t-defence

Advanced Data Extraction in iOS Forensics:

File Protection and Device Security

#TDefenceBusiness

DFIR

Il Gruppo DFIR di T-Defence è una Threat Response Unit specializzata in Digital Forensics & Incident Response che supporta imprese e pubbliche amministrazioni nella gestione di incidenti di sicurezza e nella produzione di evidenze digitali con valore probatorio.

L'attività del Gruppo integra competenze multidisciplinari e si articola in quattro aree principali:

- **Incident Response:** capacità di intervento rapido per contenere, eradicare e mitigare incidenti, riducendo l'impatto operativo;
- **Consulenze Tecniche d'Ufficio (CTU) e di Parte (CTP):** perizie informatiche conformi alle best practice di catena di custodia, a supporto del contesto giudiziario;
- **Forensic Readiness:** predisposizione preventiva di processi, tecnologie e standard per garantire che i dati raccolti siano accurati, integri e attestabili;
- **Ricerca e innovazione:** sperimentazione di tecnologie avanzate (eBPF, kernel telemetry, AI per anomaly detection, container forensics) per anticipare le minacce e sviluppare strumenti di nuova generazione.

In qualità di Threat Response Unit, il Gruppo DFIR non si limita alla fase di indagine post-evento, ma affianca i Security Operations Center (SOC) e le organizzazioni nella detection proattiva, nel threat hunting e nella gestione di crisi cyber.

La missione del Gruppo è elevare i livelli di sicurezza e resilienza delle infrastrutture critiche e dei sistemi informativi, coniugando rigore scientifico, innovazione tecnologica e capacità operativa a supporto della difesa digitale e del contesto giudiziario.

Sommario

Executive Summary

In the field of digital forensics, iOS devices represent both a challenge and a frontier for investigators. Apple's iOS ecosystem is renowned for its strong security measures, designed to protect user data against unauthorized access. While this robust security is beneficial for end-users, it introduces significant complexity for forensic professionals who need to legally extract data. Understanding the architecture of iOS, the methods for data acquisition, and particularly the intricacies of file protection classes is essential for any investigator seeking to analyze iPhones and iPads effectively.

Autori:

- Dott.ssa Gaia Calamari – Security Consultant Digital Forensics and Incident Management

iOS Data Storage and File Protection

One of the defining features of iOS is its approach to data protection, which integrates encryption at multiple levels. Every iOS device is equipped with a dedicated hardware encryption module, known as the **Secure Enclave**, which manages cryptographic keys and enforces access control policies. The Secure Enclave is responsible for protecting sensitive information, such as passwords stored in the Keychain, biometric data, and cryptographic keys used for encrypting files on the device.

Files in iOS are not simply stored in plaintext; they are assigned **data protection classes**, which determine when and how the files can be accessed. These classes leverage the device's passcode, hardware keys, and system state to enforce security. There are several primary classes, each with distinct forensic implications. For example, some files are accessible only when the device is unlocked, while others remain available after the first unlock following a reboot. This distinction can critically impact the ability of an investigator to extract evidence depending on the device's state at the time of acquisition.

Understanding these protection classes is crucial.

- **Complete Protection (NSFileProtectionComplete)**
  Files in this class are encrypted such that they are only accessible when the device is unlocked. If the device is powered off or locked, these files cannot be decrypted, even if an investigator obtains a physical image of the storage. This protection class is typically applied to highly sensitive data, such as Keychain items containing passwords or banking credentials.

- **Protected Until First Unlock (NSFileProtectionCompleteUntilFirstUserAuthentication)**
  Files in this class remain encrypted until the user unlocks the device at least once after a reboot. After this initial unlock, the files remain accessible in memory, even if the device is subsequently locked. This class is often used for less sensitive application data, such as caches or logs, and has practical implications for forensic timing: accessing a device immediately after a reboot may limit the data retrievable without the passcode.

- **Protected Unless Open (NSFileProtectionCompleteUnlessOpen)**
  This class allows files that were open before the device was locked to remain accessible in memory. For forensic investigations, this means that if an app is running or a file was accessed at the moment of the acquisition, its contents may still be retrievable, even though the device is now locked.

- **No Protection (NSFileProtectionNone)**
  Files with no protection are stored without encryption tied to the device lock state. These files are always accessible to the operating system and forensic tools, regardless of device state. They are generally used for non-sensitive content or temporary data.

Apple may also use hybrid or extended classes in certain system directories or for specific applications, such as media caches or system logs, which combine aspects of the above classes. These classes ensure that highly sensitive data is effectively isolated from casual or unauthorized access, while still enabling operational functionality for the device and apps.

Acquisition Methods

The existence of multiple protection classes in iOS means that investigators must carefully consider both **device state and acquisition method**. For example, a logical extraction performed while the device is unlocked may yield files from all classes except those that require "complete protection" and have never been unlocked. Conversely, a physical acquisition may capture encrypted data from all classes, but decryption will depend on whether the device's passcode or cryptographic keys are available.

Moreover, the "first unlock" timing creates a window where certain files become accessible, emphasizing the need for rapid, well-planned seizure procedures. Investigators must also contend with the impact of device updates and security patches, which can change encryption implementations or the behavior of file protection classes, making it critical to stay informed on iOS developments.

Forensic acquisition in iOS can be categorized into logical, file system, and physical acquisition, each offering different levels of data completeness and complexity.

- Logical acquisition is the least intrusive approach and extracts accessible data, including contacts, messages, call logs, and application files. While it avoids modifying the device and is safer legally, it may not capture encrypted files that fall under stricter protection classes or deleted artifacts.
- File system acquisition seeks to obtain a more comprehensive snapshot, including hidden files, caches, and system logs. This often requires elevated privileges on the device, which may involve exploiting vulnerabilities or using specialized tools.

- Physical acquisition, on the other hand, captures the device's memory in its entirety, including system partitions and deleted files. It is technically the most challenging and legally sensitive method, especially given the constant security updates Apple implements to close potential forensic exploits.

The interplay between acquisition method and file protection class is particularly significant. A logical extraction might never reveal files that are protected until device unlocks, whereas a physical image could, under the right conditions, allow access to even deleted files. Effective iOS forensics thus requires a nuanced approach that considers device state, file protection class, acquisition method, and legal constraints. Examiners must maintain strict documentation and evidence integrity throughout the process, ensuring that recovered data is both complete and admissible.

Challenges

The security model of iOS is not simply based on full-disk encryption, but on a multi-layered key hierarchy that combines hardware-bound secrets, user-derived material, and per-file encryption keys. Each file is encrypted with a unique per-file key (File Key), which is in turn wrapped by a class key, and ultimately protected by the hardware UID key fused into the SoC. This UID key is not accessible by software and is only usable internally by the AES engine, which means that even a full physical dump of NAND memory is insufficient for decryption without proper key material.

At the center of this architecture is the distinction between BFU (Before First Unlock) and AFU (After First Unlock) states. This distinction is critical in forensic scenarios. When a device is in BFU state, only a minimal subset of class keys is loaded into memory, meaning that most user data, especially files protected with NSFileProtectionComplete (remains cryptographically inaccessible). After the first successful unlock, several class keys are decrypted and retained in memory, transitioning the device into AFU state (After First Unlock). In this condition, a significantly larger portion of the file system becomes accessible, even if the device is subsequently locked again.

This behavior has direct forensic implications. If a device is seized while in BFU state, even advanced acquisition techniques such as file system extraction or partial physical imaging will yield largely encrypted data blobs. Conversely, if the device is in AFU state, forensic tools can leverage the presence of class keys in memory to access protected files without needing to brute-force the passcode.

Key Derivation and Passcode Entanglement

The user passcode is not used directly as an encryption key. Instead, it is processed through a PBKDF2 key derivation function, combined with the hardware UID key, and enforced by the **Secure Enclave Processor (SEP)**. The SEP introduces **rate limiting** and enforces escalating delays after incorrect passcode attempts, making brute-force attacks computationally impractical on-device.
From a forensic standpoint, this means that any attempt to derive class keys from the passcode must either:

- Occur within the SEP constraints (which is infeasible for strong passcodes), or
- Exploit vulnerabilities that allow bypassing SEP enforcement or extracting key material from memory.

This is why many forensic techniques rely on maintaining the device in AFU state rather than attempting passcode recovery.

Class Keys and Memory Residency

Each file protection class corresponds to a **class key**, which is decrypted and loaded into memory depending on device state. For example:

- The class key for NSFileProtectionComplete is only available while the device is unlocked and is purged from memory immediately upon lock.
- The class key for NSFileProtectionCompleteUntilFirstUserAuthentication is decrypted after first unlock and remains in memory even when the device is locked again.
- Less restrictive classes may have keys that are always derivable without user interaction.

This creates a nuanced situation where **two files on the same device may have entirely different accessibility profiles**, depending solely on their assigned protection class. During forensic acquisition, tools that operate in AFU state can access files whose class keys remain resident but will fail to decrypt those tied to keys that have already been evicted.

Secure Enclave and Anti-Forensic Properties

The Secure Enclave plays a critical role not only in key management but also in enforcing anti-forensic protections. It maintains its own secure memory, separate from the main application processor, and handles operations such as passcode verification and key unwrapping. Importantly, the SEP firmware is signed and isolated, making it resistant to tampering.

Additionally, the SEP enforces:

- Passcode retry counters;
- Data wipe policies (optional, after 10 failed attempts);
- Cryptographic key invalidation upon certain state transitions.

These features significantly limit traditional forensic approaches. Even if an investigator obtains a full physical image, without access to the SEP-mediated key unwrapping process, the encrypted data remains unusable.

Implications for Forensic Acquisition Techniques

From a practical perspective, the technical constraints described above directly influence acquisition strategy.

In AFU scenarios, forensic tools can perform **file system extraction** by leveraging already-unlocked class keys. This is often the most fruitful condition for investigators, as it allows access to application data, databases, and system artifacts without needing to defeat encryption.

In BFU scenarios, however, the situation changes dramatically. Without class keys in memory, even a full file system extraction results in encrypted files. In such cases, investigators must rely on:

- exploits targeting bootrom or early boot stages (e.g., checkm8-based approaches on vulnerable and old devices);
- recovery of partial artifacts stored in less protected classes;
- external sources such as iCloud backups or synchronized data.

Sandboxing and Data Fragmentation

Beyond encryption, iOS enforces strict **application sandboxing**, where each app operates within its own container directory. From a forensic perspective, this results in data fragmentation across multiple isolated domains, each with its own access controls and protection classes.

Even with full file system access, correlating data across applications requires manual reconstruction. For example, a messaging artifact might involve:

- a SQLite database in one container;
- media files in another directory;
- metadata stored in system-level caches.

Without proper decryption of each component (potentially governed by different protection classes), reconstruction of user activity may be incomplete or misleading.

Integrity and Forensic Soundness at Low Level

At a technical level, ensuring data integrity goes beyond simple hashing of extracted files. Investigators must consider:

- **APFS snapshots**, which may provide a consistent point-in-time view of the file system;
- **copy-on-write behavior**, which affects how deleted or modified data is represented;
- **hash verification at multiple stages**, including raw image, file system layer, and parsed artifacts.

Moreover, acquisition tools must avoid triggering state changes that could alter key availability, such as forcing a reboot (which would revert the device to BFU state and purge critical class keys from memory).

Conclusion

Forensic analysis of iOS devices has evolved into a highly specialized discipline that requires a deep understanding of both operating system internals and modern cryptographic design. Apple's security architecture is not merely a collection of protective mechanisms, but a tightly integrated system in which hardware-backed encryption, key management, and access control policies operate together to minimize the attack surface. In this context, the concept of file protection classes is not an isolated feature, but a fundamental component of the overall data protection model.

A key takeaway for forensic practitioners is that **data accessibility is inherently state-dependent**. The distinction between BFU (Before First Unlock) and AFU (After First Unlock) conditions is often more decisive than the acquisition technique itself. Even the most advanced extraction methods can be rendered ineffective if the relevant class keys are not available in memory. Conversely, a well-timed acquisition on a device in AFU state may allow access to a substantial portion of user data without requiring decryption attacks or passcode recovery.

Equally important is the understanding of the **iOS key hierarchy**, where per-file encryption keys are protected through class keys and ultimately bound to the hardware UID and mediated by the Secure Enclave. This layered approach ensures that raw data, even if physically acquired, remains cryptographically protected unless the correct chain of trust is satisfied. Forensic workflows must therefore shift from a purely acquisition-focused mindset to one that emphasizes **key availability, device state preservation, and memory-resident artifacts**.

The role of the Secure Enclave further reinforces this paradigm. By enforcing passcode policies, rate limiting, and secure key handling, it effectively eliminates traditional brute-force approaches and introduces strong anti-forensic characteristics. As a result, modern iOS forensics often depend on maintaining favorable conditions rather than attempting to break encryption directly.

In addition, the interaction between encryption and sandboxing introduces a second layer of complexity. Even when data is successfully decrypted, it is distributed across isolated application containers, often requiring correlation between multiple artifacts, formats, and protection classes.

This fragmentation demands not only technical expertise but also analytical rigor in reconstructing user activity in a forensically sound manner.

Ultimately, successful forensic analysis of iOS devices depends on the ability to integrate several dimensions: a precise understanding of file protection classes, awareness of the cryptographic key lifecycle, careful selection of acquisition techniques, and strict adherence to evidence handling procedures. Investigators must be capable of adapting to different scenarios, while preserving the integrity and admissibility of the evidence.

While iOS presents significant technical barriers, it does not make forensic analysis impossible. Instead, it shifts the challenge from data extraction to **data accessibility under constrained cryptographic conditions**. A methodical, informed, and technically grounded approach enables investigators to navigate these constraints and extract meaningful, reliable evidence, even within one of the most secure consumer operating systems currently available.

References

https://blog.digital-forensics.it/2025/09/exploring-data-extraction-from-ios.html

t-defence

Next | Donexit | Foramil | Innodesi

Via Giacomo Peroni, 452 – 00131 Roma
tel. 06.45752720 – info@defencetech.it
www.tinextadefence.it

#TDefenceBusiness