EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 3||pp 20-29||June||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

Incorporating Mobile Forensic Tools into a Comprehensive Toolkit (Nugget)

Asengewe Aude Michèle[1*]& Dr. KN, Jonathan[2]
[1,2]Faculty of Computing and Information Sciences, University of Lay Adventist of Kigali, Rwanda
Corresponding Author Email: asengewe3@gmail.com

Accepted: 28 April 2025 || Published: 05 June 2025

Abstract

Mobile devices have become an essential part of everyday life, playing a crucial role in various activities. Their widespread use provides digital forensic investigators with valuable insights when analyzing cases. Given the vast amount of data stored on mobile devices, their significance in digital forensic investigations has grown substantially. However, forensic investigators face major challenges due to the diversity of tools and lack of standardization in data representation. To address these challenges, Nugget a Domain-Specific Language (DSL) for digital forensics was developed. Nugget provides a structured approach to defining forensic computations while abstracting technical implementation details. It enables investigators to describe operations on digital evidence without needing to manage the underlying execution. Despite its benefits, Nugget initially lacked support for mobile forensic investigations. This study aimed to enhance Nugget's capabilities by integrating mobile forensic tools and extending its language to support mobile data analysis. Widely accessible forensic tools that support command-line execution on Android and iOS platforms were selected for integration. The implementation involved expanding Nugget's grammar, incorporating forensic tool outputs via RPC, and validating the framework using forensic corpora. Key findings show that the integration improved the interoperability of forensic tools, reduced inconsistencies in data handling, and enhanced investigative workflows. Comparative analysis with traditional approaches revealed increased accuracy and decreased processing time. This research successfully extended Nugget to support mobile forensic investigations, creating a unified and standardized framework for analyzing mobile data. The proposed solution not only addresses current gaps in forensic tool integration but also lays the groundwork for future enhancements, including greater automation and compatibility with additional tools.

Keywords: *Mobile Forensics, Digital Investigations, Standardization, Nugget, Interoperability.*

How to Cite: Michèle, A. A., & Jonathan, K. N. (2025). Incorporating Mobile Forensic Tools into a comprehensive Toolkit (Nugget). *Journal of Information and Technology, 5*(3), 20-29.

1. Introduction

Mobile devices store a wealth of sensitive information, making them prime targets in digital forensic investigations. The rapid increase in the number of digital forensic solutions has further expanded these distinctions, making it necessary for investigators to develop expertise in using them effectively and interpreting their findings accurately. A lack of universally accepted guidelines for forensic procedures and data interpretation has contributed to inconsistencies in how these tools are developed and utilized. According to (Stelly, (2018))the existence of multiple forensic solutions and the diversity in digital evidence sources have created significant challenges in standardization. However, forensic examiners face significant challenges due to the lack of standardized tools and inconsistencies in forensic data representation. Determining the most appropriate tool for a specific forensic operation requires a thorough evaluation of its functionality. To conduct an assessment, an investigator must first gain familiarity with the tools under review. Following this, a structured outline of the task requirements and success criteria is developed. The selected tool is then tested against these predefined standards, either manually or through automated methods. After analyzing the tool's performance, the investigator can determine whether it meets the intended objectives and produces accurate results.This paper explores the integration of mobile forensic tools into Nugget (an external DSL that allows users to describe forensic computations) to standardize forensic evidence processing and enhance investigative efficiency.

1.1 Problem Statement

Current mobile forensic tools generate varied outputs, making forensic data analysis cumbersome. Integrating these tools into a standardized framework like Nugget can address interoperability issues and improve forensic workflows.

1.2 Research Objectives

- Evaluate existing forensic standards for mobile data analysis.
- Analyze current mobile forensic tools and their interoperability challenges.
- Develop a framework to integrate forensic tools into Nugget.
- Assess the effectiveness of the integration through forensic case studies.

1.3 Research Scope

Given the vast number of mobile operating systems and forensic solutions, this research focused primarily on free-access or widely accessible investigative software that supports command-line execution. The study was confined to forensic tools applicable to Android and iOS platforms to maintain clarity and feasibility.

2. Literature Review

Numerous studies have highlighted the challenges of mobile forensic investigations, including data extraction complexities, tool compatibility issues, and evidence validation concerns. While existing forensic tools such as Autopsy and Cellebrite provide digital investigation capabilities, they lack a standardized framework for data representation. Currently, Nugget

offers a promising solution by ensuring consistency and interoperability in forensic data storage and analysis as shown in the figure below.

2.1. Methods of Acquiring Data from Portable Devices

Mobile forensic investigations rely on various techniques to retrieve data from devices, which can be broadly categorized into two main methods: Logical Extraction and Physical Extraction. Each approach has distinct advantages and limitations depending on the security measures and accessibility of the device in question.

Structured data retrieval utilizes specialized software to extract information from a portable device's available storage sections. This process relies on the device's operating system or other management software to facilitate data retrieval. While logical extraction is less invasive and preserves the device's integrity, it is limited to the information that the operating system permits users or applications to access. Advanced security features in modern mobile devices often restrict access to deeper system files and raw storage, thereby reducing the amount of forensic data that can be obtained through this method.

Physical Extraction, on the other hand, allows forensic analysts to bypass the operating system and extract raw memory or storage data directly from the device. This approach provides a more comprehensive view of the stored information, including deleted files, system logs, and hidden data. Specialized forensic tools and hardware are required to perform physical extraction, and in some cases, embedded storage components such as embedded MultiMediaCards (eMMC) may need to be removed for direct analysis. Despite its effectiveness, this method is highly invasive and carries the risk of damaging the device, making it a last resort when logical extraction is insufficient.

In addition to these two primary methods, (Parr, 2014) proposed a hierarchical framework for categorizing mobile forensic data retrieval techniques. This classification system, illustrated in Figure 1, organizes different forensic acquisition methods into five levels, each representing varying degrees of data accessibility and forensic complexity.

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 3||pp 20-29||June||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

Figure 1: Hierarchical Structure of Brothers' Tool Classification

As forensic analysis techniques become more advanced, they also become increasingly specialized, requiring sophisticated tools, greater technical expertise, and longer processing times. Additionally, these methods tend to be more invasive, posing potential risks to the integrity of the device being examined (Omondi, February 3, 2019).

- **Manual extraction**

One of the most basic methods of obtaining information from a mobile device involves manually navigating its interface using physical buttons or a touchscreen. Investigators visually inspect and document displayed information, often capturing evidence by photographing the screen. Despite its simplicity, this method has notable drawbacks, including the potential for human error, the inability to retrieve hidden or deleted data, and the risk of unintentionally modifying information—such as marking unread messages as read. Additionally, if a device is damaged or unresponsive, this approach becomes ineffective.

- **Logical extraction**

Another widely used method involves linking the mobile device to a computer through either a wired connection (such as a USB cable) or a wireless method (like Bluetooth or WiFi). The connected system issues commands to retrieve specific information from the device, which then transmits the requested data. Although this method is more structured than manual extraction, its effectiveness is restricted by the device's security settings and operating system limitations.

- **Hex dump and JTAG Extraction**

A hex dump involves extracting raw data from a device's memory by deploying a custom bootloader. This allows forensic specialists to retrieve a complete memory dump, bypassing standard data access restrictions. JTAG extraction is a technique based on an Institute of

Electrical and Electronics Engineers (IEEE) standard for testing and debugging hardware. It allows forensic analysts to access the raw memory of a device via a Test Access Port (TAP) on its circuit board. By connecting specialized forensic tools to these standardized test ports, investigators can retrieve stored data without relying on the device's original software. However, this method requires advanced skills and is highly technical.

- **Chip-off Forensics**

When other methods fail or a device is severely damaged, forensic specialists may resort to chip-off analysis. This technique involves physically removing the memory chip from the device's motherboard and using specialized equipment to read its stored data.

- **Micro read**

A highly intricate forensic technique, micro read analysis, involves examining a device's NAND or NOR memory chips at a microscopic level. Investigators use an electron microscope to analyze the physical structure of the chip and interpret stored binary sequences (0s and 1s), which are later converted into readable text, such as ASCII characters (Murphy, 2011)This process is extremely labor-intensive and time-consuming, making it a rare approach in standard forensic investigations.

2.2. Overview of the Nugget System

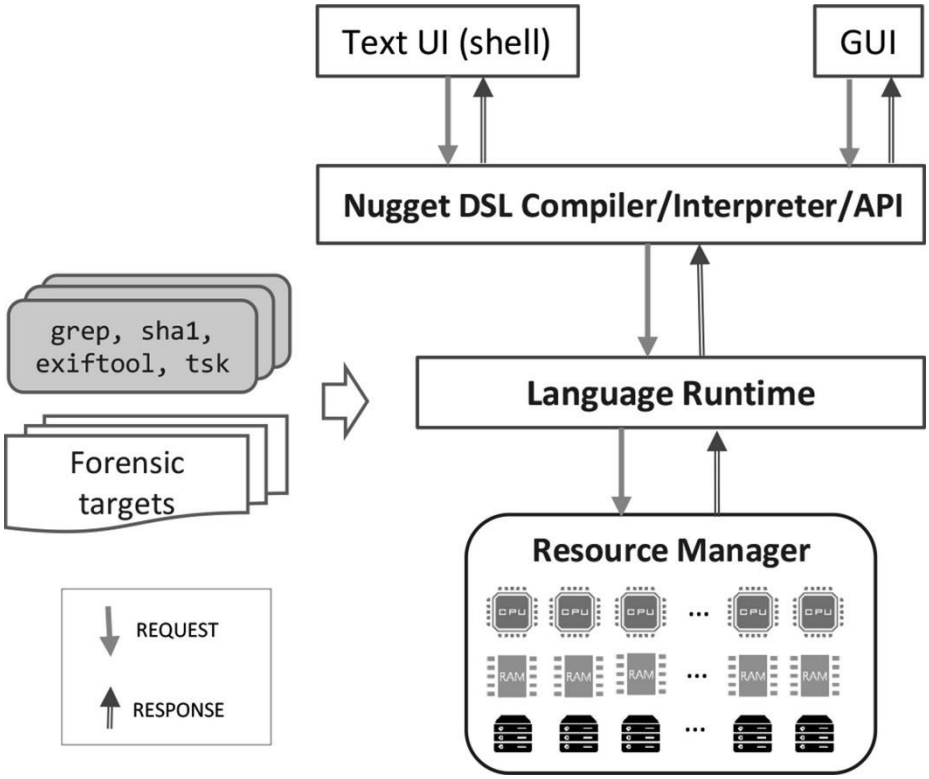
Nugget is a Domain-Specific Language (DSL) developed to facilitate digital investigative computations. According to (Mikhaylov I. &., 2016), a DSL is a programming language that is specifically tailored to a particular domain, offering limited but highly focused functionality. DSLs can be grouped into three main types: standalone DSLs, embedded DSLs, and language development environments. Standalone DSLs operate separately from any general-purpose programming language, maintaining their own syntax and execution framework, defining their own syntax, parsing mechanisms, and compilers. Common examples of such languages include SQL and Awk. Internal DSLs, in contrast, operate within the structure of an existing programming language, utilizing its syntax while expanding its capabilities for a particular domain. Nugget belongs to the category of external DSLs, which allows it to function autonomously rather than being embedded within another programming language. This independent nature ensures flexibility in defining forensic computations without restricting users to a specific software environment. The design of Nugget enables forensic analysts to describe digital forensic procedures in a structured and consistent manner, streamlining investigations and minimizing reliance on multiple forensic tools.A demonstration of these operators in action can be found in Listing1, which provides a practical example of how Nugget facilitates forensic computations.

```
1  files = "file:harddrive.EO1" | extract as ntfs [63,512]
2  jpgs  = files | filter name=="*.jpg"
3  hashes = jpgs.content | sha1, md5
4  jpgs = jpgs | add hashes
5  print jpgs
```

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 3||pp 20-29||June||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

Extracting data from forensic sources involves several stages, starting with the retrieval of relevant information before passing it through a sequence of transformations. Data extractors are essential in retrieving and isolating information from a specific source. As demonstrated in Listing-1, the initial step involves obtaining documents from an NTFS storage snapshot, starting at the 63rd block with a defined size of the block as 512. Once the data has been retrieved, it moves through filters that refine the results by either excluding or including specific types of information. For example, the second line in the listing illustrates the process of filtering out images with a .jpg extension from the retrieved data set. After the filtration stage, the modified dataset undergoes additional processing through transformation modules, which produce new outputs derived from the extracted information. A case in point is seen in the third line of the listing, where unique cryptographic signatures are computed for each document within the refined dataset. Finally, serializers changed the processed files in structured formats, making it accessible for forensic investigators. This may include text-based representations or integration into specialized digital evidence containers such as the Advanced Forensic Format (AFF), which ensures compatibility across different forensic tools.

2.3. Structure and Framework of Nugget



Nugget allows user interaction through either a Text User Interface (TUI) or a Graphical User Interface (GUI). The system processes commands written in Nugget DSL, which are interpreted and executed by the runtime environment. These instructions guide forensic tools in analyzing digital evidence. The results are then presented to the investigator via the chosen

interface. A resource manager oversees the process, handling task scheduling, logging operations, and delivering computed results.

Nugget operates using Context-Free Grammar (CFG), structured in Extended Backus-Naur Form (EBNF) and implemented through ANTLR (Lillis, 2016). ANTLR handles both lexical and syntax analysis, allowing the system to interpret and execute instructions within its Domain-Specific Language (DSL).

At present, the grammar of Nugget is influenced by the tools it integrates. For example, sha1 and listof-sha1 are linked to the SHA-1 hashing function, while sha256 and listof-sha256 correspond to SHA-256. As more forensic tools become part of the system, Nugget's grammar will continue evolving, incorporating additional language elements to support new functionalities.

3. Materials and Methods

3.1. Extensibility of Nugget

There are numerous tools used in digital forensic investigations. A majority of the tools are not yet integrated with Nugget. This requires that Nugget accommodate extra tools that it does not integrate with by default. Currently, Nugget does this by allowing end users to extend the DSL and write some boiler-plate code. Users need to generate language constructs using ANTLR to extend the DSL thereby extending Nugget to use the desired tools. Since the target audience for Nugget is both technical and non-technical users, the latter may find this approach to extensibility too technical. Due to the loose coupling brought about the separation of forensic tools from Nugget, developers extend Nugget. Some of how Nugget's extensibility can be accomplished include:

Extending the DSL: Nugget can add new functionality by updating the DSL. The implementer of the new functionality would have to generate and compile the code necessary to ac- complish the new functionality.

Extending the source code: -Nugget's source code can be modified to incorporate new features. The implementer would have to compile the new code into Nugget.

Extending capabilities via RPC: -Nugget communicates with forensic tools via RPC. The addition of a new tool means creating, building and running a new RPC target.

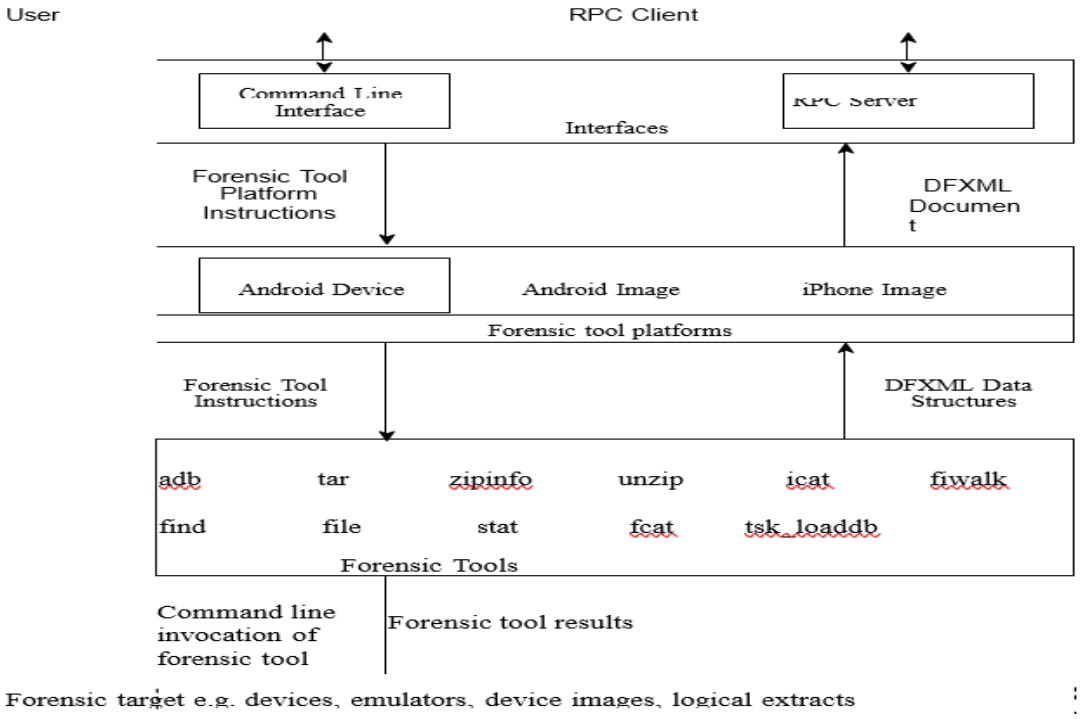
The process of extending Nugget would be accomplished by (*a*) identifying the data type to be consumed and produced, (*b*) incorporate the tool's functionality into a container, and (*c*) build Nugget to add the extension into the DSL grammar. For now, the extension of Nugget requires some level of technical or programming knowledge to be able to incorporate new features.

3.2. Conceptual Framework and Research Results

The conceptual architecture for the forensic tool wrapper is depicted in the Figure illustrating the structure and interaction of the integrated mobile forensic tools within Nugget's system as Nugget is able to add new functionality by updating the DSL. The implementer of the new

EdinBurg Peer Reviewed Journals and Books Publishers
Journal of Information and Technology
Vol. 5||Issue 3||pp 20-29||June||2025
Email: info@edinburgjournals.org||ISSN: 3080-9576

functionality would have to generate and compile the code necessary to accomplish the new functionality.



The architecture is structured into three fundamental layers, each responsible for distinct functions in the forensic process.

- Interfaces Layer: This layer provides access points for interacting with forensic tool platforms. Users can access these platforms via a command-line interface, while the Nugget runtime connects through the RPC Server to facilitate forensic processing.
- Forensic Tool Platforms: This layer manages interactions with different mobile operating systems, including Android and iOS. For instance, the Android Debug Bridge (ADB) specializes in handling Android devices, whether physical or emulated. The components within this layer receive commands from the interface and generate DFXML documents as output. Core forensic computations are delegated to the forensic tools layer.
- Forensic Tools: This layer is responsible for executing forensic operations by interacting directly with forensic utilities. Outputs from this layer are structured as DFXML objects, with file-based operations generating FileObject representations.

4. Methodology

This study adopts an exploratory research design involving the following steps:

- Literature Review: Analysis of forensic standards such as DFXML and forensic tool evaluations.

- Tool Assessment: Examination of leading forensic tools and their data structures.
- Framework Development: Designing the integration model for Nugget using standardized forensic data formats.
- Validation: Testing the integrated framework using real-world forensic corpora.

5.. Results and Discussion

5.1 Key Findings

1. Improved interoperability of mobile forensic tools through Nugget's standardized format

One of the most significant outcomes of this research was the enhanced interoperability between various mobile forensic tools achieved through the Nugget platform. Traditionally, mobile forensic tools generate outputs in distinct formats, making it difficult to analyze and combine data from multiple sources. By integrating these tools into the Nugget framework and employing a standardized data representation (e.g., DFXML), the system now facilitates seamless communication and data exchange among previously incompatible tools. This interoperability allows forensic analysts to conduct more comprehensive investigations without the need to manually convert or normalize datasets, thereby saving time and reducing the risk of data loss or misinterpretation.

2. Enhanced forensic workflow efficiency, reducing data inconsistencies

The incorporation of mobile forensic tools into Nugget significantly improved the efficiency of forensic workflows. In conventional investigative setups, switching between tools often leads to duplicated efforts, increased processing time, and inconsistencies in how evidence is handled. With the integration into Nugget's structured DSL environment, forensic operations such as extraction, filtering, transformation, and hashing can now be executed in a uniform and repeatable manner. This standardization not only speeds up the investigation process but also reduces errors caused by manual intervention. The streamlined process ensures that investigators can follow consistent procedures, enhancing both the reliability and traceability of their findings.

3. Streamlined evidence management, improving forensic data analysis and case processing

Another key finding was the improvement in evidence management and case processing. By consolidating mobile forensic operations into a single, coherent framework, Nugget allows for better organization, tracking, and reporting of digital evidence. Investigators can manage complex evidence chains with greater clarity and confidence, thanks to the structured and serialized outputs generated by the platform. These outputs are easier to audit, interpret, and present in legal or academic settings. The result is a more effective and transparent investigative process, leading to faster case resolution and increased confidence in the integrity of digital evidence.

A comparison between traditional forensic methods and the Nugget-integrated framework reveals a significant improvement in forensic efficiency and standardization. Graphical

representations of test results demonstrate the reduction in processing time and enhanced accuracy of forensic investigations.

6. Conclusion

This study successfully integrates mobile forensic tools into Nugget, providing a standardized approach for digital investigations. The proposed framework enhances forensic efficiency and interoperability, addressing key challenges in mobile forensic analysis.

7.Recommandations

Future work should focus on automating forensic data analysis and expanding Nugget's capabilities to include additional forensic tools.

References

Edgar, T. W. (2017). Research Methods for Cyber Security. Cell phone & email forensics investigation cracks NYC. Elsevier Inc. Forensicon Inc.

Fowler, M. (2010). Domain Specific Languages. Addison-Wesley Professional.

Garfinkel, S. (2012). Lessons learned writing digital forensics tools and managing a 30TB digital evidence corpus. Digital Investigation. The Proceedings of the Twelfth Annual DFRWS Conference.

Garfinkel, S. F. (2009). Bringing science to digital forensics with standardized forensic corpora. Digital Investigation, 6, S2–S11.

Kothari, C. R. (2004). Research methodology methods and techniques (2nd ed. New Age.

Levine, B. N. (2009).). DEX: Digital evidence provenance supporting reproducibility and comparison. Digital Investigation,The Proceedings of the Ninth Annual DFRWS Conference.-6,-S48–S56-.

Lillis, D. B. (2016). Current Challenges and Fu.

Mikhaylov, I. &. (2016). Chip-off technique in mobile forensics.

Mikhaylov, I. &. (2016). Chipoff technique in mobile forensics. from https://www.digitalforensics.com/blog/chipofftechnique inmobile forensics.

Murphy, C. A. (2011). Developing Process for Mobile Device Forensics.

Omondi, M. (February 3, 2019). Kenol ceo's phone seized in insider trading probe.

Parr, T. (2014). The Pragmatic Programmers, LLC. The definitive ANTLR 4 reference.

Raghavan, S. (2013). Digital forensic research: Current state of the art. CSI Transactions.

Roussev, V. B. (2016). Digital forensic science: Issues, methods, and challenges. Morgan & Claypool: Retrieved from https: / / ieeexplore . ieee . org / document/7809443.

Skulkin, O. T. (2018). Learning Android Forensics (2nd ed.). Packt.

Stelly, C. &. (2018). Nugget: A digital forensics language. In DFRWS 2108.