Mobile Device Triage Toolkit: Deterministic, Read-only Forensic Pre-Assessment

Author: Luis F. Jusino Alamo
Advisor: Nelliud Torres Batista
Master in Computer Science
Graduate Project EXPO, February 2026

Abstract

Mobile investigations face evidence backlogs and limited time to decide whether a device merits full imaging. We present the Mobile Device Triage Toolkit (MDTK), a read-only, deterministic workflow that inspects disk images (RAW and E01 via pytsk3/pyewf or E01 export) or backups/logical folders and summarizes high-value artifacts. MDTK provides a filesystem summary, app inventory , SQLite table counts, endpoint-pattern hits, and a mini-timeline, exporting both JSON and a uniform PDF. For forensic defensibility, MDTK records an append-only JSONL audit log, refuses writable mounts, pins the runtime environment, and fixes timestamps to UTC seconds; Ed25519 signing is supported. We evaluate MDTK on a manifest of sample images and report runtime, artifact coverage, and reproducibility by comparing JSON/PDF hashes across repeated runs. Results show byte-identical outputs, median execution under 8.33s and fast visibility into artifacts such as messaging and browser histories. MDTK targets triage escalation, not full analysis, and runs on Windows via WSL.

Introduction

- Digital forensics teams increasingly face "too many devices and too little time."
- The first question is often which devices deserve full imaging now, not "what happened."
- Many triage approaches trade speed for rigor, risking writes, nondeterminism, or ad-hoc procedures that weaken chain of custody.
- MDTK provides a fast, defensible snapshot using a deterministic workflow for RAW/E01 evidence and structured reporting (JSON/PDF + audit logs).

Background

Triage vs Full parsing

- Triage is a defensible "first look" to answer scope/prioritization quickly, while full parsing aim for deep decoding/correlation with higher cost/complexity.
- Mobile adds constraints (app sandboxes, multiple profiles, encryption, vendor partition layouts, rapid OS changes), making careful triage valuable.

Why determinism matters

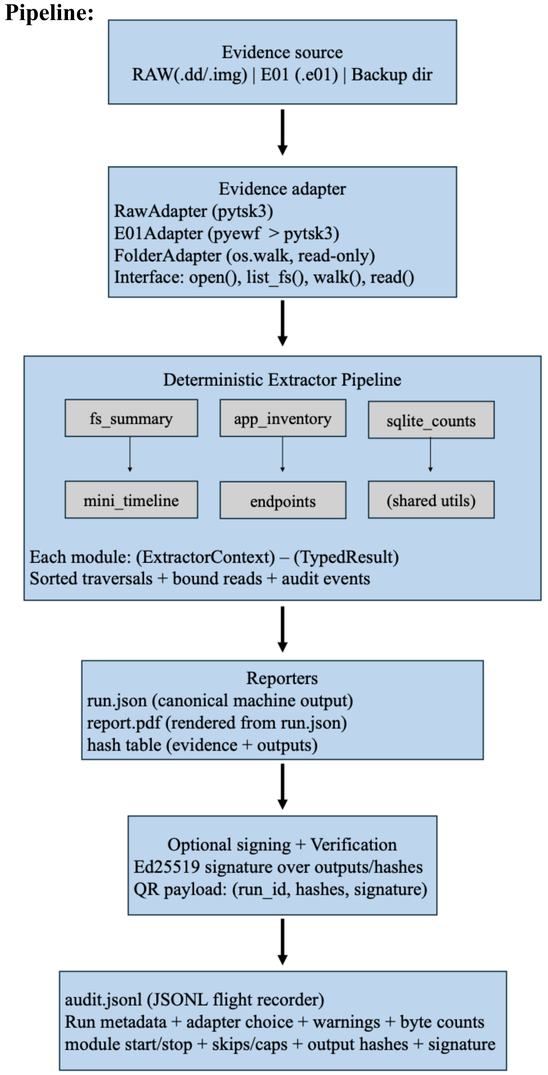
- Forensic outputs should be reproducible; nondeterminism comes from unsorted walks, locale/time formatting, concurrency effects, and dependency drift.
- MDTK treats determinism as a first-class goal to support verification and repeatable workflows.

Problem

- Investigators need triage tooling that is fast but also defensible: no accidental writes, stable outputs, and transparent documentation of what was examined or skipped.
- Existing building blocks (TSK/pytsk3, EWF/libewf/pyewf) enable access, but end-to-end pipelines often lack strict zero-write guardrails and structured auditability.
- MDTK targets this gap with fail-closed safety plus deterministic reporting.

Methodology

Pipeline:



- **RAW:** pytsk3 parsers partitions/filesystems with read-only traversal.
- **E01:** pyewf streams EWF segments into a read-only byte source for pytsk3 (or exported RAW-compatible stream).
- **Logical folders**: directory adapter provides the same walk/read semantics as image-backed adapters.
- **fs_summary:** partitions/filesystems + basic tree stats
- **app_inventory:** deterministic presence checks for apps
- **sqlite_counts:** headers/tables/row using read-only queries; logs skips/errors.
- **mini_timeline:** allow-listed timestamps normalized to UTC seconds; stable ordering.
- **endpoints:** rule-based presence hits for high-value artifact families (messaging/browser/logs)

Results and Discussion

Image ID	Format	Size	SHA-256
HTC_Desire_S (A)	RAW	2.2 GB	6d6548…
HTC_One_XL (B)	RAW	15.27 GB	b14e26…
N115015_CHIP_OFF (C)	001	7.6 GB	6e7952…
Jo-favorites-usb-2009-12-11 (D)	E01	0.211 GB	1798e0…

Image	GB	Time	Files	Apps	DBs	TL	JSON	Repro
A	2.2	5.93s	6	0	0	4439	884KB	Yes
B	15.27	14.93s	35	0	0	3	9KB	Yes
C	7.6	10.73s	44	0	0	165	40KB	Yes
D	0.21	2.83s	1	0	0	739	160KB	Yes

Performance + outputs

- Across datasets A-D, MDTK completed triage in 2.83-14.93s, generally increasing with workload size/files walked
- Mini-timeline rows ranged from 3 to 4.439; emitted JSON sizes ranged from 9KB to 884KB.
- Reproducibility was Yes(3/3) when SHA-256 hashes matched for both run.json and the PDF across repeated runs.

Safety Tests

- RW mount refusal: MDTK refused before extraction (Pass)
- Ambiguous mount state: MDTK abort fail-closed when RO could not be verified (Pass)
- RO mount control: MDTK proceeded normally and emitted JSON/PDF (Pass).

Interpretation

- MDTK supports escalation decisions using interpretable indicators (presence/counts/timeline density)
- It favors breadth-first triage over deep decoding, reducing variability and failure risk.

Conclusions

- A minimal, **read-only pipeline can produce actionable triage outputs** while preserving defensibility through deterministic execution and comprehensive audit logging.
- Across evaluated datasets, **MDTK produced stable outputs and failed closed under unsafe mount conditions**, supporting rapid decisions without sacrificing transparency.

Future Work

- Broader corpora and ground-truth labeling for strong completeness claims.
- Stronger cross-platform safety verification (especially Windows-native)
- Expanded, versioned profiles/endpoints while preserving determinism and zero-write guarantees.

Acknowledgements

I would like to acknowledge my advisor Nelliud Torres Batista, PhD, my reviewer Joann Casillas, and the Polytechnic University of Puerto Rico Master in Computer Science Program.

References

1. J. Metz. (2025, Dec.). *libewf: Libewf is a library to access the Expert Witness Compression Format (EWF)* [Online]. Available: https://github.com/libyal/libewf.
2. J. Metz. (2024, May, 5). *libewf-python 20240506* [Online]. Available: https://pypi.org/project/libewf-python/.
3. A. Robinson, R. Becker, and the ReportLab team. (2026, Jan.). *ReportLab PDF Generation User Guide* [PDF]. Available: https://www.reportlab.com/docs/reportlab-userguide.pdf.
4. SQLite. *Database File Format* [Online]. Available: https://www.sqlite.org/fileformat.html.
5. M. Cohen. *pytsk3* [Online]. Available: https://pypi.org/project/pytsk3/.
6. py4n6. pytsk: *Python bindings for The Sleuth Kit (libtsk)* [Online]. Available: https://github.com/py4n6/pytsk.
7. SQLite. *SQLite Documentation* [Online]. Available: https://www.sqlite.org/docs.html.
8. B. Carrier. *mactime(1) Arch Linux Manual Pages* [Online]. Available: https://man.archlinux.org/man/extra/sleuthkit/mactime.1.en.
9. B. Carrier. (n.d.). *mactime(1) The Sleuth Kit manual page* [Online]. Available: https://www.sleuthkit.org/sleuthkit/man/mactime.html.
10. A. Robinson, R. Becker. *ReportLab Documentation* [Online]. Available: https://docs.reportlab.com/.
11. Pallets. *Welcome to Click* (Click Documentation 8.3.x) [Online]. Available: https://click.palletsprojects.com/en/stable/