Comparative Analysis of Full File System and Advanced Logical Extractions: A Deep Dive into Retrieved Data Types and Forensic Implications

Author: Olivia D. Renfro

Reviewed and Acknowledged: George Rodriguez

The purpose of this research is to examine the differences in file types recovered from Full File System (FFS) and Advanced Logical extractions to better align mobile data extraction methods with case-specific investigative needs. As mobile devices play a crucial role in criminal and civil investigations, understanding the depth and scope of data retrieved by each method is essential for selecting the most effective approach to recover relevant digital evidence.

This research paper will compare the file structures of a FFS extraction and an Advanced Logical extraction, highlighting the differences in data accessibility and organization. We will examine how FFS provides deeper access to system directories, application sandboxes, and encrypted data, while advanced logical extraction is limited to user-accessible content. Additionally, we will analyze the depth of data retrieval in FFS compared to advanced logical, focusing on the extent to which deleted, hidden, or cached information can be recovered. Lastly, we will compare the types of forensic artifacts extracted by each method, including SMS messages, third-party app data, logs, and system files, to determine the advantages and limitations of both techniques in forensic investigations.

This research utilizes datasets extracted using two industry-leading forensic tools: a Full File System (FFS)[1] extraction performed with Magnet Verakey and an Advanced Logical[2] extraction conducted with Cellebrite. By comparing the file structures obtained through these methods, investigators can gain valuable insight into the depth and scope of data available from each approach. FFS extractions provide access to system files, databases, and other artifacts typically restricted in logical extractions, while Advanced Logical methods focus on user-accessible data such as messages, call logs, and app data. Analyzing these differences helps forensic examiners determine which extraction method is best suited for specific investigative needs, ensuring that no critical evidence is overlooked in criminal or civil cases.

[1] A Full File System extraction is a forensic data acquisition method that provides complete access to a mobile device's file system, including system partitions, application data, and hidden or protected files. This method allows forensic examiners to recover a wide range of artifacts, such as deleted data, system logs, app databases, and encryption keys, which are typically inaccessible through logical extractions.

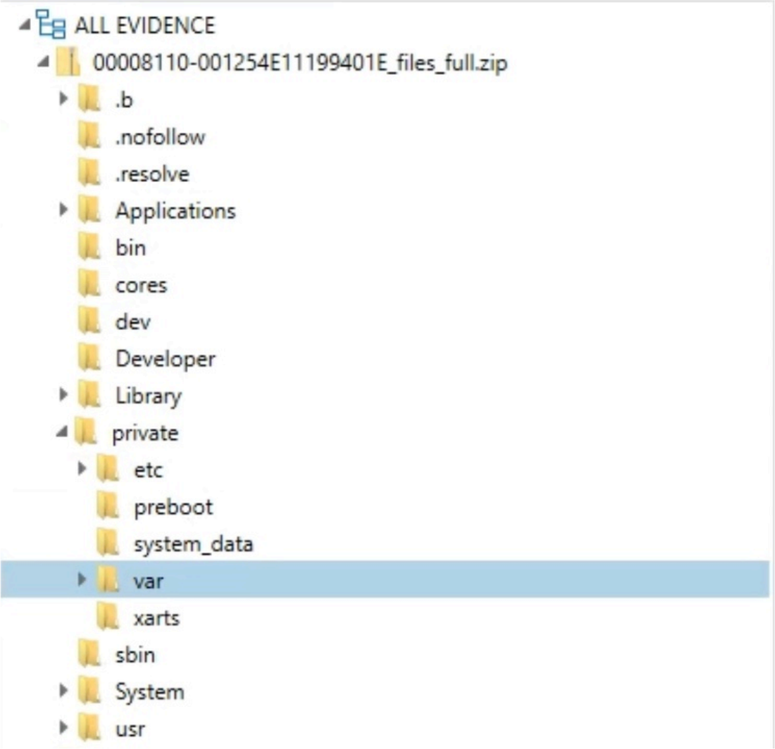
[2] An Advanced Logical extraction is a forensic acquisition technique that retrieves user-accessible data from a mobile device, including messages, call logs, media files, and application data. While more comprehensive than a basic logical extraction, it does not provide access to system-level files or certain protected areas of the device's storage.

The device under examination is an iPhone 14 running iOS 18.3.0 with 128 GB of storage, of which 107.02 GB is used. As a personal device, it is primarily utilized for daily activities, with the majority of storage consumed by applications such as Spotify, Snapchat, and Facebook. These apps suggest heavy media streaming, social interaction, and communication usage. The user maintains a high average screen time of 20 hours per day, indicating extensive engagement with the device for entertainment, messaging, and social media.

We will begin by analyzing the file structure of a FFS extraction to identify key directories and the artifacts they contain. This examination will focus on critical locations within the file system, such as application data directories, system logs, and user-generated content, to determine the depth and scope of recovered data. By mapping these directories and their contents, we aim to highlight the types of forensic artifacts available, including messages, cached files, deleted records, and encryption keys. Understanding the organization of these files will provide valuable insight into how different extraction methods retrieve data and the investigative value of specific directories in forensic analysis.

The first directory we will examine is the **/private/var** directory, a critical location in an iOS FFS extraction. This directory contains a wealth of forensic artifacts, including application data, system logs, caches, and various databases that store user interactions.

Figure 1



```
ALL EVIDENCE
  00008110-001254E11199401E_files_full.zip
    .b
    .nofollow
    .resolve
    Applications
    bin
    cores
    dev
    Developer
    Library
    private
      etc
      preboot
      system_data
      var
      xarts
    sbin
    System
    usr
```

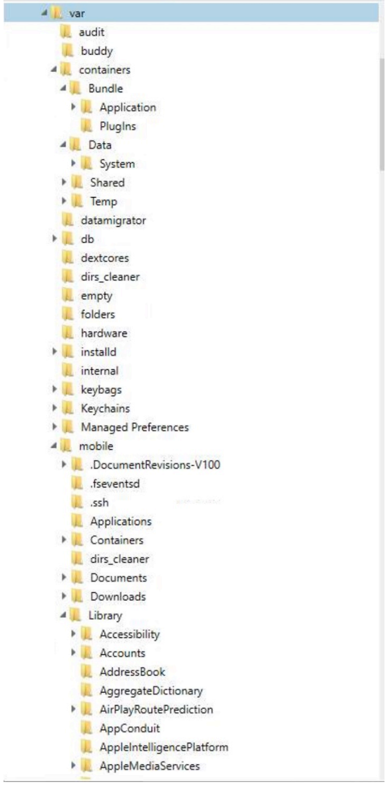
FFS File Structure

Observations

Key subdirectories within **/private/var**, hold essential evidence like messaging databases, call history, and location data. By analyzing the structure and contents of this directory, we can identify the types of data stored within it and assess their relevance to forensic investigations. Understanding the significance of **/private/var** helps examiners maximize the evidentiary value of FFS extractions in mobile forensic cases. Reference Figure, additionally, subdirectories like /**private/var/mobile/Library**[3] hold communication records from apps such as iMessage and WhatsApp, while **/private/var/db**[4] can contain important system records, including metadata about the device's usage. Because much of this data is not accessible through logical extractions, examining **/private/var** in an FFS extraction provides forensic examiners with deeper insight into a user's activities.

Let's take a look at what other crucial directories are present in a FFS extraction.

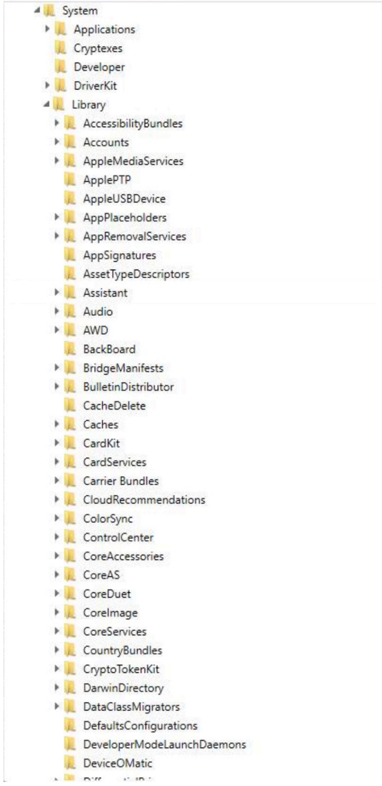
Figure 2



```
var
  audit
  buddy
  containers
    Bundle
      Application
      PlugIns
    Data
      System
    Shared
    Temp
  datamigrator
  db
  dextcores
  dirs_cleaner
  empty
  folders
  hardware
  installd
  internal
  keybags
  Keychains
  Managed Preferences
  mobile
    .DocumentRevisions-V100
    .fseventsd
    .ssh
    Applications
    Containers
    dirs_cleaner
    Documents
    Downloads
    Library
      Accessibility
      Accounts
      AddressBook
      AggregateDictionary
      AirPlayRoutePrediction
      AppConduit
      AppleIntelligencePlatform
      AppleMediaServices
```

FFS /private/var

Figure 3



```
System
  Applications
  Cryptexes
  Developer
  DriverKit
  Library
    AccessibilityBundles
    Accounts
    AppleMediaServices
    ApplePTP
    AppleUSBDevice
    AppPlaceholders
    AppRemovalServices
    AppSignatures
    AssetTypeDescriptors
    Assistant
    Audio
    AWD
    BackBoard
    BridgeManifests
    BulletinDistributor
    CacheDelete
    Caches
    CardKit
    CardServices
    Carrier Bundles
    CloudRecommendations
    ColorSync
    ControlCenter
    CoreAccessories
    CoreAS
    CoreDuet
    CoreImage
    CoreServices
    CountryBundles
    CryptoTokenKit
    DarwinDirectory
    DataClassMigrators
    DefaultsConfigurations
    DeveloperModeLaunchDaemons
    DeviceOMatic
```

FFS /system/library

[3] Encompasses additional system and application data (beyond what is backed up) such as caches and logs

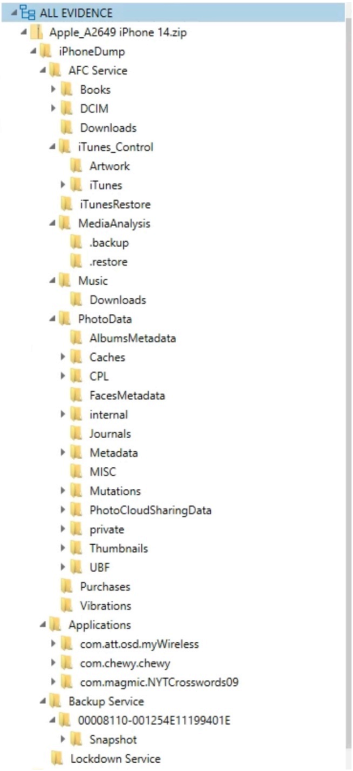
[4] Contains critical system databases (e.g., configuration, security, and metadata) that are captured only in an FFS acquisition.

Observations

The **/System/Library**[5] directory in an iOS FFS extraction is a crucial location that contains essential system files, frameworks, and configuration settings that govern the device's operation. This directory holds data related to core iOS functions, including system preferences, default application settings, and security policies. One of its key subdirectories, **/System/Library/Caches**, can contain logs, temporary files, and metadata that may provide insight into system activity. Additionally, **/System/Library/Fonts** and **/System/Library/KeyboardLayouts** store information about text input and user interactions, which can be relevant in forensic investigations involving communication analysis. While this directory does not typically contain user-generated content, its files can be instrumental in understanding system behavior, verifying device integrity, and analyzing forensic artifacts such as timestamps, system logs, and security configurations. This makes **/System/Library** an important resource when examining the deeper structure of an iOS device in a forensic investigation.

Now lets take a look at the advanced logical, and what directories can be found.

Figure 4



```
ALL EVIDENCE
  Apple_A2649 iPhone 14.zip
    iPhoneDump
      AFC Service
        Books
        DCIM
        Downloads
        iTunes_Control
          Artwork
          iTunes
        iTunesRestore
        MediaAnalysis
          .backup
          .restore
        Music
          Downloads
        PhotoData
          AlbumsMetadata
          Caches
          CPL
          FacesMetadata
          internal
          Journals
          Metadata
          MISC
          Mutations
          PhotoCloudSharingData
          private
          Thumbnails
          UBF
        Purchases
        Vibrations
      Applications
        com.att.osd.myWireless
        com.chewy.chewy
        com.magmic.NYTCrosswords09
      Backup Service
        00008110-001254E11199401E
          Snapshot
      Lockdown Service
```

AL File Structure

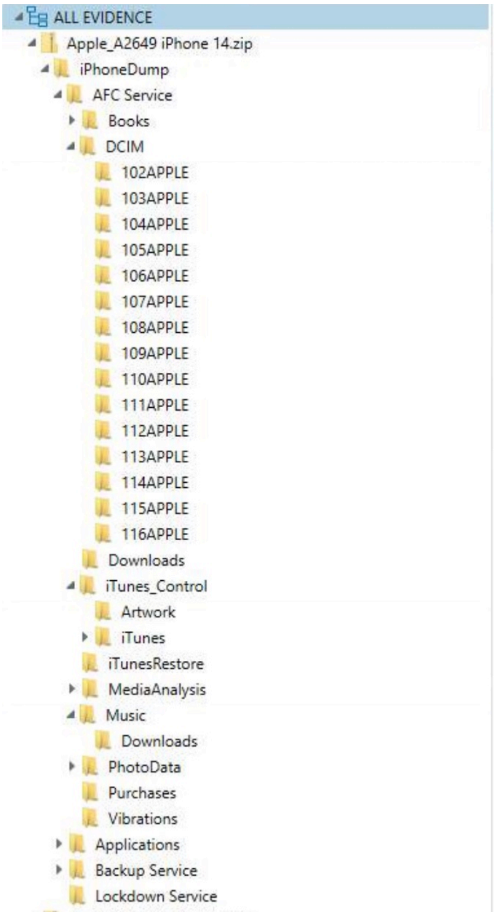
[5] reference Figure 3

Observations

It was observed that the Advanced Logical extraction (Figure 4) was unable to retrieve the **/private/var** and **/System/Library** directories. These directories are crucial for accessing system-level files, application data, and other important artifacts that are often vital in forensic investigations. The **/private/var** directory contains hidden or protected files, including messaging databases, app caches, and system logs, while **/System/Library** holds essential system files and configurations that help explain device behavior and settings. Since the Advanced Logical extraction focuses on user-accessible data, it does not have the capability to access these deeper system directories, which are only available through a Full File System (FFS) extraction. This limitation highlights the importance of utilizing FFS extractions when a comprehensive understanding of a device's data is necessary for an investigation.

Let's identify what the advanced logical was able to analyze.

Figure 5



```
ALL EVIDENCE
  Apple_A2649 iPhone 14.zip
    iPhoneDump
      AFC Service
        Books
        DCIM
          102APPLE
          103APPLE
          104APPLE
          105APPLE
          106APPLE
          107APPLE
          108APPLE
          109APPLE
          110APPLE
          111APPLE
          112APPLE
          113APPLE
          114APPLE
          115APPLE
          116APPLE
        Downloads
        iTunes_Control
          Artwork
          iTunes
        iTunesRestore
        MediaAnalysis
        Music
          Downloads
        PhotoData
        Purchases
        Vibrations
      Applications
      Backup Service
      Lockdown Service
```

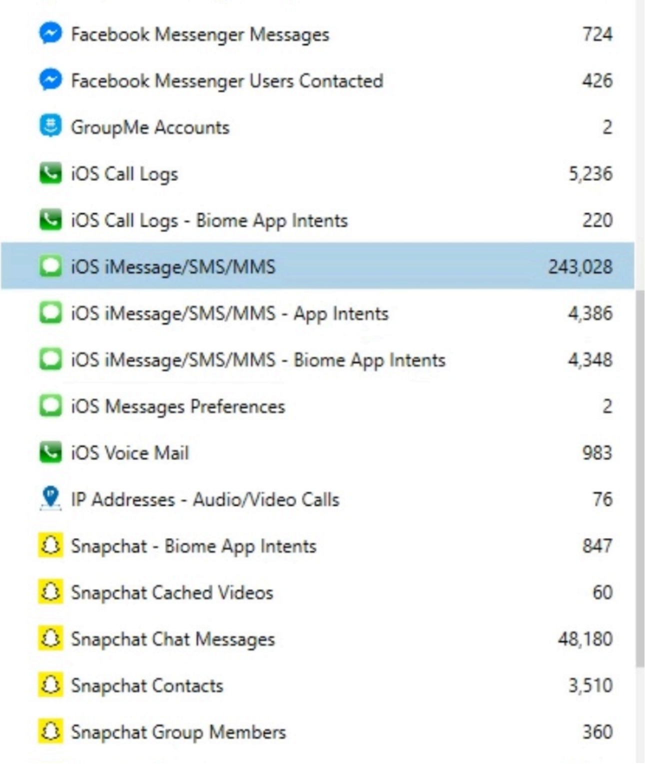
AL /DCIM

Observations (Figure 5)

The extracted file contains several important directories related to user data, system files, and applications. The DCIM and PhotoData directories store photos, videos, and related metadata, while Thumbnails aids quick access to media. iTunes_Control and iTunes Restore manage iTunes backups and synchronization, while Music, Books, and Downloads store locally saved content. For application data, the Applications directory holds app settings and files, while Caches and Metadata store temporary and structured information. From a forensic perspective, Backup Service and Snapshot track backups and system states, while Lockdown Service contains authentication files. The private directory secures restricted system and app data, and Purchases logs media and app transactions. Specialized directories like MediaAnalysis, Journals, and Mutations store system logs and forensic artifacts, while app-specific folders (com.chewy.chewy, com.magmic.NYTCrosswords09) contain user activity data. Collectively, these directories offer insights into device usage, stored content, and potential forensic evidence.

After identifying the differences in file structure, lets see where the same data is being stored and what causes the differences in artifacts found.

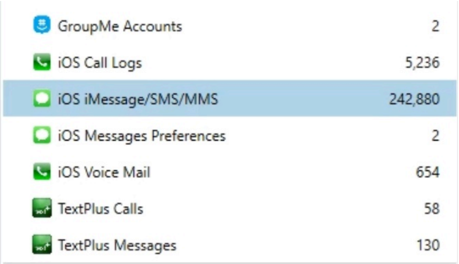
Figure 6



Facebook Messenger Messages	724
Facebook Messenger Users Contacted	426
GroupMe Accounts	2
iOS Call Logs	5,236
iOS Call Logs - Biome App Intents	220
iOS iMessage/SMS/MMS	243,028
iOS iMessage/SMS/MMS - App Intents	4,386
iOS iMessage/SMS/MMS - Biome App Intents	4,348
iOS Messages Preferences	2
iOS Voice Mail	983
IP Addresses - Audio/Video Calls	76
Snapchat - Biome App Intents	847
Snapchat Cached Videos	60
Snapchat Chat Messages	48,180
Snapchat Contacts	3,510
Snapchat Group Members	360

FFS

Figure 7



GroupMe Accounts	2
iOS Call Logs	5,236
iOS iMessage/SMS/MMS	242,880
iOS Messages Preferences	2
iOS Voice Mail	654
TextPlus Calls	58
TextPlus Messages	130

AL

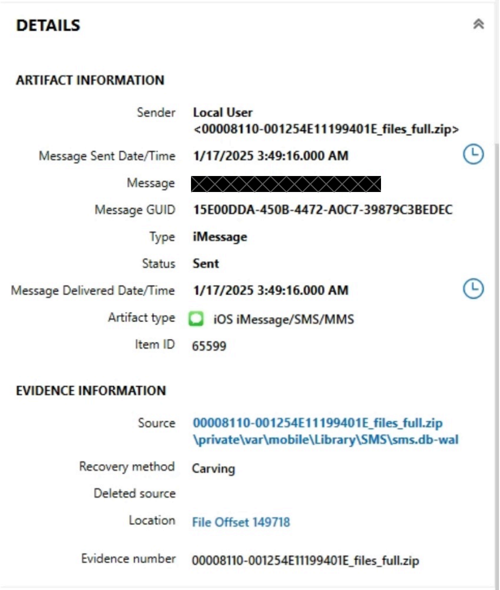
Observations

A FFS extraction can recover more SMS messages from an iPhone than an advanced logical extraction because it provides deeper access to the device's storage. iPhones store text messages in an SQLite database called **sms.db**, and while an advanced logical extraction can pull active messages, it often misses deleted texts, message fragments, and other hidden data. Since a FFS dump captures everything, including deleted records, metadata, and unallocated storage, it gives forensic analysts a better chance of recovering more messages. This method also grabs related artifacts like message attachments and timestamps, which might not be accessible through an advanced logical extraction. The FFS captured 243,028 SMS/MMS artifacts, while the advanced logical located 242,880 SMS/MMS artifacts. Both the Advanced Logical and FFS extractions were able to retrieve the **sms.db** database, which contains the primary records of text messages. However, the FFS extraction provided additional access to the **sms.db-shm** (shared memory) and **sms.db-wal** (write-ahead log) files, which are crucial for recovering deleted or uncommitted messages. These auxiliary files store temporary and historical data that may not yet be written to the main database, allowing forensic analysts to uncover remnants of deleted messages, message fragments, and other valuable artifacts. This expanded access makes FFS a more comprehensive method for SMS data recovery compared to the AL extraction, which is limited to retrieving only the active database without these supplementary sources.

A FFS extraction was able to recover Snapchat and Facebook Messenger texts, while the Advanced Logical extraction could not, due to the level of access each method provides. Advanced logical extraction is limited to user-accessible data, relying on iOS APIs to pull active messages stored within app databases, which often exclude deleted or hidden content. In contrast, FFS extraction grants access to the entire **/private/var** directory, including application sandboxes where messaging apps store their data.

What are the differences in extractions between an FFS and an Advanced Logical?

Figure 8



DETAILS

ARTIFACT INFORMATION

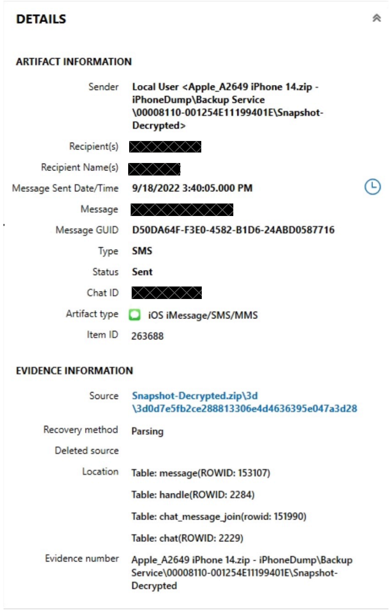
Sender	**Local User <00008110-001254E11199401E_files_full.zip>**
Message Sent Date/Time	**1/17/2025 3:49:16.000 AM**
Message	[redacted]
Message GUID	**15E00DDA-450B-4472-A0C7-39879C3BEDEC**
Type	**iMessage**
Status	**Sent**
Message Delivered Date/Time	**1/17/2025 3:49:16.000 AM**
Artifact type	🟩 iOS iMessage/SMS/MMS
Item ID	65599

EVIDENCE INFORMATION

Source	00008110-001254E11199401E_files_full.zip \private\var\mobile\Library\SMS\sms.db-wal
Recovery method	Carving
Deleted source	
Location	File Offset 149718
Evidence number	00008110-001254E11199401E_files_full.zip

FFS SMS Artifact

Figure 9



DETAILS

ARTIFACT INFORMATION

Sender	**Local User <Apple_A2649 iPhone 14.zip - iPhoneDump\Backup Service \00008110-001254E11199401E\Snapshot-Decrypted>**
Recipient(s)	[redacted]
Recipient Name(s)	[redacted]
Message Sent Date/Time	**9/18/2022 3:40:05.000 PM**
Message	[redacted]
Message GUID	**D50DA64F-F3E0-4582-B1D6-24ABD0587716**
Type	**SMS**
Status	**Sent**
Chat ID	[redacted]
Artifact type	🟩 iOS iMessage/SMS/MMS
Item ID	263688

EVIDENCE INFORMATION

Source	Snapshot-Decrypted.zip\3d \3d0d7e5fb2ce288813306e4d4636395e047a3d28
Recovery method	Parsing
Deleted source	
Location	Table: message(ROWID: 153107) Table: handle(ROWID: 2284) Table: chat_message_join(rowid: 151990) Table: chat(ROWID: 2229)
Evidence number	Apple_A2649 iPhone 14.zip - iPhoneDump\Backup Service\00008110-001254E11199401E\Snapshot-Decrypted

AL SMS Artifact

Observations

Figures 8 and 9 show that the data from the FFS used a carving method[6], while the advanced logical used parsing[7]. A FFS extraction uses carving because it provides access to raw storage, allowing forensic tools to search for deleted, fragmented, or unallocated data that is no longer referenced by the file system. This method is essential for recovering remnants of messages, logs, and other artifacts that standard database queries cannot access. In contrast, an Advanced Logical extraction relies on parsing, which involves reading structured data from active databases and system files using predefined schemas. Since advanced logical extraction works within the constraints of iOS APIs, it can only retrieve existing, logically stored data, making it less effective for recovering deleted or hidden information compared to FFS.

[6] Carving is the process of recovering deleted or fragmented data by searching for specific patterns in raw storage, without relying on file system metadata. It is commonly used when files are deleted or corrupted, and traditional access methods fail. Carving techniques identify data signatures, such as file headers and footers, to reconstruct lost information.

[7] Parsing is the process of extracting and interpreting structured data from known file formats, databases, or system records based on predefined rules. It involves reading files in their intended format, following the logical structure, and presenting the data in a readable and organized manner.

Conclusion

In digital forensics, understanding the differences between FFS and Advanced Logical extractions is crucial for investigators. An FFS extraction provides access to the **/private/var** directory and other critical system directories, allowing for a deeper forensic analysis of logs, system caches, application data, and even deleted files. This level of access enables investigators to retrieve hidden or encrypted data that may be vital for uncovering evidence in criminal cases. Additionally, FFS extractions use decryption, which allows more data to be recovered and provides a more detailed file structure for analysis.

In contrast, an Advanced Logical extraction primarily focuses on user-accessible data, such as messages, call logs, photos, and app data. However, because this method processes encrypted data without fully decrypting the file system, it has limitations in retrieving deleted or hidden information. While this method still provides valuable insights into user activity, communication patterns, and app usage, it does not offer the same depth of analysis as FFS. By understanding these differences, forensic investigators can determine the most appropriate extraction method based on case requirements. Accessing system-level directories through FFS can reveal deeper insights into device usage, tampering attempts, or even evidence of deleted data, while Advanced Logical extraction ensures quick access to relevant user data. This knowledge helps forensic professionals choose the best approach for gathering evidence while maintaining data integrity and admissibility in legal proceedings.

This research has been reviewed and improved with feedback from George Rodriguez, whose insights have strengthened the accuracy and depth of this analysis.[8]

[8] This research is subject to change and may vary depending on data types and software updates. The effectiveness and capabilities of data recovery tools can be influenced by the nature of the data being analyzed and any modifications or improvements made to the software over time.