Designing an Ideal Filesystem for Digital Forensic Investigations

Nandankumar Desai
Information Networking Institute
Carnegie Mellon University
Pittsburgh, PA, United States
nandankd@andrew.cmu.edu

***Abstract*—Digital forensic investigations rely heavily on the authenticity and reliability of digital evidence, but recent cases of fabricated evidence have raised questions about the effectiveness of current forensic tools and procedures. This research paper introduces IdealFS, a file system designed specifically for digital forensic investigations. IdealFS provides strong attribution and non-repudiation for each file and directory in the file system, and ensures forensic soundness by requiring cryptographic signature verification for all file system operations. The paper also emphasizes the importance of formally verified software and digital forensic tools in ensuring the reliability of evidence in forensic investigations. By addressing the issues of authenticity and reliability of digital evidence, IdealFS offers a promising solution for enhancing the effectiveness of digital forensic investigations.**

***Index Terms*—Digital Forensics, File Systems, IdealFS, Forensic Investigation Tools, Court Evidence, Data Analysis, Evidence Collection, Legal Proceedings, Data Recovery, Disk Imaging.**

I. Introduction

The use of digital evidence in forensic investigations has become increasingly common in recent years, with hard disks and other digital storage devices often providing crucial insights into criminal activity. However, the reliability of such evidence has been called into question in several high-profile cases, highlighting the need for a more robust and trustworthy approach to digital forensics.

One such case is that of Stan Swamy [1], an 84-year-old Indian Catholic priest who was arrested and charged by the India's National Investigation Agency (NIA) in 2020 for his alleged role in the 2018 Bhima Koregaon violence [4]. The NIA claimed to have recovered several incriminating letters from Swamy's hard disk during a forensic analysis, which formed a key part of their charges against him. The letters were said to be written by Swamy to members of a banned left-wing militant organization, providing funding, leadership, and intellectual support for their activities.

However, subsequent investigations by independent digital forensics firms have revealed that the evidence was fabricated and planted by hackers who had infiltrated Swamy's device [3]. The same firm that investigated Swamy's hard drive, Arsenal Consulting, also examined the hard drive of another accused in the same case, Rona Wilson, and concluded that evidence on his hard disk was also planted [5] [6]. This revelation has raised questions about the reliability and authenticity of digital evidence in forensic investigations.

The problem of fabricating and planting evidence on hard disks is not new, but it has become more prevalent in recent years due to the increasing sophistication of cyberattacks and the availability of tools and techniques for covering one's tracks. Forensic investigators face several challenges when it comes to recovering and analyzing digital evidence, including issues of data integrity, tamper resistance, and anti-forensic techniques. In some cases, the use of anti-forensic techniques may make it impossible to recover digital evidence at all, leaving investigators with no leads or clues to follow.

To address these challenges, this paper proposes a hypothetical file system called IdealFS that incorporates metadata to enhance the reliability and authenticity of digital evidence. IdealFS is designed to be forensic-friendly, meaning that it prioritizes the needs of forensic investigators over other considerations such as performance or security. We argue that such a file system would help to address the challenges of data integrity, tamper resistance, and anti-forensic techniques, making it more difficult for malicious actors to fabricate or plant evidence on a hard disk.

The paper concludes by calling for a more rigorous and transparent approach to digital forensics, which prioritizes the protection of civil liberties and the rights of individuals. It is essential that forensic investigators have access to reliable and trustworthy tools and techniques for recovering and analyzing digital evidence, particularly in cases where individuals' freedom or reputation is at stake. The development of IdealFS represents an important step towards achieving this goal, but further research and development are needed to make it a practical reality.

II. IdealFS

A. Background

This paper proposes the development of IdealFS, a new file system that maintains forensically-relevant metadata for every file and directory in the file system. The aim of IdealFS is to create a file system that is specifically designed to aid forensic investigations by incorporating metadata in a way that makes it difficult to fabricate or tamper with evidence. To achieve this goal, IdealFS is based on the FAT16 file system, which has relatively simple and straightforward internal data structures that can be extended to include new metadata entries.

In designing IdealFS, we have created the specifications from scratch, reasoning about why certain metadata entries are necessary. In FAT16, there are several ”unused” places in the data structures, like the slack space of a cluster, unrelated Long File Name (LFN) entries in the Directory Entry, etc. These unused sections of the disk allow a bad actor to hide data while the file system still remains valid. One of the goals of IdealFS is to minimize these ”unused” sections and define what value should be stored in those places, if they happen to be unused. This allows for better consistency checks and enables tools to determine whether the file system is in a valid or invalid state.

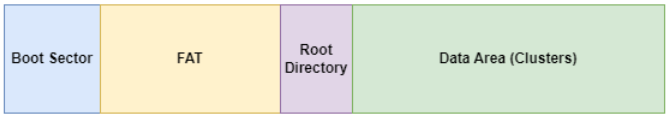

Fig. 1. Layout of IdealFS.

B. Boot Sector

One of the key features of IdealFS is the use of digital (cryptographic) signatures to establish the integrity of certain important data structures, such as the Boot Sector. The Boot Sector in IdealFS is similar to that of FAT16, but includes an additional entry in the byte range 62 to 93, as shown in Table 1. This entry is the digital signature that is obtained from the Trusted Platform Module (TPM) of the host computer by the OS or tool used to create IdealFS on a disk. The signature ensures the integrity of the data in the byte range 0 to 61, as shown in Table 1. The purpose of this signature is to confirm that the file system was created on a specific device. That’s why the TPM is used to obtain the signature. In any future forensic investigation, the analyst can prove whether the file system was created on the host computer or not.

C. File Allocation Table (FAT)

The Boot Sector occupies the first 512 bytes of the file system. The FAT structure starts immediately after the Boot Sector and is exactly the same as in FAT16. The purpose of the FAT structure is to store the Cluster Chain, again, the same as in FAT16. Each FAT entry of IdealFS is 16-bits.

Multiple copies (backup copies) of a FAT structure can also exist. The number of copies is given in the Boot Sector, and the size of the FAT structure will also be given in the Boot Sector (as shown in Table 1).

D. Root Directory and the Directory Entries

One important component of IdealFS is the Root Directory, which comes after the FAT region. The Root Directory is located at the top of the file and directory tree.

Each directory, including the Root Directory, in IdealFS is a special file that contains a list of 128-byte entries, unlike FAT16, which has 32-byte entries. The structure of each entry is defined in Table 2.

TABLE I
IDEALFS BOOT SECTOR

Byte Range	*Description*	*Essential**
0-2	Assembly instruction to jump to boot code	No
3-10	OEM Name in ASCII	No
11-12	Bytes per sector	Yes
13-13	Sectors per cluster	Yes
14-15	Value (0x1)	Yes
16-16	Number of FATs	Yes
17-18	Max. number of files in root directory	Yes
19-20	Value (0x0)	No
21-21	Media Type	No
22-23	Size (in sectors) of FAT	Yes
24-25	Sectors per track of storage device	No
26-27	Number of heads in storage device	No
28-31	Value (0x0)	No
32-35	Number of sectors in the file system	Yes
36-36	BIOS INT13h drive number	No
37-37	Value (0x0)	No
38-42	Volume serial number	No
43-53	Volume label in ASCII	No
54-61	”IdealFS” in ASCII	No
62-93	Signature of byte range 0-61	Yes
94-509	Value (0x0)	No
510-511	Signature value (0xBEEF)	No

*Essential file system data are those that are needed to save and retrieve files. [2]

The first 32 bytes of this structure are identical to those of the Directory Entries of FAT16. However, IdealFS has additional metadata after these first 32 bytes.

Byte range 32-35 (4 bytes) contains a unique ID, called the Unique Directory Entry ID, that must be unique amongst all directory entries on the file system. The IdealFS implementation software can determine how to maintain the uniqueness of this value.

Byte range 36-39 (4 bytes) contains the ID, called the Creator ID, of the binary file, which is running as a process in the operating system, that is creating/writing to this file. For example, if chrome.exe is downloading the test.pdf file, then the Unique Directory Entry ID of chrome.exe must be written in this field for the test.pdf file.

Byte range 40-71 (32 bytes) contains the MD5 hash of the binary file (called the Creator) that is creating/writing to this file. For example, if chrome.exe is downloading the test.pdf file, then the MD5 hash of the chrome.exe file must be written in this field for the test.pdf file.

The purpose of the Creator ID and the MD5 hash of the Creator is to establish the relationship between a file and the process that created it. The IdealFS implementation software needs to take care of the Creator ID, in case it changes when the Creator is moved to a different path in the file system.

Byte range 72-103 (32 bytes) contains the digital signature of the data in byte range 0-71 and the contents of the entire file. When chrome.exe is downloading test.pdf, the Directory Entry byte range 0-71 and the new/updated contents of test.pdf file are concatenated and submitted to the TPM. The digital signature obtained from the TPM is written in this field.

Byte range 105-127 is unused and should contain 0x0 value. This is mainly for alignment purposes so as to make each

Directory Entry 128 bytes in size.

Aside from changes to the Boot Sector and the Directory Entry structure, almost all other data structures in IdealFS are the same as those in FAT16. The slack space in a cluster must have a value of 0x0, and any unused spaces within a data structure must also have a value of 0x0 written.

TABLE II
IDEALFS DIRECTORY ENTRY

Byte Range	*Description*	*Essential*
0-0	First character of file name in ASCII (0xe5 or 0x00 for unallocated)	Yes
1-10	Characters 2 to 11 of filename in ASCII	Yes
11-11	File Attributes (same as FAT16)	Yes
12-12	Value (0x0)	No
13-13	Created time (tenths of second)	No
14-15	Created time (hours, minutes and seconds)	No
16-17	Created day	No
18-19	Accessed day	No
20-21	Value (0x0)	Yes
22-23	Written time (hours, minutes, seconds)	No
24-25	Written day	No
26-27	Lower 2 bytes of first cluster address	Yes
28-31	Size of file (0 for directories)	Yes
32-35	Unique ID for this entry	Yes
36-39	Entry ID of the creator	Yes
40-71	MD5 Hash of the creator	Yes
72-103	Signature of byte range 0-71	Yes
104-127	Value (0x0)	No

E. File Creation and Deletion

The process of creating a file in IdealFS involves a few extra steps, as described briefly in the previous section. Whenever a file is being created or written to, the TPM is used to obtain a digital signature. Additionally, the creator or writer of the file must have their Directory Entry ID and MD5 hash written in the appropriate fields, as described previously.

On the other hand, file deletion in IdealFS follows the same process as in FAT16. When a file is deleted from IdealFS, the first byte of the Directory Entry is replaced with 0xe5, and the related FAT entries are written with 0x0.

While a separate IdealFS specification document will need to be drafted, this paper provides a brief overview of the structure and processes used in IdealFS.

F. Goals

The primary goal of IdealFS is the attribution and non-repudiation of each file and directory in the file system. Attribution refers to the ability to identify the origin or creator of a file, while non-repudiation ensures that the creator cannot deny their involvement in creating or modifying a file.

Attribution is achieved through the use of the TPM, which provides a digital signature for each file and directory created or written to. The digital signature ensures that the identity of the creator or writer of the file or directory. Non-repudiation is achieved by establishing a link between a file and its Creator using the Creator ID and MD5, as described in the previous section.

The use of the TPM in IdealFS helps to achieve these goals by providing a secure and tamper-evident way of verifying the digital signature of each file and directory in the file system. In this way, IdealFS ensures that any evidence obtained from the file system is reliable and can be used in court, provided that the cryptographic signature verification is successful.

However, in the case of a cryptographic signature verification failure, IdealFS considers the entire file system on that disk as invalid. This is because any tampering with the digital signatures can compromise the forensic soundness of the file system, and any evidence drawn from it would be considered invalid in a court of law. Therefore, it is crucial to ensure that the cryptographic signature verification process is robust and reliable, to maintain the integrity and forensic soundness of IdealFS.

G. Practical applications

The practical applications of IdealFS are vast and can benefit individuals and organizations alike. One primary use case is for individuals who may be targeted with false accusations. A file system like IdealFS can provide an indisputable record of their actions, which can be used as evidence in a court of law.

IdealFS can also be useful for servers, assuming that the TPM can match the speed of modern CPUs and concurrently sign requests at a high speed. In such a scenario, the metadata that IdealFS stores can be invaluable when performing incident response. In addition to the logs typically maintained at the application level, IdealFS can also establish a clear link between the files and the processes that created them, as well as the hardware on which they were written to the disk. This information can be crucial for forensic investigations and identifying the source of a security breach.

Furthermore, IdealFS can also be beneficial in compliance and regulatory environments. For example, certain industries such as finance or healthcare have strict regulations regarding data retention and data access. IdealFS can provide a tamper-evident trail of all file activity, which can be used to demonstrate compliance with these regulations.

Overall, IdealFS has many practical applications and can be a valuable tool for maintaining the integrity of digital information in various contexts.

H. Implementation

The concept of formally verified software is an important one when it comes to ensuring the reliability of IdealFS. It refers to software that has been rigorously and mathematically proven to be correct according to a set of formal specifications. This approach provides a level of confidence in the software that cannot be achieved through conventional testing methods alone.

For IdealFS, having a formally verified software implementation means that the software has been proven to handle all aspects of the file system correctly, including its unique features such as the TPM-based digital signature and metadata storage. By using a formally verified software, we can significantly reduce the risk of bugs and errors that could compromise the forensic soundness of the file system.

Therefore, in addition to a well-defined specification, the use of a formally verified software is essential to ensure the integrity and reliability of IdealFS.

III. Formal Verification of Digital Forensic Tools

In addition to having formally verified software that handles IdealFS, it is also important to have digital forensic tools that can extract valid evidence from IdealFS. These tools must be designed and tested to ensure that they do not modify or alter the data during the acquisition process [7]. Moreover, they should be able to analyze the file system in a forensically sound manner and provide accurate and reliable results.

Developing formally verified digital forensic tools can be a challenge, as it requires a significant amount of effort and resources. However, it is crucial to ensure the accuracy and reliability of digital evidence, especially in legal proceedings. In recent years, there have been several efforts to develop formally verified digital forensic tools [8].

Apart from the challenges of developing these tools, there are also practical issues such as the wide variety of file systems and the lack of official specifications for some proprietary file systems. Nevertheless, the development of formally verified digital forensic tools is a critical step in ensuring the validity and reliability of digital evidence, particularly when using file systems like IdealFS that rely heavily on digital signatures and cryptographic mechanisms.

IV. Conclusion

In conclusion, the case of Stan Swamy highlights the need for a robust and reliable file system that can withstand forensic scrutiny and prevent the possibility of evidence tampering or fabrication. IdealFS was designed with these considerations in mind and provides a solution to the challenges faced by digital forensic investigators. By incorporating features such as attribution and non-repudiation, forensic soundness, and support for verified digital forensic tools, IdealFS offers a level of security and reliability that is crucial for digital forensic investigations.

Moreover, the use of formally verified software and digital forensic tools can further enhance the reliability and trustworthiness of evidence obtained from IdealFS. While the development and maintenance of such tools may be time-consuming and expensive, their importance cannot be overstated in ensuring that digital forensic investigations are conducted with accuracy and integrity.

Overall, IdealFS presents a significant step forward in the field of digital forensics, addressing the challenges of evidence tampering and fabrication, and providing a reliable framework for forensic investigations. It is our hope that the adoption of IdealFS by law enforcement agencies and digital forensic investigators will lead to more reliable and trustworthy digital evidence, and ultimately, to a more just legal system.

References

[1] “Stan Swamy.” Wikipedia, https://en.wikipedia.org/wiki/Stan_Swamy. Accessed 5 May 2023.

[2] Carrier, Brian. File System Forensic Analysis. Pearson Education, Limited, 2005.

[3] Masih, Niha. “Stan Swamy’s computer was hacked, evidence planted in Bhima Koregaon case.” The Washington Post, 13 December 2022, https://www.washingtonpost.com/world/2022/12/13/stan-swamy-hacked-bhima-koregaon/. Accessed 5 May 2023.

[4] “NIA Files Chargesheet Against Activists Stan Swamy, Teltumbde, 6 Others in Bhima Koregaon case.” News18, 9 October 2020, https://www.news18.com/news/india/nia-files-chargesheet-against-activists-stan-swamy-anand-teltumbde-and-5-others-in-bhima-koregaon-case-2946645.html. Accessed 5 May 2023.

[5] Masih, Niha, and Joanna Slater. “They were accused of plotting to overthrow the Modi government. The evidence was planted, a new report says.” The Washington Post, 10 February 2021, https://www.washingtonpost.com/world/asia_pacific/india-bhima-koregaon-activists-jailed/2021/02/10/8087f172-61e0-11eb-a177-7765f29a9524_story.html. Accessed 5 May 2023.

[6] Greenberg, Andy. “Police Linked to Hacking Campaign to Frame Indian Activists.” WIRED, 16 June 2022, https://www.wired.com/story/modified-elephant-planted-evidence-hacking-police/. Accessed 5 May 2023.

[7] Carrier, Brian. “Open Source Digital Forensics Tools: The Legal Argument.” 2002.

[8] Guo, Yinghua & Slay, Jill & Beckett, Jason. (2009). Validation and verification of computer forensic software tools—Searching Function. Digital Investigation. 6. 10.1016/j.diin.2009.06.015.