

Acquisition proceeds in ascending order of intrusiveness on every device: logical, then file system, then full file system, then physical where feasible. The order follows NIST SP 800-101r1 [20] and ensures that a method which alters the device more is never performed before a method that alters it less.

Before the first acquisition on a device, the device state is recorded. That means power state, lock state, whether the device has been unlocked since boot, Android version, build number, patch level and installed application versions. The record establishes the encryption state under which the run was performed, without which the interpretation in section 5.2 cannot be made.

Each acquisition produces an output file or directory. It is hashed with SHA-256 immediately on completion, before any analysis tool opens it. The hash, the method, the tool version, the start and end times, the output size and any warnings issued by the tool all go into the acquisition log in Appendix D.

Each acquisition is repeated three times under identical conditions. Where the three outputs do not hash identically, the difference is investigated and reported rather than resolved by selecting one run. This follows Yoo, Park and Jeong, who showed that hash inconsistency across repeated full file system extractions is a real property of live acquisition rather than a procedural fault [11]. The analysis uses the first run. Variation between runs is reported as a separate result.

Where an acquisition fails, the failure is recorded with the error returned, and no alternative method is substituted silently. Failed acquisitions are reported in the outcome matrix in section 5.1, because the set of methods that do not work on a current device is itself a finding.

A deviation log records every departure from the protocol, with the reason and the assessed effect on the result. The log is reproduced in Chapter 4 rather than omitted.

3.8 Analysis protocol

Both tool chains process identical acquisition outputs. The commercial chain is Cellebrite UFED with UFED Physical Analyzer. The open-source chain is Autopsy with The Sleuth Kit, with ALEAPP applied to Android application artefacts. Neither chain is given a hint about what to look for, and both are run with default parsing settings first, with any additional module or setting recorded as a separate pass.

Scoring is performed against the manifest using three outcome categories per item. A true positive is a reported item whose embedded token matches a manifest item designated for deletion. A false positive is a reported item that does not match any manifest item, including a reported item that matches a live item but is presented as deleted. A false negative is a manifest deleted item that neither appears in the tool output nor is recoverable from it.

Content and metadata are scored separately. An item may be recovered in full content with an incorrect timestamp, and counting that as an unqualified success would overstate evidential value. Content scoring records whether the recovered content is complete, partial or absent. Metadata scoring records, per item, whether the timestamp, the path and the attribution match the manifest.

Recovery source is recorded for every true positive, using categories of live-table record, freelist or unallocated record, write-ahead log entry, carved fragment, thumbnail, and MediaStore or provider entry without the underlying file. This is necessary because the same item recovered from a thumbnail and from a full file are not evidentially equivalent, and because the distribution across sources is what explains any difference between the two chains [4].

All matching rules are fixed and documented before scoring begins, and are reproduced in Appendix F together with the scoring script. Where a tool reports an item that requires examiner judgement to classify, the judgement is recorded with its reasoning rather than absorbed into the count.

3.9 Evaluation framework

Six metrics are defined in advance, and no metric is introduced after the results are seen.

Recovery rate is the number of true positives divided by the number of manifest items designated for deletion, reported per artefact class, per acquisition method, per device and per tool chain. Reporting a single overall figure would conceal the class-level differences that the study exists to measure.

Precision is true positives divided by the sum of true positives and false positives. It is computable only because of the control device and the token scheme, and it is the metric absent from the comparative studies reviewed in section 2.10 [6], [8], [15]. Recall is equivalent to recovery rate under the definitions above, and F1 is reported as their harmonic mean to give a single comparable figure per condition.

Metadata fidelity is reported as the proportion of recovered items with a correct timestamp, a correct path and a correct attribution, given separately rather than combined, since a timestamp error and a path error have different consequences in a report.

Integrity is reported as agreement between the hash taken at acquisition and the hash taken after analysis, and separately as agreement across the three replicate acquisitions. A failure of the first kind indicates that analysis altered the working copy. A failure of the second kind indicates acquisition variation, which is expected on a live device and is reported as a measurement rather than as an error [11].

Efficiency is wall-clock time for acquisition and for analysis, recorded per run. Examiner effort is the count of manual interventions required to reach a result, including manual database parsing, manual carving and manual interpretation of an ambiguous artefact. This distinguishes a tool that produces an answer automatically from one that produces the same answer after two hours of work.

The four factors proposed by Ismail and Zainol Ariffin, capability, reliability, transparency and standards compliance, are used to organise the comparative summary in section 5.8, mapping recovery and precision to capability, replicate agreement to reliability, auditability of the parsing decision to transparency, and the documentation trail to standards compliance [13].

3.10 Validity, reliability and repeatability

Internal validity is protected by the control device, which establishes the false positive baseline, and by the fixed deletion rule, which prevents the deleted set from being chosen in a way that favours either tool. Both chains process the same images, so any difference in output is attributable to the chain rather than to the acquisition.

Reliability is addressed by triplicate acquisition and by triplicate analysis, following the replicated design used by Ismail and Zainol Ariffin [7]. Agreement across replicates is reported as a result rather than assumed.

Repeatability is addressed by fixing every decision in advance and documenting it. Tool versions are frozen and recorded, matching rules are written before scoring, and the scoring script is included in Appendix F. An independent examiner given the manifest, the tool versions and the protocol should be able to reproduce the procedure, which is the requirement ACPO principle three imposes [16].

External validity is limited and the limitation is stated rather than argued away. Three test devices cannot represent the Android population, synthetic content does not have the fragmentation history of a device in daily use for two years, and a seven-day ageing interval is short relative to the delays common in real casework. The study's claims are therefore about mechanisms and about relative tool behaviour under controlled conditions, not about expected recovery rates in the field.

Construct validity rests on the token scheme. Because matching is determined by an embedded identifier rather than by content similarity, the measurement of recovery is the measurement of the intended construct rather than of examiner judgement.

3.11 Legal, ethical and data protection considerations

Ethical approval is obtained from the UWTSD ethics committee before any device work begins, and the approval documentation is reproduced in Appendix A.

No personal data is processed. All content is synthetic, created for the study, and contains no information relating to an identified or identifiable person, so the study falls outside the material scope of UK GDPR and the Data Protection Act 2018 [26]. Test accounts used for messaging applications are created for the study, are not linked to any real identity, and are deleted at the end of the project.

All devices are owned by or lawfully supplied to the researcher or the university, and all access is authorised by the owner. No unauthorised access to any computer material takes place, which is the requirement under the Computer Misuse Act 1990 [27]. No passcode bypass or exploit against a device the researcher does not control is attempted, which is also recorded as a scope exclusion in section 1.7.

The study is not conducted for the purposes of a criminal investigation, so PACE and its codes do not bind the work directly [28]. The ACPO principles are nonetheless followed as the professional standard of the discipline, because the study's conclusions are intended for practitioners who are bound by them [16].

Both tools are used under licence. The Cellebrite licence held by the university permits the extraction types used here. Autopsy, The Sleuth Kit and ALEAPP are used under their respective open-source licences. No attempt is made to reverse-engineer, decompile or circumvent any part of the commercial software, and the commercial chain is characterised only by its observable output.

Findings are reported whether or not they favour a tool, and no result is withheld because it is unfavourable to a vendor. Where a tool fails, the failure is described factually with the version number and the conditions under which it occurred.

3.12 Methodological limitations

Six limitations follow from the design and are stated here rather than deferred to Chapter 6.

The device sample is small. Three test devices and one control support mechanism-level conclusions but not population-level estimates, and manufacturer-specific storage behaviour cannot be separated from Android version with a matrix of this size.

Synthetic data does not reproduce the storage state of a real device. A handset used for two years has a fragmented and heavily rewritten data partition, and page reuse behaviour on such a device may differ from what is observed here.

The ageing interval is short. Seven days captures early garbage collection activity but not the months that often elapse between an event and an examination.

Physical acquisition is conditional. Verified boot and file-based encryption mean that physical acquisition may be unavailable on the unrooted devices, in which case RQ2 is answered across three methods rather than four, and the gap is reported explicitly [23].

The commercial chain cannot be audited. Where Physical Analyzer produces a result the open-source chain does not, the study can measure the difference but cannot attribute it to a specific parsing decision, which is the transparency asymmetry identified in section 2.8 and by Gruber, Hargreaves and Freiling [10].

Finally, a single examiner performs all work. This removes inter-examiner variation, which improves internal consistency, but it also means that examiner effort figures reflect one person's familiarity with the two tool chains and should be read as relative rather than absolute.

Chapter 4: Experimental Work and Acquisition Outcome

4.1 Overview

This chapter records what was done, in the order it was done, and what each step returned. The protocol in Chapter 3 assumed that full file system acquisition would be available on the unrooted baseline handset, because it is the only non-destructive method that reaches the application databases, journals and write-ahead logs in which deleted messages persist (section 2.6). On the device used, it was not available. Section 3.7 committed the study to reporting a failed acquisition as a finding rather than substituting another method silently. Sections 4.2 to 4.5 describe the work, section 4.6 explains the failure, section 4.7 lists the steps taken to resolve it, and section 4.8 is the deviation log.

4.2 Device, shared acquisition and test environment

The test handset was a Samsung Galaxy S25 Ultra, model SM-S938B/DS, owned by a fellow MSc student, Smit Patel. With the approval of the project supervisor (Appendix [X]), a single acquisition of this handset was used as a shared dataset by two dissertations with different research questions: the other study uses it to develop a framework for examining encrypted messaging applications with the Cellebrite chain, while this study uses it to compare the Cellebrite chain against an open-source chain. The acquisitions were performed by [state who performed each acquisition, e.g. "S. Patel, with the author present"]; the analysis, comparison and scoring reported in Chapters 5 and 6 were carried out by the author. The device owner consented to its use, only the seeded test corpus was examined, personal content on the device was not analysed or reported, and identifiers in the figures are redacted.

UFED reported Android 16 on a Qualcomm SM8750 chipset, 512 GB of storage with 238 GB in use, 168 installed applications and a security patch level of 5 July 2026 (Figure 4.1). The bootloader was locked and the device was not rooted. All work was carried out in the After First Unlock state with USB debugging enabled, the screen lock set to None and Stay Awake enabled, as UFED instructed. The messaging applications were WhatsApp 2.26.36.74, Telegram 12.10.3 and Signal 8.27.1, versions read from the package manager.

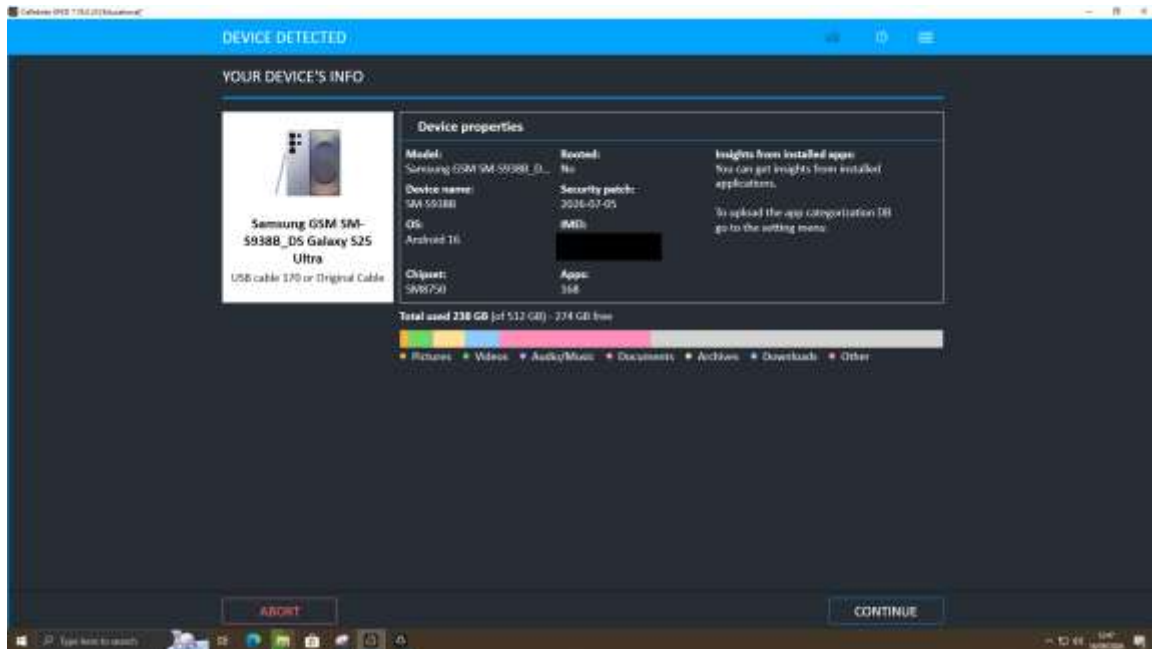


Figure 4.1: Device properties reported by UFED 4PC on 16 September 2026 (IMEI redacted).

The commercial chain was Cellebrite UFED 4PC 7.76.0.24 under an educational licence, with Physical Analyzer 7.73.0.11 for decoding. The open-source chain was Android Debug Bridge 1.0.41 (platform-tools 37.0.1), ALEAPP v2026.4.1 and Autopsy 4.23.1 with its Android Analyzer and aLEAPP modules. Output was written to a dedicated external SSD organised into acquisition, hash, ALEAPP, Autopsy and notes directories, and the Autopsy installer was hashed with SHA-256 before installation.

4.3 Ground-truth population and deletion

The handset was populated with a ground-truth corpus of 390 messages, 130 in each of WhatsApp, Signal and Telegram, each carrying an identifier token as described in section 3.5. Fourteen messages per application, 42 in total, were then deleted through the ordinary user interface under the rule in section 3.6. [INSERT: date and time of deletion and the interval to the first acquisition.] The manifest is reproduced in Appendix B. The corpus was created as part of the shared dataset and is narrower than the design in section 3.5, which also specified SMS, media and application-data items; this is recorded as deviation D2.

4.4 Commercial chain

On 16 September 2026 UFED detected the handset and offered six extraction types: Advanced Logical, File System, Physical (Rooted), Camera, Screenshot and Chat Capture (Figure 4.2). No Full File System option was offered for this model, and Physical required a rooted device. The File System type offered two modes, Android Backup and Android Backup APK Downgrade.

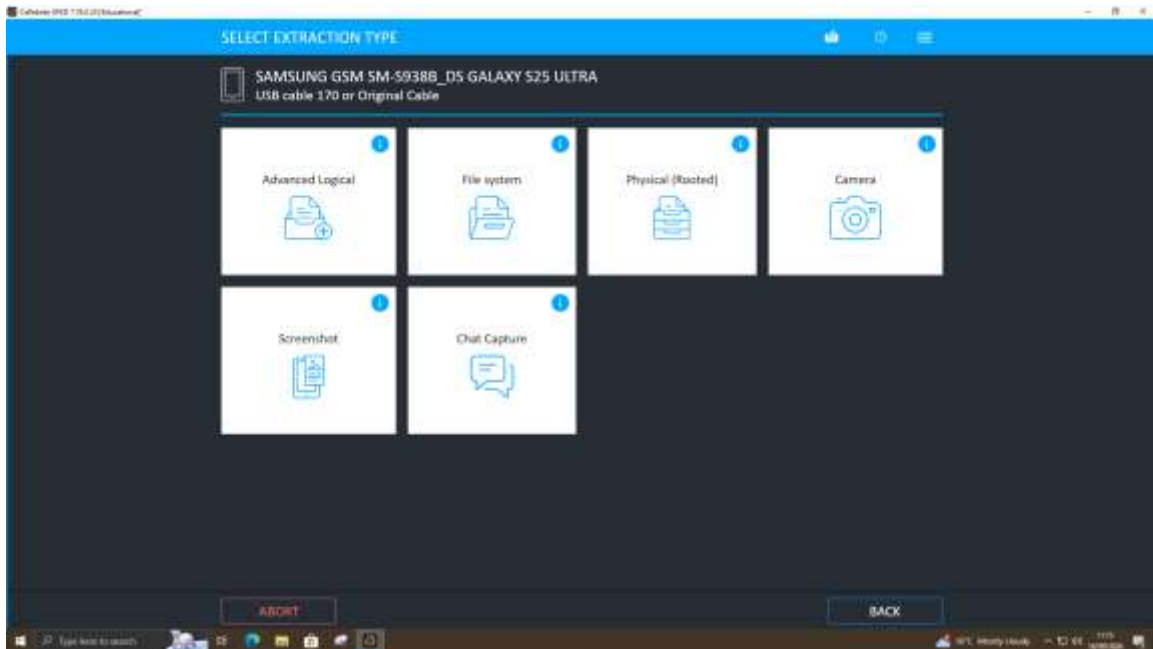


Figure 4.2: Extraction types offered by UFED 4PC 7.76 for the SM-S938B/DS. No Full File System option is available.

File System extraction in Android Backup mode was attempted twice, at 11:26 and again from 12:01. Both runs ended with the same error, “ADB backup failed – Shared memory was partially extracted or failed” (Figure 4.3). Neither produced a usable extraction. The APK Downgrade mode was not attempted, because it works by replacing installed applications with older versions, which alters the exhibit and falls outside the non-destructive scope in section 1.7.

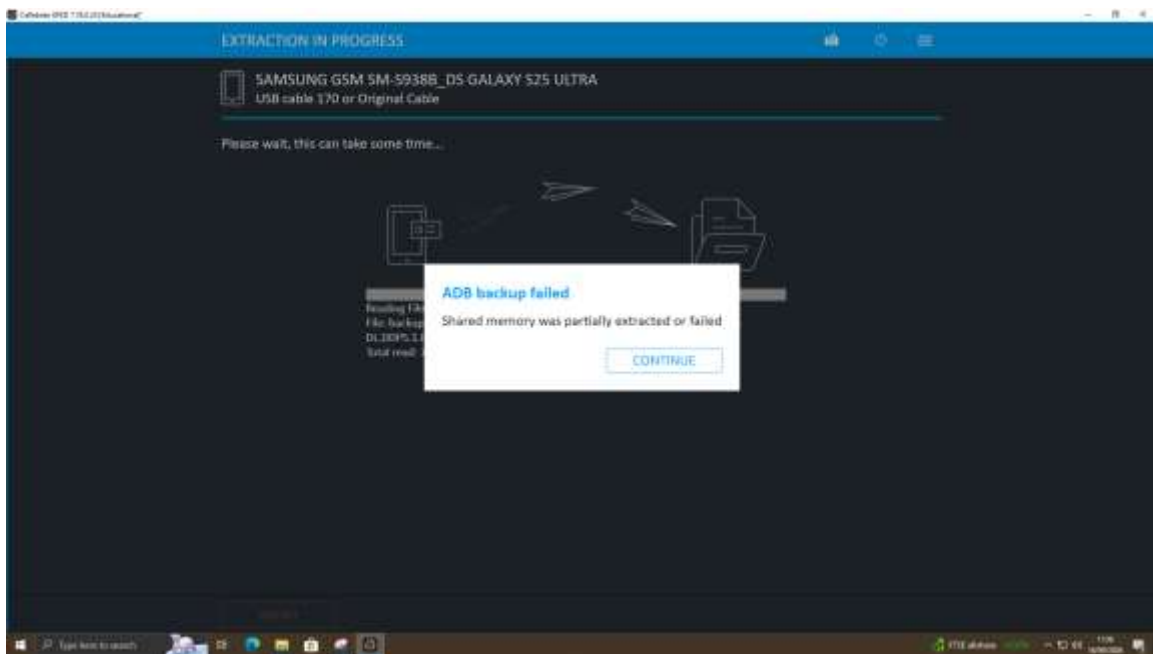


Figure 4.3: The File System (Android Backup) attempt ending in failure; the same error recurred on the second run.

Advanced Logical extraction was then run. A first attempt was started on 16 September at 12:45. The retained extraction is the second run, on 22 September from 11:38:42 to 12:25:24, which UFED reported as completed successfully (Figure 4.4). The ADB backup error reappeared during the Files stage of this run, and the IM, Email, Browsing Data and User Dictionary categories were reported as not supported. Physical Analyzer decoded 8,294 messages, of which 179 were native messages and 8,115 phone (SMS) messages, together with 1,998 call records and 897 contacts. It listed three applications, none of which it decoded. No WhatsApp, Signal or Telegram message was present.

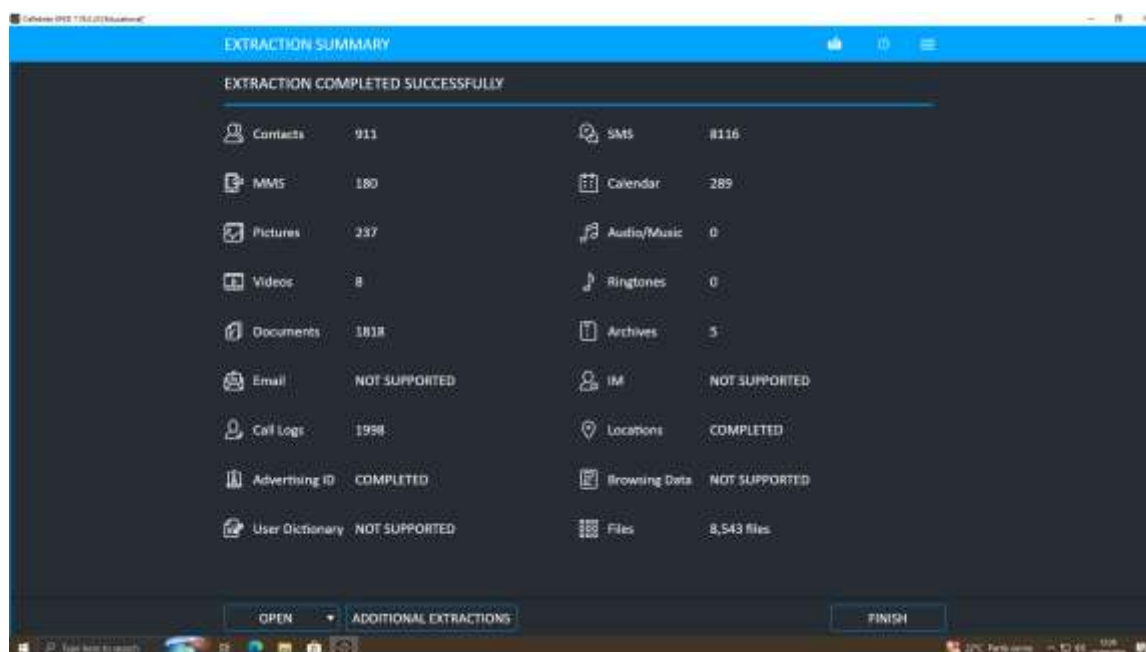


Figure 4.4: UFED Advanced Logical extraction summary, 22 September 2026. Instant messaging is reported as not supported.

Physical Analyzer also showed that the security patch level had changed to 5 August 2026 between 16 and 22 September, which means the handset installed a system update between the two sessions. This is recorded as deviation D4.

4.5 Open-source chain

Because neither chain can acquire the /data partition of this handset without root, the open-source chain was run against the only area an unprivileged ADB connection can read: shared storage mounted at /sdcard. The partition reported 185 GB in use. Shared storage was first copied with adb pull and then packed on the device into two archives, one for the Android directory (23.5 GB) and one for the remainder (5.8 GB). Both archives were hashed with SHA-256 on the workstation before any tool opened them, and the hash records were themselves hashed.

A file listing of the archives showed 14,649 entries under com.whatsapp, 98 under org.telegram.messenger and none for Signal (org.thoughtcrime.securesms). The WhatsApp entries included msgstore.db.crypt14, three daily backups and eight incremental backups dated 16 to 22 September (Figure 4.5). These are the application's encrypted local backups, not its working database. The Telegram entries were two empty media folders.

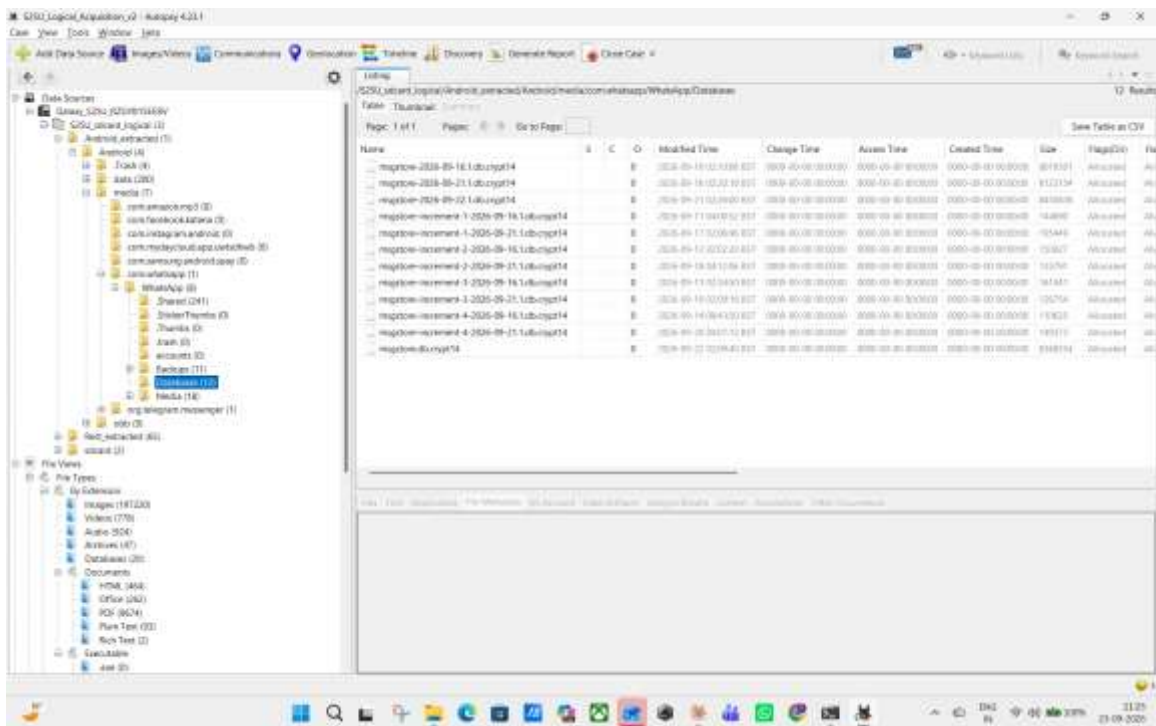


Figure 4.5: WhatsApp encrypted message-store backups located in shared storage, viewed in Autopsy.

ALEAPP could not read the Android archive directly. It stopped with a Python RecursionError while indexing the archive headers (Figure 4.6), because the archive contained redundant GNU long-name headers that the Python tarfile module could not process. Repacking the archive in pax format produced a file of only 5.5 GB with 8,527 entries, which was incomplete and was discarded. The archive was instead extracted with the Windows tar utility, yielding 21.8 GB. During extraction, 187 files whose names ended in a full stop and one directory whose name ended in a space could not be represented on NTFS and were renamed with a suffix. The renaming affected names on the working copy only, and it is recorded as deviation D6.

```
Command Prompt
File "tarfile.py", line 1426, in _proc_member
File "tarfile.py", line 1464, in _proc_guolong
File "tarfile.py", line 1488, in _frontarfile
File "tarfile.py", line 1464, in _proc_member
File "tarfile.py", line 1464, in _proc_guolong
File "tarfile.py", line 1488, in _frontarfile
File "tarfile.py", line 1426, in _proc_member
File "tarfile.py", line 1464, in _proc_guolong
File "tarfile.py", line 1488, in _frontarfile
File "tarfile.py", line 1381, in _frombuf
RecursionError: maximum recursion depth exceeded

C:\Users\smi patel>
C:\Users\smi patel>tar -c -f G:\Case_5250\01_Acquisition\sdcard_Android.pax.tar --format=pax @G:\Case_5250\01_Acquisition\sdcard_Android.tar
tar: Error reading archive G:\Case_5250\01_Acquisition\sdcard_Android.tar: Damaged archive: Redundant 'L' headers may cause filenames to be incorrect
tar: Error exit delayed from previous errors.

C:\Users\smi patel>dir G:\Case_5250\01_Acquisition
Volume in drive G is IT Shield
Volume Serial Number is 0A96-CD99

Directory of G:\Case_5250\01_Acquisition

22-09-2026 18:41 <DIR>
22-09-2026 18:41 <DIR>
22-09-2026 19:05 <DIR> sdcard
22-09-2026 20:22 21,436,540,266 sdcard_Android.tar
22-09-2026 20:29 1,845,751,808 sdcard_Rest.tar
23-09-2026 10:14 1,511,883,428 sdcard_Android.pax.tar
               3 File(s) 34,814,181,888 bytes
               3 Dir(s) 232,435,963,796 bytes free

C:\Users\smi patel>tar -tf G:\Case_5250\01_Acquisition\sdcard_Android.pax.tar | find /c /v ""
8527

C:\Users\smi patel>del G:\Case_5250\01_Acquisition\sdcard_Android.pax.tar

C:\Users\smi patel>mkdir G:\Case_5250\01_Acquisition\Android_extracted

C:\Users\smi patel>tar -x -f G:\Case_5250\01_Acquisition\sdcard_Android.tar -C G:\Case_5250\01_Acquisition\Android_extracted 2> G:\Case_5250\01_Notes\android_extract_errors.txt

C:\Users\smi patel>powershell -command "(Get-Childitem -LiteralPath G:\Case_5250\01_Acquisition\Android_extracted -Recurse -Force -ErrorAction SilentlyContinue | Measure-Object -Property Length -Sum).Sum / 1GB"
21.8328496179989
```

Figure 4.6: ALEAPP terminating on the Android archive, and the workaround by extraction.

ALEAPP was then run against the extracted folder. It indexed 30,689 files, ran 1,288 artefact modules in 12 minutes 52 seconds, and returned no message artefacts for any of the three applications. The WhatsApp parser was the slowest module (391 seconds) and found no readable database. The extracted folders were loaded into Autopsy as a logical file set. Ingest ran from 16:27 to 19:32 on 23 September and indexed 225,247 files, the larger count reflecting the second archive and the Embedded File Extractor module. Autopsy reported no deleted files, 480 encrypted files, 2,060 files with EXIF metadata and an empty Communications view (Figure 4.7). Only modification times were populated; change, access and creation times read as zero.

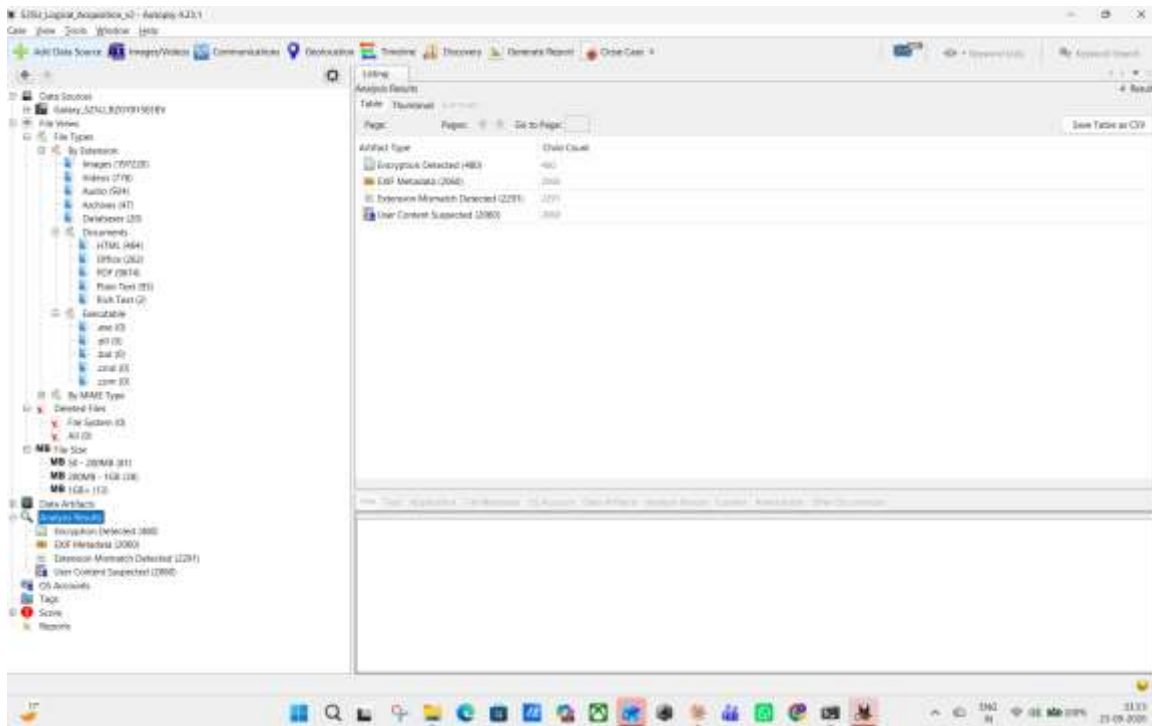


Figure 4.7: Autopsy results for the shared-storage data source. Deleted Files is zero by construction.

4.6 Causes of the acquisition failure

The failure has several causes, and they reinforce one another. They are set out here from the most direct cause to the most fundamental.

- Tool support. Full file system extraction of a locked, current Android handset depends on a vendor method specific to the chipset and patch level. The educational licence of UFED 4PC 7.76 offered no such method for the SM-S938B/DS, and Physical extraction was available only for rooted devices. Cellebrite states that its Inseyets platform has restored full file system extraction for a broad range of newer Android models, including Samsung [31]; that capability was not available through the educational UFED 4PC licence used here. A newer UFED release (7.78) was available during testing but not installed, and the Physical Analyzer version (7.73) was older than the UFED version.
- Patch level. The handset carried the July and then the August 2026 security patch. Vendor methods typically lag behind new patch levels, so the more current the device, the less likely a method exists. The mid-study update moved the device further from any supported configuration.
- Deprecation of ADB backup. Since Android 12, adb backup excludes the data of any application that targets API level 31 or higher unless the application is debuggable [29]. Release builds of WhatsApp, Signal and Telegram are not debuggable, so a backup-based file system extraction cannot return their data on Android 16 [33]. This matches the repeated shared-memory error and the absence of IM content.
- File-based encryption and a locked bootloader. Application-private data is credential-encrypted and readable only to a privileged process on the running device (section 2.4). Without root, nothing on the workstation side can read /data/data. Obtaining root requires unlocking the bootloader, which wipes user data under verified boot [23], so it would have destroyed the evidence the study set out to measure.

- Application design. WhatsApp encrypts its local backups with a key held in its private directory [32]; Signal keeps its database only in private storage; Telegram keeps its message cache there. Shared storage therefore holds, at best, encrypted copies and media.
- Samsung security controls. Auto Blocker can block commands received over USB and installation of applications from unauthorised sources [30], which would prevent a UFED client from being installed. It was switched off before testing (section 4.7), so it is excluded as the cause in this case, but it remains a likely obstacle in casework.

The combined effect is that every method that could reach deleted message records was either unavailable to the licence, blocked by the platform, or destructive. Tool choice did not determine the outcome. Access did, which is the position section 2.4 anticipated.

4.7 Steps taken to resolve the failure

The following steps were taken, each recorded in the acquisition log in Appendix D:

- The File System extraction was repeated with a second run to exclude a transient fault.
- A different USB cable and port were used.
- Samsung Auto Blocker was switched off, USB debugging and Stay Awake were enabled, the screen lock was set to None and the USB mode was set to file transfer, following the UFED device instructions.
- The Advanced Logical method was run instead, as the most complete method available under the licence, and repeated on a second day.
- An open-source acquisition of shared storage was carried out with ADB so that both chains could still be compared on identical material.
- When ALEAPP failed on the archive, the archive was repacked, and when that proved incomplete it was extracted and processed as a folder.

Root access, bootloader unlocking and the APK Downgrade mode were deliberately not used. Each alters or destroys the exhibit, and each is excluded by section 1.7 and by the ethical approval.

4.8 Deviation log

Table 4.1: Deviations from the Chapter 3 protocol

ID	Deviation	Reason	Effect on results
D1	One test device; no second test device, rooted device or control device	Only the S25 Ultra was available	No cross-version comparison, no upper bound, precision not computable (RQ4, part of RQ3)
D2	Corpus limited to 390 messages in three apps	Scope of the shared dataset	SMS, media and app-data classes not scored
D3	Full file system and physical acquisition not achieved	Section 4.6	Deleted records unreachable; RQ1 and RQ2 answered only in part
D4	Security patch changed from 5 July to 5 August 2026 between sessions	Automatic system update	Sessions not on identical builds
D5	Single retained run per method, not triplicate	Two failed runs; one successful logical run	Replicate hash agreement not measured

D6	187 file names and one directory renamed on the working copy	NTFS cannot store trailing dot or space	Names differ from source; content hashes unaffected
D7	Temporary archive error files written to /data/local/tmp and later removed	On-device archiving	Minor change to the exhibit, logged under ACPO principle 2
D8	Seven-day ageing condition not run	Acquisition failed before ageing was relevant	Time-dependent loss not measured
D9	Acquisition shared with another study and not performed solely by the author	Supervisor-approved shared dataset	The single-examiner control in section 3.2 applies to analysis only

Chapter 5: Results

5.1 Acquisition outcome matrix

Table 5.1 sets out the outcome of every acquisition method against the handset. It is the primary result of the study, because it fixes what the analysis stage could possibly find.

Table 5.1: Acquisition outcome by method

Method	Chain	Outcome	Reaches app-private data?
Advanced Logical	UFED 4PC 7.76	Completed; IM, email, browsing not supported	No
File System (Android Backup)	UFED 4PC 7.76	Failed twice (ADB backup error)	No
File System (APK Downgrade)	UFED 4PC 7.76	Not attempted (alters exhibit)	–
Full File System	UFED 4PC 7.76	Not offered for this model	–
Physical	UFED 4PC 7.76	Requires root; not attempted	–
Shared-storage copy (ADB)	Open source	Completed; 29.3 GB in two archives	No

Only two methods produced output, and neither reaches /data/data. RQ2 asked how acquisition method affects the volume and fidelity of recovered deleted data. On this device the answer is categorical rather than graded: the methods that would recover deleted message records were unavailable, and the methods that were available cannot recover them.

5.2 Commercial chain output

The Advanced Logical extraction took 46 minutes 42 seconds. Table 5.2 summarises what it returned, as reported by UFED and as decoded by Physical Analyzer.

Table 5.2: UFED Advanced Logical output

Category	UFED summary	Physical Analyzer
SMS / MMS	8,116 / 180	8,294 messages (8,115 phone, 179 native)
Call logs	1,998	1,998
Contacts	911	897
Calendar	289	152
Pictures / videos	237 / 8	Media: 8,018 items
Documents	1,818	2,930
Files	8,543 (partial; ADB backup error)	–
Instant messaging	Not supported	0 WhatsApp, Signal or Telegram messages
Applications	–	3 listed, 0 decoded

The differences between the two columns are informative. UFED counts what it collected; Physical Analyzer counts what it parsed and deduplicated, and it also classifies files found inside the collection. Neither column contains any content from the three applications under study.

5.3 Open-source chain output

The open-source chain processed a larger volume of material but found no more message content. ALEAPP indexed 30,689 files and completed 1,288 modules in 12 minutes 52 seconds with no message artefacts. Autopsy indexed 225,247 files in 3 hours 5 minutes and produced 6,891 analysis results: 2,060 files with EXIF metadata, 2,291 extension mismatches, 480 encrypted files and 2,060 files flagged as suspected user content. The Communications view and the Data Artifacts message categories were empty, and Deleted Files was zero.

The zero in Deleted Files is an expected result, not a tool defect. A logical copy contains only allocated files, so there is no unallocated space or file system metadata from which Autopsy could recover a deleted file. The same holds for timestamps: the archives kept modification time but not change, access or creation time, so three of the four MAC times were unavailable.

5.4 Messaging application artefacts

Table 5.3: Artefacts found for each messaging application

Application	In shared storage	UFED Advanced Logical	ALEAPP / Autopsy
WhatsApp	msgstore.db.crypt14, 3 daily and 8 incremental backups; media folders	Not supported	Files listed; database encrypted, not parsed
Signal	Nothing	Not supported	Nothing
Telegram	Two empty media folders	Not supported	Nothing

WhatsApp is the only application that left message-bearing files within reach, and it did so because it writes daily encrypted backups to shared storage. The backups were acquired and hashed intact, but without the key held in the application's private directory they cannot be decrypted [32]. Signal and Telegram left nothing, which follows from their design: neither writes its message store to shared storage.

5.5 Scoring against the ground truth

Table 5.4 scores the corpus against the categories defined in section 3.8. The WhatsApp items are scored as present but undecoded rather than not acquired, because the backups that contain them were acquired intact; the ones dated before deletion would contain the 14 deleted messages in their pre-deletion state. The Signal and Telegram items were not acquired in any form.

Table 5.4: Recovery scoring against the ground-truth manifest

App	Items	Deleted	Fully recovered	Partially recovered	Present, undecoded	Absent	Not acquired
WhatsApp	130	14	0	0	130	0	0
Signal	130	14	0	0	0	0	130
Telegram	130	14	0	0	0	0	130
Total	390	42	0	0	130	0	260

Recovery rate is 0 of 42 for both chains, and precision cannot be computed, because neither chain reported any item for the three applications and no control device was available (D1). The zero recovery rate must be read with care. It does not show that the deleted records were destroyed, and it does not show that either tool failed to parse them. It shows that the records were never presented to either tool. In the terms of section 1.2, a missing artefact here has a third cause beyond platform destruction and software failure: the platform withheld access.

5.6 Metadata fidelity

Section 3.9 defined metadata fidelity as the accuracy of timestamp, path and attribution for each recovered item. Because no ground-truth item was recovered, it could not be scored against the manifest, but the acquired material shows where fidelity would have been lost. In the open-source chain, the on-device archive kept each file's modification time and discarded change, access and creation times, which Autopsy displayed as zero. The WhatsApp backups illustrate the consequence: their modification times (for example 02:09:40 on 22 September for msgstore.db.crypt14) record when the application wrote the backup, not when any message in it was sent or deleted. Paths were preserved except for the 187 renamed files in D6, where the working-copy path no longer matches the device path exactly. Attribution to an account was not possible for any shared-storage file, since shared storage records no owning user beyond the application folder name.

In the commercial chain, the SMS and call records carried the timestamps stored by the content providers, and Physical Analyzer flagged that the extracted time-zone stamp should be checked and that the last IMEI digit might be incorrect. Both are small warnings, but they show that even a successful logical extraction carries metadata caveats that belong in the report.

5.7 Integrity and efficiency

Acquisition hashes were recorded for both archives before analysis (Appendix D). The on-device archiving step wrote two temporary error files to /data/local/tmp, which were logged and removed (D7); this is the only recorded change to the exhibit. The renaming in D6 changed names on the working copy but not file content. Table 5.5 compares the time each chain took.

Table 5.5: Processing time by stage

Stage	Commercial chain	Open-source chain
Acquisition	46 min 42 s (Advanced Logical)	About 1 h 35 min (copy and two archives)
Parsing	Physical Analyzer decoding (automatic)	ALEAPP 12 min 52 s
Indexing and analysis	Included in decoding	Autopsy ingest 3 h 5 min
Manual interventions	0	4 (archive repack, extraction, renaming, re-run)

The commercial chain was faster and required no manual work. The open-source chain required four manual interventions to reach a result, which is the examiner-effort difference section 3.9 was designed to capture.

Chapter 6: Discussion

6.1 Answering the research questions

The study was designed to measure recovery; what it measured instead was access. Table 6.1 states how far each research question can be answered from the evidence obtained.

Table 6.1: Status of the research questions

RQ	Status	Basis
RQ1 Which classes of deleted data remain recoverable?	Answered for access only	0 of 42 deleted messages presented to either tool; WhatsApp copies held encrypted
RQ2 Effect of acquisition method	Answered	Only logical methods succeeded; neither reaches app-private data (Table 5.1)
RQ3 Commercial vs open-source chain	Partly answered	Compared at logical level on output, time and effort; precision not computable
RQ4 Effect of version, encryption and root	Not answered	Single unrooted device (D1)
RQ5 Implications for evidential weight	Answered	Section 6.5

6.2 Why access, not tool capability, set the result

Section 2.4 argued that device state at seizure, rather than tool capability, sets the ceiling for an examination. The results support a stronger version of that claim. The handset was in the most favourable state an examiner can hope for: unlocked, After First Unlock, with debugging enabled and the owner co-operating. Even so, the ceiling was the shared-storage partition, because every route to credential-encrypted application data required a vendor method that the licence did not include or a privilege that could be gained only by wiping the device.

The failure was also layered. Removing any one barrier would not have been enough. A newer UFED release might have added support for the chipset, but the patch update during the study shows that such support is a moving target. ADB backup would still have excluded the applications' data [29], and bootloader unlocking would still have erased it [23]. This matters for interpretation, because it rules out the explanation that the study simply used the wrong setting. Blanckesteijn, Fukami and Geradts found that what survives on current handsets depends on the device and its implementation [3]; this study found that, on a current flagship, what can be reached depends on the same things before persistence even becomes the question.

6.3 The two chains compared at the logical level

At the level both chains could reach, they did different jobs. UFED Advanced Logical worked through the content providers and returned structured SMS, call, contact and calendar records in 47 minutes with no manual effort. The open-source chain could not query those providers in the same way, but it copied the whole of shared storage and so captured the WhatsApp backups that UFED's logical method did not collect. The two outputs overlap only partly, which supports the dual-tool position in section 2.8: the second chain added evidence the first missed, although here the added evidence was encrypted.

The open-source chain was also more transparent about its failures. ALEAPP's crash and the NTFS renaming were visible, reproducible and explainable from the tool output and the source code, which is the transparency advantage Ismail and Zainol Ariffin identify [13]. UFED's shared-memory error, by contrast, gives no reason, and the count differences between UFED and Physical Analyzer in Table 5.2 cannot be traced to a parsing decision. Gruber, Hargreaves and Freiling warn that error can enter at the interpretation stage where it is hard to detect [10]; a closed chain that reports "completed successfully" while a component has failed is an example of that risk.

The cost of that transparency was effort. The open-source chain needed four manual interventions and about four hours of processing against a largely automatic commercial run. For a practitioner, this reverses the usual framing: on a device of this kind, the open-source chain is worth its extra time not for parsing depth but for the breadth of files it collects.

6.4 Comparison with prior studies

The result sits uneasily with parts of the literature, and the differences are instructive. Sudiana et al. recovered WhatsApp disappearing-message evidence from an unrooted Android handset under a NIST SP 800-101r1 process [14]. Their devices also ran Android 12, after the change to ADB backup [29], but they obtained WhatsApp data using WhatsApp Key/DB Extractor, which works by installing an older version of the application. This study rejected that APK-downgrade route because it alters the exhibit (section 1.7). The difference in outcome therefore reflects what an examiner is willing to do to the device, not only the Android version, and published unrooted results should be read together with the method used to obtain them.

Heath, MacDermott and Akinbi found that ephemeral and deleted messages frequently survive in application storage [9]. Nothing in this study contradicts that. The WhatsApp backups dated before the deletion very probably contain the 14 deleted WhatsApp messages in their original form; the barrier is the key, not the absence of data. Their finding and this one are compatible once persistence and access are treated as separate questions, which is the distinction section 5.5 draws.

Geus, Ottmann and Freiling evaluated backup-based acquisition and found it could return a correct copy of original storage with identifiable corner cases [12]. On Android 16 the backup route did not return the application data at all, which is a larger failure than any corner case, and it shows how quickly a validated acquisition path can close. Yoo, Park and Jeong reported hash inconsistencies across repeated full file system extractions [11]; this study could not test that, but it did record a different kind of instability between sessions, the system update in D4, which changed the build under examination without any action by the examiner.

Finally, the comparative tool studies [6], [8], [15] each worked from an image that had already been acquired. By starting from the device rather than the image, this study shows a stage those comparisons leave out. On current hardware, the comparison between tools can be settled at acquisition, before any parsing difference comes into play.

6.5 Evidential implications

Three points follow for evidential weight. First, a zero recovery figure from a logical acquisition says nothing about whether deleted data exists on the device. A report that presents it without stating the access ceiling would mislead a court, and the outcome matrix in Table 5.1 is the kind of disclosure that prevents that.

Second, the study recorded three changes that an examiner must be able to explain under ACPO principle 2 [16]: the temporary files written to the device during archiving, the automatic system update between sessions, and the renaming of files on the working copy. None affects file content, but the update in particular shows that a device kept powered and connected between sessions will not stay in the state in which it was seized. Flight mode and update deferral should be part of the preservation step. The shared use of one acquisition by two studies (D9) also shows why ACPO principle 3 requires an audit trail that records who performed each action; without it, a third party could not tell which examiner produced which result.

Third, the Forensic Science Regulator requires a method to be validated for the class of exhibit on which it is used [19]. For current Samsung flagships on Android 16, this study shows that neither the educational UFED 4PC licence nor an ADB-based workflow is a valid method for recovering deleted messages. That is a validation result in its own right, and it is the kind of negative finding that the tool-comparison literature rarely reports [6], [8], [15].

6.6 Limitations

The limitations in section 3.12 apply, and the deviations in Table 4.1 add to them. The study rests on one device, one licence tier and one pair of patch levels, so it shows what happened under these conditions and cannot say how common the outcome is. The corpus covered three messaging applications only, so SMS, media and application-data classes were not scored against ground truth. No control device was used, so precision was not measured. The acquisition was shared with another study, so this study relies on the acquisition log for the steps it did not perform itself. Finally, the conclusion about tool support is tied to the versions tested; a later UFED release or a higher licence tier may reach the same handset [31], and that possibility should be read as a direction for further work rather than a weakness in the result.

Chapter 7: Conclusion

7.1 Objectives revisited

O1 was met in Chapter 2, which reviewed Android security, deletion semantics and standards and identified the absence of ground-truth comparisons of commercial and open-source chains. O2 was met in Chapter 3, which set out a manifest-based design with metrics fixed in advance. O3 was met in part: logical and shared-storage acquisitions succeeded, while file system, full file system and physical acquisition did not, and the reasons were established. O4 was met at the logical level only, since no deleted record reached either chain. O5 could not be met with a single device. O6 is addressed in section 7.3.

7.2 Contributions

The study makes three contributions. The first is empirical: on a Samsung Galaxy S25 Ultra running Android 16 with a current patch level, in the most favourable After First Unlock state, neither an educational-licence UFED 4PC nor an ADB-based open-source workflow could reach the application-private storage in which deleted messages persist. All 42 deleted ground-truth messages remained out of reach of both chains, and only WhatsApp left an encrypted copy in accessible storage.

The second is a structured explanation of the failure. It combines tool support, patch level, the deprecation of ADB backup, file-based encryption with a locked bootloader, application design and Samsung's own controls, and it shows that removing any single barrier would not have changed the outcome.

The third is methodological. Scoring against a manifest separated items that were acquired but encrypted from items that were never acquired, which a tool-against-tool comparison could not have done. It also showed that a recovery rate needs an accompanying access statement before it can be interpreted.

7.3 Recommendations for practitioners

- Establish the access ceiling before choosing tools. Record which acquisition methods the tool and licence offer for the exact model and patch level, and report failed methods alongside successful ones.
- Preserve device state immediately: isolate the handset from networks and defer system updates, since an update between sessions changes the exhibit.
- Use both chains on current Android handsets. The commercial logical method gives structured provider records quickly; a full copy of shared storage captures application backups and media that the logical method omits.
- Do not report a zero from a logical acquisition as absence of deleted data. State the method, the partitions reached and the reason deeper methods were unavailable.
- Examine working copies on a file system that preserves Android file names, or log every renaming, to avoid silent changes to the evidence.
- Record who performed each step whenever an acquisition is shared or reused, so that the audit trail remains attributable.

7.4 Future work

The design in Chapter 3 remains untested, and the obvious next step is to run it under conditions where full file system acquisition is possible. That could mean a higher Cellebrite licence tier such as Inseyets, which the vendor states supports current Samsung models [31], or an older or research handset that can be

rooted before it is populated, so that no evidence is lost when the bootloader is unlocked. With that access, the corpus should be extended to SMS, media and application data, a control device added to make precision measurable, and the seven-day ageing condition run as planned. A second line of work is the WhatsApp result. Measuring how many deleted messages survive in the encrypted daily backups, where a lawful route to the key exists [32], would show whether the backups are a practical source of deleted-message evidence. Finally, iOS remains outside the study and would benefit from the same manifest-based design.

7.5 Closing remark

The study set out to measure how much deleted data two tool chains could recover from a current Android handset. On the device tested, the answer was determined before either tool parsed anything: the data was out of reach. That is not the result the design anticipated, but it is a measured result, and it is one that examiners working with current flagship devices need to be able to state plainly.

References

- [1] P. S. Vinayagam, "Mobile forensics: Investigation and tools," *International Journal of Computer Trends and Technology*, vol. 73, no. 6, pp. 7–15, 2025, doi: 10.14445/22312803/IJCTT-V73I6P102.
- [2] S. Ntshangase, N. Nelufule, D. Muliase, M. Mtshali, C. Mokoena, and P. Moloi, "A survey of digital forensic tools for Android and iOS smart phones," in *Proc. IEEE Int. Conf. Cyber Security and Resilience (CSR)*, 2024, doi: 10.1109/CSR61664.2024.10679486.
- [3] M. B. Blanckesteijn, A. Fukami, and Z. J. M. H. Geradts, "Assessing data remnants in modern smartphones after factory reset," *Forensic Science International: Digital Investigation*, vol. 46, art. 301587, 2023.
- [4] S. Lee, S. Park, I. Lee, and J. Choi, "A comprehensive analysis and evaluation of SQLite deleted record recovery techniques: A survey," *Forensic Science International: Digital Investigation*, vol. 55, art. 302031, 2025, doi: 10.1016/j.fsidi.2025.302031.
- [5] J. Oh and H. Hwang, "Advanced forensic recovery of deleted file data in F2FS," *Forensic Science International: Digital Investigation*, 2025.
- [6] J. Mehta, Y. Bhadania, P. Shah, and P. Prajapati, "Comparative study of mobile forensics tools: Autopsy, Belkasoft X and Magnet Axiom," in *Proc. Int. Conf. Electronics and Sustainable Communication Systems (ICESC)*, 2024, doi: 10.1109/ICESC60852.2024.10689971.
- [7] I. Ismail and K. A. Zainol Ariffin, "The admissibility of digital evidence from open-source forensic tools: Development of a framework for legal acceptance," *PLOS ONE*, 2025, doi: 10.1371/journal.pone.0331683.
- [8] M. Farnan, J. Pratt, A. David, and M. Shakiba, "Digital forensic tools: Comparison of Autopsy TSK and Forensic Explorer," in *Proc. Int. Conf. Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, 2024, doi: 10.1109/ICITEICS61368.2024.10625076.
- [9] H. Heath, A. MacDermott, and A. Akinbi, "Forensic analysis of ephemeral messaging applications: Disappearing messages or evidential data?," *Forensic Science International: Digital Investigation*, 2023. [Online]. Available: <https://researchonline.ljmu.ac.uk/id/eprint/20016/>

- [10] J. Gruber, C. J. Hargreaves, and F. Freiling, "Contamination of digital evidence: Understanding an underexposed risk," *Forensic Science International: Digital Investigation*, vol. 44, art. 301501, 2023, doi: 10.1016/j.fsidi.2023.301501.
- [11] S. Yoo, Y. Park, and D. Jeong, "Preserving evidence integrity in full file system extractions: Assessing hash inconsistencies and potential solutions," *Journal of Forensic Sciences*, 2026, doi: 10.1111/1556-4029.70362.
- [12] J. Geus, J. Ottmann, and F. Freiling, "Systematic evaluation of forensic data acquisition using smartphone local backup," presented at DFRWS USA 2023; arXiv:2404.12808, 2024. [Online]. Available: <https://arxiv.org/abs/2404.12808>.
- [13] I. Ismail and K. A. Zainol Ariffin, "Open source tools for digital forensic investigation: Capability, reliability, transparency and legal requirements," *KSII Transactions on Internet and Information Systems*, vol. 18, no. 9, pp. 2692–2716, 2024, doi: 10.3837/tiis.2024.09.012.
- [14] D. Sudiana, C. H. Nuruddin, M. Rizkinia, and D. Husna, "Forensic analysis of WhatsApp disappearing message on unrooted Android using mobile device forensics methodology NIST SP 800-101r1," *Evergreen*, 2024, doi: 10.5109/7172316.
- [15] T. Wangchuk, Y. Tshering, N. Mandela, and P. Rughani, "Forensic analysis of Scientific Linux image using commercial and open-source forensic tools," *Journal of Applied Engineering, Technology and Management*, vol. 4, no. 1, 2024, doi: 10.54417/jaetm.v4i1.133.
- [16] Association of Chief Police Officers, *ACPO Good Practice Guide for Digital Evidence*, version 5. London, UK: ACPO, 2012.
- [17] *Information Technology — Security Techniques — Guidelines for Identification, Collection, Acquisition and Preservation of Digital Evidence*, ISO/IEC 27037:2012, International Organization for Standardization, Geneva, Switzerland, 2012.
- [18] *General Requirements for the Competence of Testing and Calibration Laboratories*, ISO/IEC 17025:2017, International Organization for Standardization, Geneva, Switzerland, 2017.
- [19] Forensic Science Regulator, *Forensic Science Regulator Code of Practice*, Version 2. London, UK: Home Office, 2025.
- [20] R. Ayers, S. Brothers, and W. Jansen, *Guidelines on Mobile Device Forensics*, NIST Special Publication 800-101 Revision 1. Gaithersburg, MD, USA: National Institute of Standards and Technology, 2014.
- [21] National Institute of Standards and Technology, "Computer Forensics Tool Testing (CFTT) Program." [Online]. Available: <https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt>.
- [22] Android Open Source Project, "File-based encryption," Android Platform Documentation. [Online]. Available: <https://source.android.com/docs/security/features/encryption/file-based>
- [23] Android Open Source Project, "Verified boot," Android Platform Documentation. [Online]. Available: <https://source.android.com/docs/security/features/verifiedboot>
- [24] Android Developers, "Storage updates in Android 11," Android Developer Documentation. [Online]. Available: <https://developer.android.com/about/versions/11/privacy/storage>

[25] SQLite, "Write-ahead logging," SQLite Documentation. [Online]. Available: <https://www.sqlite.org/wal.html>.

[26] *Data Protection Act 2018*, c. 12, United Kingdom.

[27] *Computer Misuse Act 1990*, c. 18, United Kingdom.

[28] *Police and Criminal Evidence Act 1984*, c. 60, United Kingdom.

[29] Android Developers, "Behavior changes: Apps targeting Android 12," Android Developer Documentation. [Online]. Available: <https://developer.android.com/about/versions/12/behavior-changes-12>

[30] Samsung Electronics, "Protect your Samsung Galaxy device with the Auto Blocker feature," Samsung Support. [Online]. Available: <https://www.samsung.com/us/support/answer/ANS10003636/>

[31] P. Lorentz, "The access gap is closed: What Cellebrite can unlock in 2026," Cellebrite Blog, 29 May 2026. [Online]. Available: <https://cellebrite.com/en/blog/the-access-gap-is-closed-what-cellebrite-can-unlock-in-2026>.

[32] W. A. Prabowo, F. Mohsen, and S. R. Selamat, "WhatsApp mobile applications in the lens of digital forensics: Deciphering the msgstore.db.crypt14 file," *Journal of Cyber Security and Mobility*, vol. 14, no. 4, pp. 823–848, 2025, doi: 10.13052/jcsm2245-1439.1443.

[33] I. Morosan, "Mobile pentesting 101 – The death of ADB backup: Modern data extraction in 2026," Security Cafe, Feb. 2026. [Online]. Available: <https://securitycafe.ro/2026/02/02/mobile-pentesting-101-the-death-of-adb-backup-modern-data-extraction-in-2026/>