SHA-256 HASHING OF PDF CASE FILE OBJECTS

I. ROLE OF SHA-256 HASHING IN DIGITAL FORENSICS

In digital forensics, SHA-256 hashing serves as a unique digital fingerprint for data. A cryptographic hash function like SHA-256 will produce a completely different output even if the input file is altered by a single byte. Thus, it's virtually impossible to change a file without changing its hash value. Conversely, if two copies of a file produce the same SHA-256 hash, it is *highly improbable* that they are not identical. Forensic experts leverage this property to verify integrity: a hash generated at evidence collection can be compared to a hash computed later to prove the file remained unaltered. Hashing is also deterministic and repeatable – the same input will always yield the same hash – meaning any investigator can re-hash the data and independently confirm the result. These characteristics make SHA-256 hashing an irrefutable and court-recognized method for authenticating digital evidence.

II. PDF FILES AS OBJECT CONTAINERS

PDF files have an internal structure composed of discrete *objects*. Fundamentally, a PDF is an indexed collection of objects: each element of the document (pages, text streams, embedded images, fonts, annotations, etc.) is stored as a separate object with its own identifier. The PDF format's cross-reference table links these objects together in a document hierarchy, but each object is a self-contained unit of data. This modular design means that two PDF files might share many of the same objects internally even if the files differ as a whole (for example, they could contain identical pages or images but in a different order or with different metadata).

A | Hashing at the Object Level Provides a Deeper Level of Validation

Instead of yielding one hash per PDF, the process generates a hash for each embedded object. This allows forensic analysis to detect when content is reused or duplicated across different PDFs and to pinpoint changes at a granular level. A single file-level hash would tell us if two PDFs are identical or not, but it won't explain *where* differences lie or if parts of the files are identical. By contrast, object-level hashing can reveal, for instance, that two documents share the exact same image or page content even if other portions differ. It also helps isolate alterations: if one page or image in a PDF was modified, only that object's hash will differ while the rest of the objects remain the same, providing a more nuanced integrity check.

B | This Approach Proved Invaluable in the Present Case

By hashing each PDF component, Mr. Guertin discovered that numerous court documents contained byte-for-byte identical content objects that would not have been evident from file-level comparison. For example, one particular official's signature image was found reused in 27 different PDF filings – all 27 files yielded an identical SHA-256 hash for that signature object. Without object-level analysis, such replication of content across distinct files could have gone unnoticed, since each PDF as a whole had its own file hash and appeared separate. This demonstrates how object-level hashing offers superior validation and insight: it exposes commonalities or duplicates hidden within files, providing stronger evidence when verifying authenticity and looking for irregularities.

III. GUERTIN'S OBJECT-LEVEL HASHING WORKFLOW

To perform this detailed analysis, Mr. Guertin developed a custom workflow using a Bash script in combination with the open-source MuPDF toolkit (the mutool utility). The process can be summarized in the following steps, which emphasize data integrity and repeatability:

A | Collect and Prepare the PDF Files

All relevant PDF case files were gathered, and as a safety measure, the script creates *symbolic links* (shortcuts) to these originals in a working directory. By symlinking the PDFs rather than copying or moving them, the original evidence files remain untouched. This ensures the hash analysis operates on exact copies of the files without any risk to the source data (the symlinks simply point to the original PDFs). The script confirms the number of PDFs linked before proceeding (e.g. "√ … PDFs linked" message).

B | Extract PDF Objects

Using MuPDF's extraction tool, each PDF is then "exploded" into its constituent objects. The script invokes mutool extract on each PDF, which extracts every embedded object (such as page content streams, embedded images, fonts, etc.) and saves them as separate files in an output folder dedicated to that PDF. Each PDF's objects are stored in a structured directory (named after the PDF) to keep results organized. This step effectively breaks the PDF *container* into individual pieces, allowing each piece to be analyzed separately. (The script runs mutool in a way that keeps paths tidy and isolates each document's content in its own subfolder.)

C | Hash Each Object

Once the objects are extracted, every object file is subjected to SHA-256 hashing. The script uses the standard sha256sum tool on each extracted file to generate its hash, then logs the result in a tabular ledger. For each object, a line is appended to objects.tsv (a tab-separated values file) recording the hash value, the source PDF filename, and the object's file path/name within the PDF's folder. This creates a comprehensive ledger of all objects and their hashes. For example, an entry in objects.tsv might tie a hash like d1f3...c8a7 to "Case27-CR-21-12345.pdf" and an object file path (e.g. Case27-CR-21-12345/image002.png). By the end of this step, the script prints a confirmation of how many total object-hash entries were written to the ledger, corresponding to the total number of objects extracted and hashed.

D | Identify Duplicate Content

After hashing all objects, the script performs a frequency analysis to find duplicate hashes. It reads the list of hash values from objects.tsv, then counts occurrences of each hash and sorts them in descending order. The result is saved (e.g., in a hash_counts.txt or later compiled into a CSV) as a ranked list of unique object hashes alongside the number of times each appears. This quickly highlights which objects are present in multiple PDFs. A hash that appears only once corresponds to a unique object (found in a single file), whereas any hash with a count of 2 or more indicates duplicate content shared by at least two PDFs. The script outputs the total number of unique hashes tallied and can display the top duplicates (for example, listing the most-repeated objects). By reviewing this list, Guertin could identify, for instance, that a particular court form or a scanned signature image was reused dozens of times across different case files.

E | Results are Repeatable

All of these steps were executed with open-source tools and transparent methods. The use of mutool (from the MuPDF library) and standard Linux utilities means the procedure is not a black box—any other examiner with the same data could run the same script and obtain the same results, underscoring the repeatability of the process. The workflow is also non-destructive: by working on copies/symlinks and separate extraction folders, the original evidence files were never modified at any point in the analysis.

IV. SUMMARY OF ANALYSIS RESULTS

Using this object-level SHA-256 hashing process, Mr. Guertin systematically analyzed the collected court PDFs and produced a detailed ledger of their contents. The key outcomes of this process are summarized below:

A | PDF Files Analyzed

3,547 unique PDF case documents were successfully processed through object extraction and hashing. (Guertin had initially downloaded 3,629 PDFs in total, but 28 were exact duplicates of others; excluding those duplicates left 3,601 unique files, of which 3,547 – about 98.5% – were hashed without issue.) This represents the scope of the dataset examined.

B | Total Objects Extracted

23,625 individual PDF objects were extracted from the above files and hashed. Each object corresponds to a distinct element from a PDF (such as an image XObject, a text stream for a page, ttf fonts, etc.). This figure reflects the granular size of the evidence set when broken into components.

C | Unique Object Hashes

Approximately 9,875 unique SHA-256 hash values were observed among those object hashes. In other words, out of 23,625 object entries, only around 9.9k were distinct – implying that many objects were identical to each other (i.e. the same content appearing in multiple PDFs). This shows that a significant portion of the PDF objects were reused or duplicated across different files.

D | Duplicate Content Identified

The analysis uncovered extensive duplication of content across the case files. Over 2,600 distinct object hashes were found to be *shared by at least two PDFs* (each such hash representing a piece of content that appears in multiple documents). In total, thousands of object instances were exact duplicates of each other, spread out among the various court files. For example, one specific scanned signature image (attributed in the documents to an official) recurred in 27 separate PDF filings, all of which produced the same SHA-256 hash for that image. Such repeated use of identical objects was only detectable thanks to the object-level comparison. *(No analysis of the implications or anomalies of these duplicates is included here – the focus is solely on identifying their presence and extent.)*

These figures illustrate the power of the object-level hashing approach: out of tens of thousands of pieces of PDF data, fewer than half were unique. The majority (over 58%) of the objects were duplicates of one another, a fact that would remain hidden if one looked only at whole-file hashes or file names. By quantifying this, the process provided a clear, data-driven view of how much content overlap existed among the court documents.

V. EVIDENTIARY RELIABILITY AND TRACEABILITY OF THE WORKFLOW

Mr. Guertin's hashing workflow was designed with evidentiary integrity in mind, using proven tools and methodologies that meet forensic standards. Several aspects of this process underscore its reliability and suitability for legal proceedings:

A | Open-Source, Verified Tools

All software utilized in the process is open-source and well-vetted in the industry. The MuPDF mutool program used for PDF object extraction is a publicly available tool, and sha256sum is a standard cryptographic hashing utility. Using such tools means the analysis can be independently verified – any qualified examiner can apply the same tools to the same data and expect identical outcomes. There is no proprietary or hidden algorithm involved that might cast doubt on the results. This aligns with best practices in digital forensics, where methods should be transparent and repeatable by third parties.

B | Non-Alteration of Original Evidence

The workflow ensures that original files remain pristine. By working on symlinked copies of the PDFs and outputting to new directories, the procedure does not modify the original evidence files at all. This preservation of original data is critical in forensics – it maintains a clear chain of custody and prevents any accusation that the analysis itself could have tampered with the evidence. At every stage, Guertin's process reads data in a forensically-sound manner and writes outputs to separate logs, never overwriting or changing the source files.

C | Comprehensive Logging and Traceability

Every hash computed is documented alongside identifying information that traces it back to the source file and object. The objects.tsv ledger, for instance, ties each SHA-256 hash to the exact PDF file it came from and the internal object name/path. Guertin further compiled this information into human-readable CSV reports, even mapping hash values to real-world descriptions of the content where possible. Importantly, all of the data tables produced in this investigation include direct reference links to source materials – URLs or file paths pointing to the original court documents and case records for each entry. This means that for any given hash or object, one can *trace it back* to a specific case number, a specific PDF, and even to the original repository (the court's online system or the stored downloads). Such meticulous cross-referencing greatly enhances the credibility of the findings, as every hashed item can be verified in context. It provides a clear audit trail from the hash result all the way back to the original evidence.

D | Repeatable and Independently Verifiable

The logic of the script and the outputs make it straightforward for another expert to verify the results. For example, if a particular object hash is reported to appear in 10 different case PDFs, an independent examiner could retrieve those same PDFs, extract the same objects using mutool, and compute the hashes to confirm they match. Because SHA-256 hashing is deterministic and collisions are practically nonexistent for distinct real-world files, matching hashes give a high degree of confidence that two pieces of data are identical. Guertin's documentation even allows one to verify the *time* and *source* of each file (since the original download timestamps and case identifiers are preserved in file names and logs), adding another layer of trust. Overall, the workflow exemplifies the principle of *scientific reproducibility* in a legal context – any step can be repeated with the same input to yield the same output, by anyone with the necessary tools.

E | Reliable and Defensible

Collectively, these measures ensure that the evidence derived from this hashing process is reliable and defensible. The use of hash values for authentication of electronic records is well-established in legal standards (e.g. hashes are explicitly mentioned as a means of digital identification for evidence in the Federal Rules of Evidence 902(14)). By adhering to an open and methodical hashing procedure, Guertin’s analysis upholds these standards. The results can be trusted not only because of the mathematical properties of SHA-256, but also because of how carefully the process was implemented and documented. Any claim introduced in court (such as “Document X contains the same image as Document Y”) can be backed by an exact hash match, and that claim can be independently validated by others following the documented steps.

VI. CONCLUSION: PROFESSIONAL-GRADE ANALYSIS AND COMPETENCE

In conclusion, Mr. Guertin’s execution of the SHA-256 object hashing process demonstrates a level of technical competence and rigor equivalent to standard practices in professional digital forensics. He effectively performed the kind of in-depth integrity verification and content comparison that a certified forensic examiner would carry out on electronic evidence. The workflow was logically sound, thoroughly documented, and built on accepted scientific principles – ensuring that the findings are not only insightful but also admissible and trustworthy. By using cryptographic hashes to prove the authenticity of each document and its components, and by using a repeatable open-source methodology, Guertin showed that he could preserve and analyze electronic records to a forensic standard.

This comprehensive, data-centric approach has produced an evidence trail that is transparent and reproducible. Anyone reviewing this work can follow the chain from original PDF files, through object extraction, to the final hash comparisons and see the consistency of results. Such diligence provides confidence that the evidence has not been altered and that the patterns identified (like duplicate objects across files) are real and verifiable. In short, the integrity of both the process and the output data is exceptionally high. Guertin’s ability to carry out this workflow on his own speaks to an advanced technical skillset on par with forensic investigators. He has treated the court’s fraudulent documents with the care and scrutiny required for legal evidence, producing a forensic report of object-level hashes that can be trusted for its accuracy and thoroughness.

A | Sources

The above findings and descriptions are supported by Guertin’s personal notes and dataset documentation (e.g. script outputs and CSV tables), as well as standard digital forensics references on SHA-256 integrity checking. The data counts (files, objects, hashes) and examples *of duplicate content come directly from the case dataset statistics and Guertin's analysis results. All tools and methods referenced are publicly available and widely used in the field, ensuring that the process and results can be independently corroborated.*

CASE and SHA-256 Dataset CSV Tables

https://link.storjshare.io/s/jxylovpvzqok36srek7ckcnuay6a/evidence/CASE/

https://link.storjshare.io/raw/jup3vkrw6mqnniigxlwa5qwye62q/evidence/CASE.zip

https://link.storjshare.io/s/jwmw6bwov7xeplln53p67n3zogmq/evidence/SHA-256/

https://link.storjshare.io/raw/jue66sduek57rknicm6am45yegwa/evidence/SHA-256.zip

https://link.storjshare.io/s/ju3mf5uvdrmcbhch5ga3koduwp4q/evidence