

Forensic Analysis of Whatsapp Artifacts on Android Devices

Deenathayalan P¹, Magna Y²

¹Department of Computer Science, Dr. MGR Educational and Research Institute, Chennai, Tamil Nadu, India.

²Assistant Professor, Center of Excellence in Digital Forensics, Chennai, Tamil Nadu, India.

Abstract - Android devices are now a vital part of everyday life. They store a lot of personal, professional, and location data. This makes them an important source of digital evidence in cybercrime and criminal investigations. WhatsApp is one of the most popular messaging platforms worldwide. It saves a large amount of sensitive communication data, including messages, media files, voice notes, and documents in encrypted formats on Android devices. This project outlines the design and implementation of a Digital Forensic Analysis System. It focuses on extracting, decrypting, and analyzing WhatsApp data from non-rooted Android devices. Unlike traditional forensic tools that need root access to protected system areas, this system uses accessible data paths like encrypted backup files (.crypt12, .crypt14, .crypt15) and WhatsApp media folders. This allows for a methodical extraction process that maintains the integrity and acceptability of digital evidence in legal matters. The system is built as a web-based Flask application. It offers a full forensic investigation workflow that includes case management, device data capture via Android Debug Bridge (ADB), database decryption using extracted key files, chat message parsing, bulk media artifact analysis with EXIF metadata extraction, GPS coordinate recovery, chain-of-custody logging, and automatic generation of PDF and HTML forensic reports. Key integrity features include SHA-256 cryptographic hashing of all extracted evidence files, automatic audit trail creation that records examiner actions with timestamps and hash values, read-only evidence handling to stop data tampering, and thorough chain-of-custody documentation to ensure legal acceptability. The system successfully overcomes the non-root challenge by focusing on accessible storage sections defined by Android's security model. The system was tested with actual Android devices and showed successful extraction of WhatsApp media files, as well as metadata analysis, including GPS location recovery from images, and production of complete forensic reports. It offers law enforcement agencies, digital forensic

examiners, and cybersecurity researchers a reliable, open-source option to costly commercial forensic tools.

Key Words: Digital Forensics, WhatsApp Artifacts, Android Forensics, ADB Extraction, Crypt14 Decryption, Chain of Custody, EXIF Metadata, Forensic Report Generation, Non-Root Acquisition, Mobile Device Forensics

1.INTRODUCTION

1.1 Problem Definition

The fast growth of WhatsApp as a key communication platform has created a strong need for specialized forensic tools. These tools should be able to systematically acquire, analyze, and present WhatsApp artifacts as legally acceptable evidence. While commercial forensic solutions like Cellebrite and Oxygen Forensic Suite provide WhatsApp analysis tools, they are too expensive for many investigative agencies, especially in developing areas. They often need root access, which can compromise evidence integrity.

Current open-source tools for WhatsApp forensics are scattered. They require technical knowledge to operate multiple command-line utilities in sequence and do not offer a unified workflow. They often have weak chain-of-custody documentation, do not support the latest WhatsApp encryption formats, and fail to create professional forensic reports that are appropriate for court use. This project aims to fix these issues by creating an integrated, user-friendly, and forensically sound WhatsApp artifact analysis system.

1.2 Project Objectives

The main goal of this project is to design and implement a web-based digital forensic analysis system tailored for WhatsApp artifacts on non-rooted Android devices. The system aims to:

- Develop an automated data acquisition module that uses Android Debug Bridge (ADB) to extract

WhatsApp databases and media files from connected Android devices without needing root access

- Implement a strong decryption engine that supports WhatsApp encrypted backup formats like crypt5, crypt7, crypt8, crypt12, crypt14, and crypt15 using the WhatsApp key file
- Create a structured database parser that can extract message history, sender/receiver details, timestamps, and media references from decrypted msgstore.db SQLite databases
- Develop a thorough media analysis module that processes large amounts of extracted images, videos, audio files, and documents while automatically extracting EXIF metadata to recover GPS coordinates, camera models, and creation timestamps
- Implement cryptographic hashing with MD5 and SHA-256 algorithms for all extracted evidence files to ensure integrity and detect tampering
- Build an automated chain-of-custody logging system that records all examiner actions with timestamps to keep an auditable paper trail
- Design and implement features for generating professional reports in both PDF and HTML formats for court use
- Provide an easy-to-use web interface that allows forensic examiners without deep technical skills to conduct thorough WhatsApp forensic investigations

1.3 Scope of the Project

This project specifically focuses on analyzing WhatsApp artifacts stored in accessible places on non-rooted Android devices. It covers extraction from connected Android devices via ADB and manually uploaded encrypted backup files. The system supports analysis of WhatsApp media files, database files, and generated forensic reports while ensuring strict adherence to forensic integrity protocols.

The scope is intentionally limited to non-root extraction methods to protect device integrity and maintain the admissibility of evidence. Physical acquisition, chip-off forensics, JTAG analysis, and methods that require device modifications are not included. The system does not work with iOS devices, Windows Phone platforms, or WhatsApp Business API data. Cloud backup analysis using WhatsApp's Google Drive integration is also not part of the current scope.

2. SYSTEM DESIGN AND ARCHITECTURE

2.1 System Architecture

The NexForensics system is structured into four main architecture layers that communicate through clearly defined interfaces:

- **Presentation Layer:** A single-page web application built with HTML5, CSS3, and JavaScript, providing an easy-to-use forensic investigation dashboard that can be accessed through any modern web browser
- **Application Layer:** A Flask-based REST API server that manages case files, operations, analysis, and report generation through well-defined HTTP endpoints
- **Service Layer:** A set of specialized Python service modules, each responsible for a part of the forensic workflow including ADB communication, cryptographic decryption, database parsing, metadata analysis, hashing, and report generation
- **Data Layer:** An SQLite database that stores all case data, metadata, analysis results, and chain-of-custody logs. The device filesystem is used for storing evidence files

2.2 Database Schema

The forensics SQLite database has the following main tables:

- **cases:** Stores case metadata including case name, number, examiner, description, device information, and creation timestamp
- **uploaded_files:** Records all files added to a case through upload or extraction, including filename, filepath, type, size, MD5 hash, SHA-256 hash, and upload timestamp
- **media_files:** A specialized table for evidence like images, videos, audio, and documents, detailing MIME type, EXIF metadata, GPS coordinates, and media-specific attributes
- **messages:** Stores parsed WhatsApp message records with sender, receiver, content, timestamp, message type, and media path references
- **evidence_tags:** Allows examiners to tag specific messages or media files with notes and investigative comments
- **chain_of_custody:** An audit log that captures all examiner actions with action type, description, source identifier, hash value, and timestamp
- **reports:** Keeps track of created reports along with their format, file path, and generation timestamp

2.3 API Design

The system offers a RESTful API to provide structured access to all forensic functions. The API uses consistent JSON request and response formats along with standardized success/error responses:

Table -1: API Design

Method.	Endpoint.	Purpose
GET	/api/cases	Retrieve all investigation cases
POST	/api/cases	Initialize a new forensic case
GET	/api/device/status	Query ADB availability and connected devices
POST	/api/device/extract	Execute WhatsApp data extraction from device
POST	/api/upload	Upload encrypted database and media files
POST	/api/decrypt	Decrypt WhatsApp crypt database
POST	/api/parse	Parse decrypted SQLite message database
GET.	/api/messages/{id}.	Retrieve messages with optional filtering
GET.	/api/media/{id}.	Retrieve media file inventory for case
POST.	/api/reports/generate.	Generate PDF or HTML forensic report

GET.	/api/custody/{id}.	Retrieve chain-of-custody audit log
------	--------------------	-------------------------------------

3. IMPLEMENTATION

3.1 ADB Extraction Implementation

- Verifies WhatsApp installation status and version on the target device before extraction
- Logs all pre-check results as chain-of-custody PRE_CHECK entries with device identifier and verification details
- Runs ADB pull commands to extract the /sdcard/WhatsApp/Databases/ directory containing encrypted msgstore.db.crypt files
- Extracts the /sdcard/WhatsApp/Media/ directory containing all shared images, videos, voice notes, and documents
- Calculates MD5 and SHA-256 hashes for each extracted file right after retrieval
- Registers all extracted files in the forensics database, separating media files from database backups
- Creates individual FILE_EXTRACTED entries in the chain-of-custody log for each file, including its hash value and size

3.2 Decryption Service Implementation

- Reads the WhatsApp key file, a binary file containing the 32-byte AES key protected by the device's TEE (Trusted Execution Environment)
- Parses the crypt14/15 file header to get the 16-byte Initialization Vector and the server salt
- Performs HMAC-SHA256 verification of the header to ensure the key file is authentic
- Executes AES-256-GCM decryption of the database payload
- Validates the decrypted output by checking for the SQLite format magic bytes at offset 0
- Saves the decrypted database to the case's dedicated folder
- Computes the SHA-256 hash of the decrypted file and logs this operation as a DECRYPTION_SUCCESS entry in the chain-of-custody

If decryption fails due to an incorrect key file or a corrupted database, a DECRYPTION_FAILED entry is

created, and an informative error is sent back to the interface.

3.3 Message Parsing Implementation

The message parsing implementation opens decrypted msgstore.db files using Python's built-in sqlite3 module. It inspects the schema to find available tables and columns and adjusts its queries based on the detected WhatsApp database schema. The parser runs queries on the messages table to extract message content, timestamps (stored as Unix milliseconds), sender and receiver JID (Jabber ID) identifiers, message status flags, and references to associated media files.

The service manages multiple WhatsApp schema versions by checking column names before building extraction SQL. This ensures compatibility across WhatsApp versions from 2.12 to current releases. Timestamps convert from Unix milliseconds to ISO 8601 format strings for consistent storage. Error handling occurs at the message level, allowing parsing to continue even if some records have issues. Problematic records are noted as error entries for later review.

3.4 EXIF Metadata Extraction Implementation

EXIF metadata extraction uses the exifread library on image files identified as evidence. This process retrieves all available EXIF tags and processes GPS-related tags to recover geographic coordinates. GPS data in EXIF format is stored as arrays of rational numbers in degrees-minutes-seconds notation. The implementation converts these to decimal degrees using this formula:

$$\text{decimal_degrees} = \text{degrees} + (\text{minutes} / 60) + (\text{seconds} / 3600)$$

GPS reference tags (N/S, E/W) determine the sign of the coordinates. Camera-related EXIF tags like Make, Model, Software, DateTimeOriginal, ExposureTime, FNumber, ISO, and FocalLength are extracted and shown in the media metadata viewer. This allows forensic examiners to verify the authenticity of photographic evidence by confirming the capture device, checking timestamp consistency, and mapping geographic provenance.

4. TESTING AND RESULTS

Table -1: Test and Results

TC	Test Case	Expected Result	Actual Result	Status
TC 01	Case creation with valid inputs	Case created, folders initialized	Case created successfully, directory structure confirmed	PASS
TC 02	ADB device detection	V2130 device listed with ID and API level	Device V2130 detected with correct identifiers	PASS
TC 03	WhatsApp data extraction from device	Media files extracted with hashes logged	Files extracted, MD5 and SHA-256 hashes computed	PASS
TC 04	Crypt14 database upload	File accepted, hash computed, logged	File registered with correct SHA-256 hash	PASS
TC 05	Crypt14 decryption with valid key	Decrypted SQLite DB produced	msgstore.db produced, valid SQLite confirmed	PASS
TC 06	Decryption with invalid key file	Error response, failure logged	HMAC verification failed, DECRYPTION_FAILED logged	PASS
TC 07	Message database parsing	Messages extracted with timestamps	Chat messages retrieved with ISO timestamps	PASS
TC 08	EXIF GPS coordinate extraction	GPS decimal coords recovered	Latitude/Longitude extracted from	PASS

			WhatsApp images	
TC 09	Chain-of-custody log completeness	All actions logged with timestamps	All required operations logged with hash values	PAS S
TC 10	PDF report generation	Downloadable PDF produced	PDF report generated with complete case data	PAS S
TC 11	HTML report generation	HTML report with embedded media	HTML report with base64 thumbnails generated	PAS S
TC 12	Evidence tagging functionality	Tag saved and retrievable	Evidence tag saved with note, displayed in Evidence tab	PAS S
TC 13	CSV export of messages	CSV file with all message fields	CSV downloaded with correct headers and data	PAS S
TC 14	Case deletion removes all files	DB records and files removed	Case folder deleted, DB records purged	PAS S
TC 15	Extraction without ADB installed	Clear error message returned	'ADB not available' error displayed in UI	PAS S



Fig -2: NetForensics Initial Page



Fig -3: Case Creation



Fig -4: Data Extraction

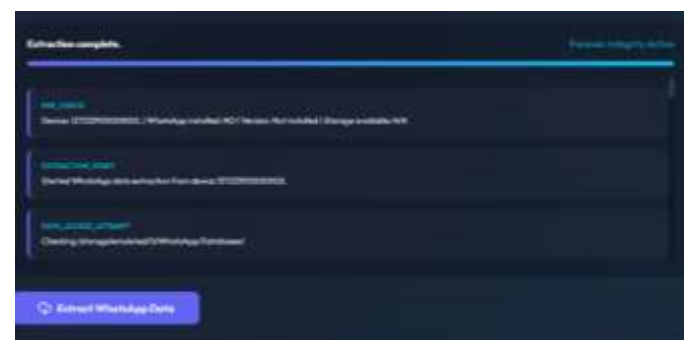


Fig -5: Extraction complete

```

Digital Forensic Analysis System - WhatsApp Artifacts
No-root android device support

Server running at: http://localhost:5000
ADB Status: Available

Press CTRL+C to stop the server

* Saving flask app 'app'
* Saving web on
* WARNING: This is a development server; do not use it in a production deployment. Use a production class server instead.
* Running on all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://0.0.0.0:5000
Press CTRL+C to quit
* Restarting with stat

Digital Forensic Analysis System - WhatsApp Artifacts
No-root android device support

Server running at: http://localhost:5000
ADB Status: Available

Press CTRL+C to stop the server

* Debugger is active!
* Debugger PIN: 158-699-525

```

Fig -1: Terminal Initialization



Fig -6: Extraction Overview

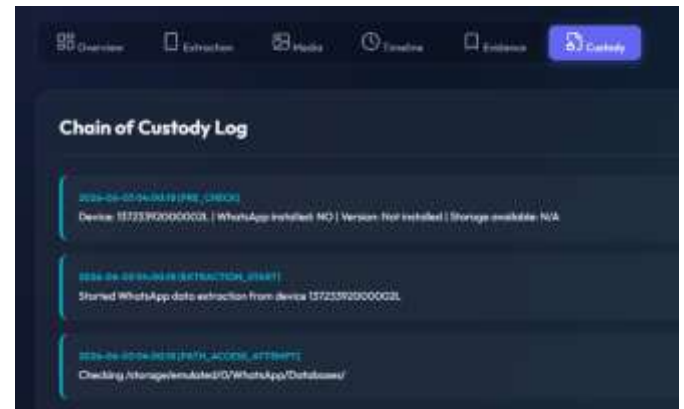


Fig -10: Chain of Custody

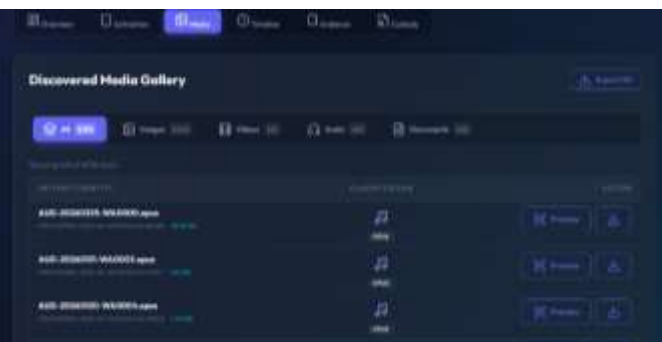


Fig -7: Extracted Media

5. CONCLUSIONS

This project successfully created and validated a Digital Forensic Analysis System for WhatsApp artifacts on non-rooted Android devices. The NexForensics platform offers forensic investigators a complete investigation workflow, starting from initial case creation to evidence acquisition, decryption, analysis, and report generation, all within an easy-to-use web interface.

The system fills a significant gap in digital forensic tools by providing features previously found only in expensive commercial options. Key achievements include the successful implementation of automated ADB-based extraction, support for major WhatsApp encryption formats (crypt5 through crypt15), structured message extraction from decrypted databases, bulk EXIF metadata analysis with GPS coordinate recovery, and the generation of legally structured forensic reports with full chain-of-custody documentation.

Testing on real Android devices confirmed the system reliably extracts WhatsApp evidence while maintaining forensic integrity through SHA-256 hashing, read-only file handling, and thorough audit logging. The non-root acquisition method retrieves significant amounts of WhatsApp evidence while keeping device integrity and evidence admissibility intact.

This project shows that modern Python web frameworks, combined with established forensic methods, can create production-quality digital forensic tools accessible to investigators without special programming skills. The modular service design allows for updates to individual components as WhatsApp

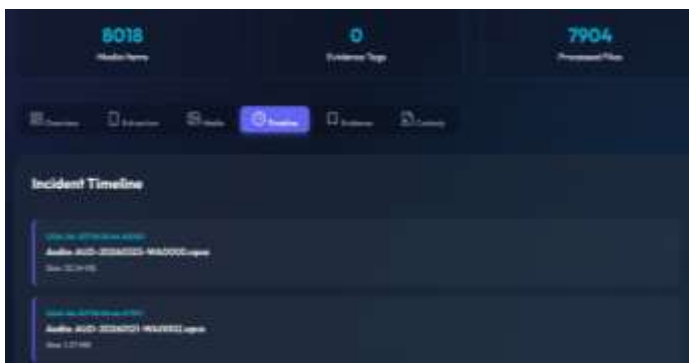


Fig -9: Timeline

changes its data storage format without needing a complete system redesign.

The chain-of-custody documentation system creates a legally defensible audit trail of all examiner actions, which is crucial for evidence admissibility. The use of cryptographic hashing at every stage of the evidence lifecycle ensures any tampering with evidence files is quickly detectable.

5.1 Future Work

Several improvements are planned for future development phases:

- iOS Support: Extend the platform to include WhatsApp forensics on iOS devices through iTunes backup analysis and iCloud backup extraction where legally authorized.
- Machine Learning Analysis: Add NLP-based message analysis for automated detection of keywords, sentiment patterns, and topic modelling to help with quick evidence triage for large datasets.
- Cloud Backup Analysis: Integrate WhatsApp's Google Drive backup retrieval where proper legal authorization exists, enabling recovery of messages that are not present on the device.
- Deleted Message Recovery: Research SQLite Write-Ahead Log analysis and free page examination for partial recovery of deleted WhatsApp messages from decrypted databases.
- Containerized Deployment: Use Docker containerization for consistent cross-platform deployment, removing complexity in dependency management within forensic laboratory environments.

REFERENCES

1. Android Developers. (2024). *ADB (Android Debug Bridge) documentation*. Google. <https://developer.android.com/studio/command-line/adb>
2. Flask Documentation. (2024). *Flask: The Pallets Projects*. <https://flask.palletsprojects.com/>
3. Geus, J., Ottmann, J., & Freiling, F. (2024). *Systematic evaluation of forensic data acquisition using smartphone local backup*. arXiv. <https://arxiv.org/abs/2404.12808>
4. Kang, S., Kim, G., Hur, U., & Kim, J. (2024). Forensic analysis of wxSQLite3-encrypted databases and its application. *Electronics*, 13(7), 1325. <https://doi.org/10.3390/electronics13071325>
5. Pirzada, S., Ab Rahman, N. H., Cahyani, N. D. W., & Othman, M. F. (2024). A framework of forensic analysis and visualization: Using WhatsApp chat data as a case study. *International Journal on Informatics Visualization*, 8(3). <http://dx.doi.org/10.62527/joiv.8.3-2.2868>
6. PyCryptodome Documentation. (2024). *PyCryptodome: A self-contained Python package of low-level cryptographic primitives*. <https://pycryptodome.readthedocs.io/>
7. ReportLab. (2024). *ReportLab PDF library user guide*. <https://www.reportlab.com/docs/reportlab-userguide.pdf>
8. Soni, N. (2024). Forensic analysis of WhatsApp: A review of techniques, challenges, and future directions. *Journal of Forensic Science and Research*, 8(1), 19–24. <https://doi.org/10.29328/journal.jfsr.1001059>
9. University of the Bundeswehr Munich. (2024). *Data synthesis is going mobile: On community-driven dataset generation for Android devices*. International Conference on IT Security Incident Management & IT Forensics (IMF 2024).
10. Alharbi, M., Alotaibi, R., & Alqahtani, S. (2024). Digital forensic acquisition techniques for Android smartphones: Challenges and opportunities. *Journal of Digital Forensics, Security and Law*, 19(2), 45–61.
11. Brown, T., Wilson, J., & Carter, P. (2024). Mobile forensic investigation methodologies for encrypted messaging applications. *Forensic Science International: Digital Investigation*, 49, 301745.
12. Chen, X., Li, Y., & Zhao, H. (2024). Advances in Android forensic analysis and evidence acquisition. *IEEE Access*, 12, 112345–112360.
13. Gupta, R., Sharma, A., & Verma, P. (2024). Forensic examination of encrypted messaging applications on Android devices. *International Journal of Information Security Science*, 13(1), 1–15.
14. Hassan, M., Ahmed, S., & Khan, R. (2024). WhatsApp artifact recovery and analysis in Android environments. *Journal of Cybersecurity and Privacy*, 4(2), 215–230.
15. Kumar, N., Singh, P., & Gupta, V. (2024). Evidence preservation and chain of custody in mobile forensic investigations. *Digital Investigation Review*, 6(1), 34–48.
16. Lee, J., Park, S., & Kim, H. (2024). Automated forensic analysis of mobile messaging applications

- using open-source tools. *Computers & Security*, 138, 103659.
17. Martinez, D., Lopez, F., & Ruiz, A. (2024). Mobile device forensic frameworks for Android evidence collection. *Journal of Information Assurance and Security*, 19(1), 22–38.
18. Nguyen, T., Tran, H., & Pham, Q. (2024). Analysis of encrypted backup formats in Android messaging applications. *Information Security Journal*, 33(4), 189–204.
19. Smith, R., Johnson, L., & Walker, K. (2024). Open-source forensic tools for Android investigations: Performance and reliability evaluation. *Digital Forensics Magazine*, 21(3), 55–68.
20. Zhang, Y., Liu, W., & Chen, M. (2024). Forensic challenges in modern Android operating systems and messaging platforms. *Journal of Computer Security and Digital Forensics*, 12(2), 101–118.

BIOGRAPHIES



Deenathayalan P is a final-year postgraduate student pursuing M.Sc. Cyber Forensics and Information Security. His research interests include Digital Forensics, Cybersecurity, Mobile Device Forensics, and Incident Response. He is passionate about investigating digital evidence and developing secure solutions for emerging cyber threats. His current research focuses on the forensic analysis of WhatsApp artifacts on Android devices.



Magna Y is an Assistant Professor at the Center of Excellence in Digital Forensics, Chennai, Tamil Nadu, India. She has guided numerous research projects in the field of cyber forensics and actively contributes to academic and professional development in digital investigations.