Chapter 21

# DATA RECOVERY FUNCTION TESTING FOR DIGITAL FORENSIC TOOLS

Yinghua Guo and Jill Slay

**Abstract** Many digital forensic tools used by investigators were not originally designed for forensic applications. Even in the case of tools created with the forensic process in mind, there is the issue of assuring their reliability and dependability. Given the nature of investigations and the fact that the data collected and analyzed by the tools must be presented as evidence, it is important that digital forensic tools be validated and verified before they are deployed. This paper engages a systematic description of the digital forensic discipline that is obtained by mapping its fundamental functions. The function mapping is used to construct a detailed function-oriented validation and verification framework for digital forensic tools. This paper focuses on the data recovery function. The data recovery requirements are specified and a reference set is presented to test forensic tools that implement the data recovery function.

**Keywords:** Digital forensic tools, validation, verification, data recovery

## 1. Introduction

Digital forensics is the process of identifying, preserving, analyzing and presenting digital evidence in a manner that is acceptable in courtroom proceedings [5]. As identified in [1, 2], one of challenges in the discipline is to ensure that the digital evidence acquired and analyzed by investigative tools is forensically sound.

In our previous work [2], we proposed a function-oriented framework for digital forensic tool validation and verification. The framework identified fundamental functions involved in digital forensic investigations such as search, data recovery and forensic copying. A process called "function mapping" was used to further identify the details of each function (e.g., sub-categories and components). The results enable the speci-

K.-P. Chow, S. Shenoi (Eds.): Advances in Digital Forensics VI, IFIP AICT 337, pp. 297–311, 2010.

fication of the requirements of each function and help develop a reference set against which digital forensic tools may be tested.

Our previous work addressed the first task in creating a validation and verification framework, i.e., the "search" function. This paper attempts to address the second task – to complete the function mapping, requirements specification and reference set development of the "data recovery" function. The following sections review our function-oriented validation and verification framework, present the details of the data recovery function mapping, and describe a pilot reference set for testing the data recovery function.

## 2. Validation and Verification Framework

Our validation and verification framework [2] is function-oriented and incorporates detailed specifications that are absent in other work. The methodology begins with a systematic description of the digital forensic field using a formal model and function mapping. Digital forensic components and processes are defined in this model and fundamental functions in the investigative process such as searching, data preservation and file identification are specified (i.e., mapped). Having developed the model and function mapping, the validation and verification of a digital forensic tool is accomplished by specifying its requirements for each mapped function. Next, a reference set is developed comprising a test case (or scenario) corresponding to each function requirement. The reference set enables the forensic tool and/or its functions to be validated and verified independently.

This paper engages the CFSAP model [6] to describe the basic procedures involved in a digital forensic investigation: identification, preservation, analysis and presentation. In the context of validation and verification, identification and presentation are skill-based concepts. On the other hand, preservation and analysis are predominantly process-, function- and tool-driven concepts and are, therefore, subject to tool validation and verification.

Beckett and Slay [1] have dissected the processes of preservation and analysis into fundamental functions. Figure 1 presents a function categorization of validation and verification.

In this work, we attempt to complete the mapping of the functional categories of the digital forensics discipline at a level of abstraction that would serve the purposes of a specification for a software developer, technical trainer or educator; or for tool validation or verification. In particular, we detail the specification of function categories (e.g., searching, data preservation and file rendering) and their sub-categories. Our

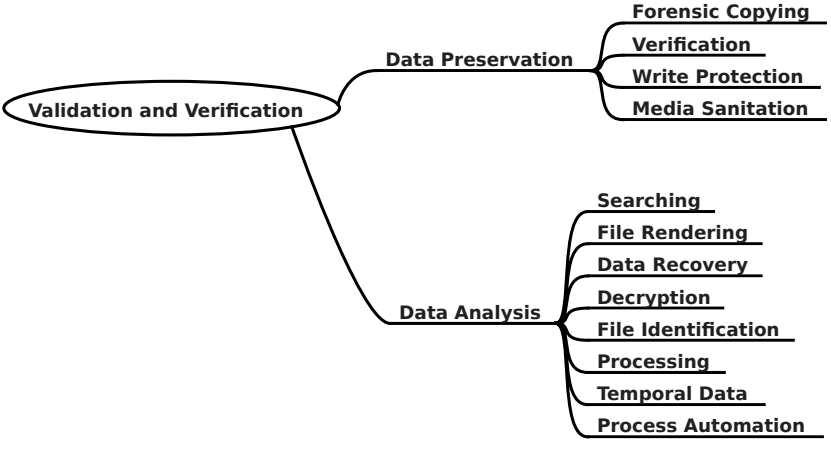


*Figure 1.* Validation and verification top-level mapping.

focus is on the data recovery function: mapping the function, specifying its requirements and developing the reference set to validate and verify tools that implement the data recovery function.

If the domain of digital forensic functions and the domain of expected results (i.e., requirements of each function) are known, in other words, the range and specification of the results are known, then the process of validating a tool can be as simple as providing a set of references with known results. When a tool is tested, a set of metrics can also be derived to determine the fundamental scientific measurements of accuracy and precision. In summary, if the discipline is mapped in terms of functions (and their specifications) and, for each function, the expected results are identified and mapped as a reference set, then any tool, regardless of its original design intention, can be validated against known elements. As claimed in [2], our function-oriented validation and verification regime has several distinctive features such as detachability, extensibility, tool version neutrality and transparency.

## 3. Data Recovery Function Mapping

Data recovery is generally regarded as the process of salvaging data partially or completely from damaged, failed, corrupted or inaccessible storage media. Recovery may be required due to physical damage to the storage device or logical damage to the file system that prevents it from being mounted by the host operating system.

A variety of failures can cause physical damage to storage media. CD-ROMs can have their metallic substrate or dye layer scratched off;

hard disks can suffer any of several mechanical failures; tapes can simply break. The logical damage to the data may take the form of corrupt or missing boot-related records (e.g., main boot record, disk partition table and directories) or the loss of file signatures (e.g., header and footer). Since our focus is on validating and verifying digital forensic tools in terms of the data recovery function, the consideration of physical damage recovery techniques is outside the scope of this paper and is considered to be complementary to logical damage recovery techniques. Consequently, in the rest of this paper, data recovery refers to logical damage recovery unless otherwise stated.

Data recovery in the context of digital forensics has its own peculiarities and differs from traditional data recovery in the computer science discipline. First, data recovery in the digital forensic context is a process by which digital evidence is recovered for use in court. Therefore, it should be conducted by certified investigators, conform to standard operating procedures, utilize tools that are validated and verified by the appropriate authorities, and be supervised and documented. Traditional data recovery does not have these requirements because its goal is to recover as much data as possible without concern for its forensic soundness. Second, the techniques used in traditional data recovery and in the digital forensic context differ because of the forensic soundness issue. For example, in traditional data recovery, a corrupted main boot record may be repaired by laying a FAT2 over a FAT1 if the FAT2 is intact. However, this is not an appropriate forensic data recovery technique because the original evidence (FAT1) is modified. Instead, it would be necessary to repair the corrupted main boot record in a duplicate (i.e., image). Finally, forensic data recovery embraces a broader view of recovering data than traditional data recovery and, consequently, must consider issues (e.g., hidden data and trace data) that are beyond the purview of traditional data recovery.

The data recovery function is mapped by detailing its components, processes and relevant factors. Since the goal of data recovery is to retrieve data due to storage media abnormalities and/or intentional human manipulation, the function mapping is performed from three angles: (i) storage media; (ii) recovery object; and (iii) recovery reason. Figure 2 presents the top-level ontology of the data recovery function.

## 3.1 Storage Media

Data is typically stored as files on storage media. The files are managed (i.e., created, modified and deleted) by file systems. In order to perform data recovery effectively and efficiently, forensic investigators

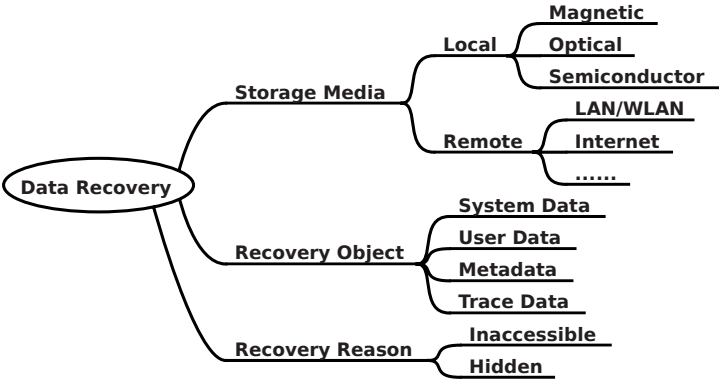


*Figure 2.* Top-level data recovery function mapping.

need to understand the physical media as well as the logical structures of files on the media.

Data recovery may be conducted locally (i.e., the storage media are seized and are under the custody of the investigator) or remotely (i.e., the investigator uses a network to access the storage media).

From the physical (material) point of view, storage media can be categorized as: magnetic, optical or semiconductor. The magnetic storage media category includes hard drives, RAID arrays, floppy disks, zip disks and tape drives (Figure 3). Typical hard drive types include ATA, SATA, SCSI, IDS and USB. The file systems used include FAT (12, 16, 32), NTFS, HFS (HFS+) and EXT (2, 3, 4).

Typical optical storage media are CDs and DVDs (Figure 4). CD storage media are in the form of CD-ROM, CD-R and CD-RW. Common file systems for CD media are ISO-9660, UDF, Joliet, HFS and HSG. DVD media include DVD-ROM, DVD-R(+R) and DVD-RW(+RW). The principal file systems for DVD media are UDF and HFS.

The principal semiconductor-based storage media are RAM and ROM (Figure 5). Flash memory, a type of EPROM (erasable programmable read-only memory), is widely used in computers and electronic devices and includes compact flash (CF) cards, smart media (SM) cards, secure digital (SD) cards, memory sticks and USB flash drives. File systems commonly used in flash memory include FFS, JFFS, LogFS and YAFS.

## 3.2 Reasons for Data Recovery

A data recovery method is used when data is unavailable. In the context of digital forensics, data is unavailable and must be salvaged for various reasons, including damage, corruption or hiding. From the

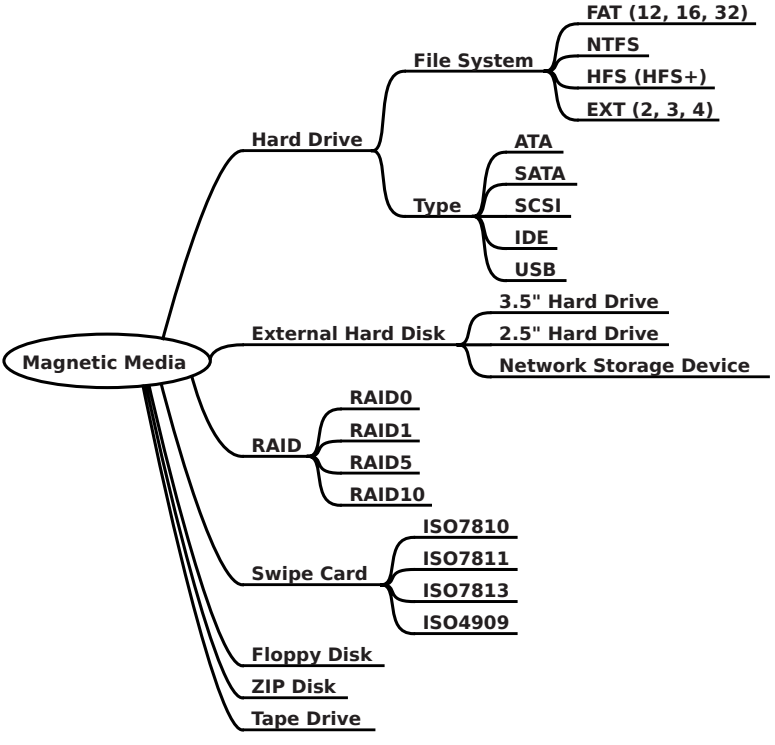


*Figure 3.* Magnetic storage media category.

point of view of the user (e.g., investigator), we assume that the data is unavailable because it is inaccessible or hidden. By inaccessible data, we mean that the user is aware of the existence of the data, but is unable to access it in a normal manner. On the other hand, hidden data is invisible to the user and the user does not know of its existence.

**Inaccessible Data** "Orphaned" files are inaccessible to users under normal operations. An orphaned file is one that no longer has a parent (the parent is the folder in which it was originally located) [4]. The term orphaned is a broad concept that includes deleted files. In most cases, orphaned files are deleted files, but a file can be orphaned when the association with its parent is lost through other means (e.g., by removing a symbolic link in a Unix environment).

Ambient space (unallocated space) or space that is orphaned from the operating system or file system has many forms. Data in such space cannot be accessed by users under normal operations. For example, file slack space is ambient or unallocated space that exists at the end of a file

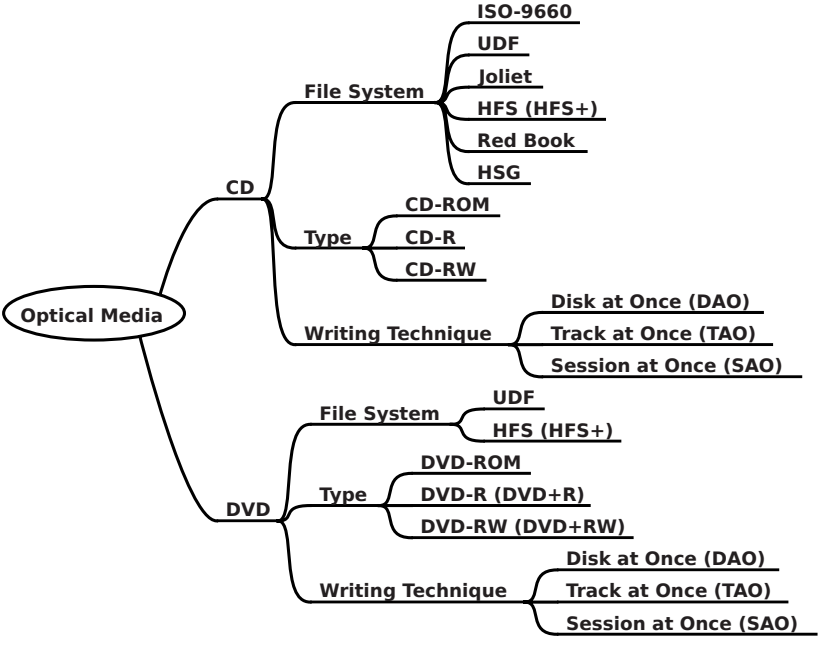


*Figure 4.* Optical storage media category.

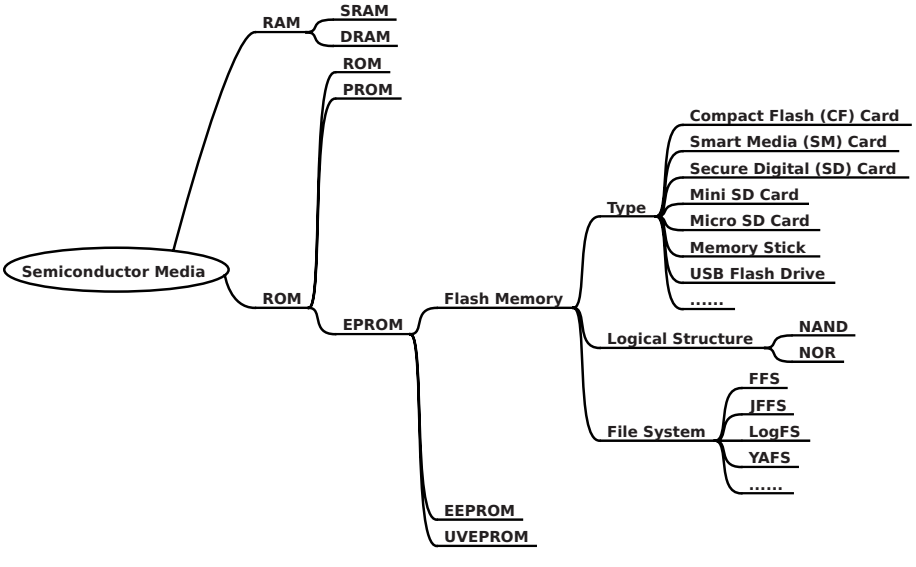


*Figure 5.* Semiconductor storage media category.

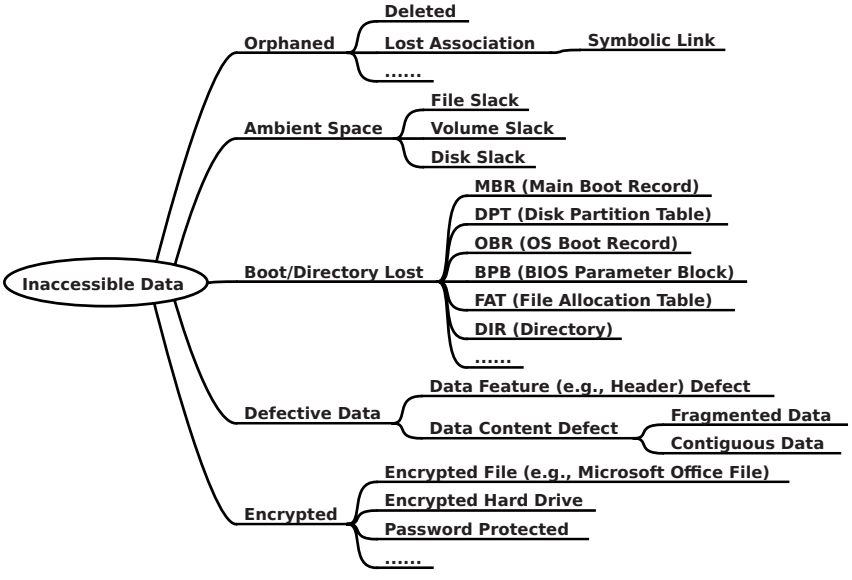


*Figure 6.* Inaccessible data category.

in certain operating systems and can contain a variety of data, including data dumped from RAM (RAM slack) or the remnants of previously-allocated files that may have been orphaned and partially overwritten.

Data may also be inaccessible because its metadata is corrupted or missing. The associated metadata includes MBR, DPT, OBR (operating system boot record), BPB (BIOS parameter block), FAT and DIR (directory). In such scenarios, the file may not be located, but its data is intact and, therefore, can be recovered by "file carving" [8].

Alternatively, a file can be located using metadata, but the data itself cannot be accessed because it is defective. This can occur for two reasons. One possibility is that the data feature (e.g., header or footer) is damaged. A file header is a "signature" placed at the beginning of a file to enable the operating system or application to know what to do with the following contents. The file cannot be recognized when this feature is damaged. The second possibility is that the data content is corrupted. In this case, it is necessary to analyze the structural characteristics and code of the damaged file to recover the data or portions of the data.

Finally, data may be inaccessible due to encryption and steganography. Although an encrypted file is visible to users, its contents are inaccessible without the key. Figure 6 summarizes the inaccessible data category.

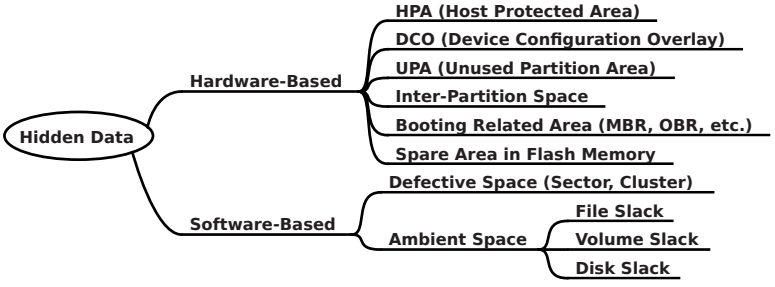


*Figure 7.* Hidden data category.

**Hidden Data** In the digital forensic context, it may be necessary to recover data that has intentionally been hidden. Data hiding methods may be categorized as hardware-based or software-based (Figure 7). Hardware-based methods hide data in specific areas of storage media. For example, data on a hard disk may be stored in the HPA (host protected area), DCO (device configuration overlay), UPA (unused partition area) and inter-partition space.

Software-based methods hide data using file system and/or operating system utilities [3]. For example, modern hard disk controllers handle bad sectors without the involvement of the operating system by slipping (modifying the LBN (logical block number) to physical mapping to skip the defective sector) or remapping (reallocating the LBN from a defective area to a spare sector). For older hard disks that do not have this capability, the operating system and file system have to retain the ability to detect and mark defective sectors and clusters as damaged. This feature can be used to exclude undamaged clusters from normal file system activities and use them to hide data.

Software-based data hiding methods may also use ambient space. Slack space, which includes file slack space, volume slack space and partition slack space, are areas on the disk that cannot be used by the file system because of the discrete nature of space allocation. Data can be hidden in any of these locations.

## 3.3 Recovered Objects

File system data to be recovered belongs to one of four categories: system data, user data, metadata and trace data.

**System Data** System data includes general hardware and software information. Data recovery techniques include hardware rendering and software (operating system and file system) rendering (Figure 8).

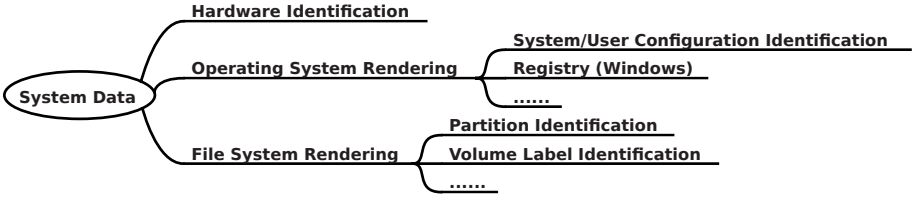


*Figure 8.* System data.

Hardware rendering refers to the ability to accurately identify particular types of devices and media. This is accomplished through physical interaction with the device or media or by using metadata located on the device or media.

The goal of operating system or file system rendering is to reveal the underlying structure. Operating systems and file systems have a general structure, but each instance is unique. In addition, many of these systems are proprietary in nature and, as a result, are poorly documented (e.g., the detailed structure of NTFS has not been publicly released). Operating system and file system rendering may specify where certain structures are found and the data unit size that enables file folders, data and metadata to be accurately retrieved. For example, the volume label and the associated data are indicators of the method used to create the allocated components of a device. Different file systems record this information differently, so a digital forensic tool must be able to render the volume label(s) from a device or partition.

**User Data** User data is the principal object of data recovery. User data are categorized as document, graphic, sound or Internet files. Figure 9 presents the classification and provides typical instances of each class. This classification is by no means exhaustive and will have to be updated constantly to accommodate new applications and file formats. Note that user data files may be in special forms (e.g., compressed and encrypted), which should be taken into account by forensic examiners.

**Metadata** Metadata is data that describes data or files. It includes data about where the file content is stored, file size, dates and times of the last read and write, and access control information. Figure 10 presents examples of metadata in various storage and file systems. Metadata must be analyzed to determine details about a specific file or to search for a file that meets certain requirements.

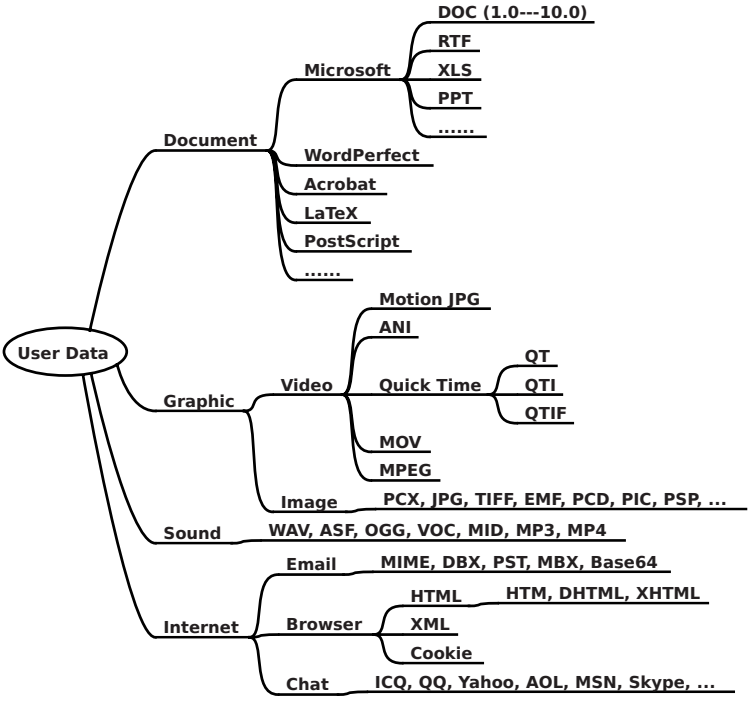


*Figure 9.* User data.

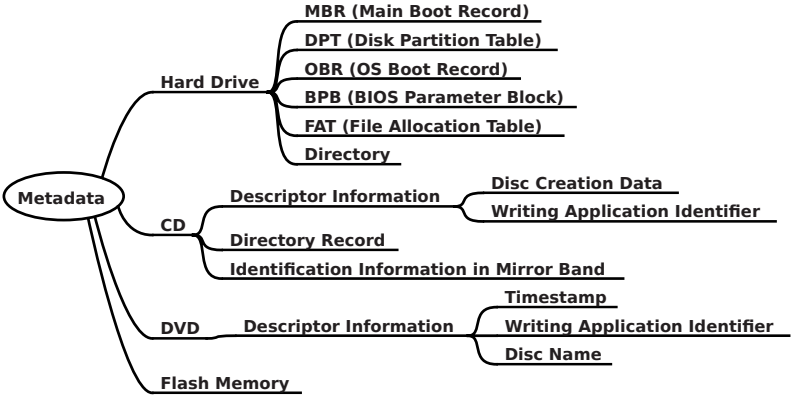


*Figure 10.* Metadata.

**Trace Data** As mentioned above, data recovery in the digital forensic context is a much broader concept than traditional data recovery. Trace data is the data that remains on the storage media after operations

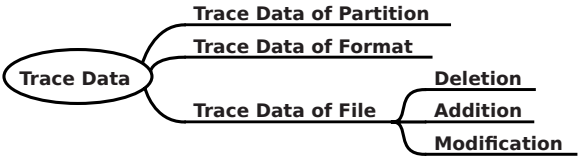


*Figure 11.* Trace data.

such as hard drive partitioning, formatting and file deletion (Figure 11). Trace data may not be substantial, but may constitute important digital evidence. For example, file operations (e.g., creation, modification and deletion) leave traces in the form of temporary files. Most temporary files are deleted by the operating system after the file operations are completed. However, if a temporary file is deleted, it contents can be recovered if the clusters allocated to the file are not reallocated. Also, even if the allocated clusters are reallocated, file metadata (e.g., name and timestamp) may exist and may prove to be useful in a digital forensic investigation.

## 4. Requirements Specification

Requirements specification is the second step of the validation and verification framework. The data recovery function requirements are specified in the same way as the search function requirements in [2].

The requirements are specified in an extensible and customized manner. As seen in function mapping, several issues have to be considered when specifying the requirements. For example, the storage media could be a hard disk, CD/DVD, flash memory, etc. The file system that manages data files on the storage media could be FAT12, FAT16, FAT32, EXT2, EXT3, NTFS, HFS(+), FFS, etc. The data could be inaccessible for any number of reasons; it could be orphaned, corrupted, encrypted, etc. Each of these sub-categories again has many variations.

The method of specifying requirements is highly abstract and generalized. We use italicized "variables" to reflect these variations. Thus, when a requirement has to be changed, it is only necessary to adjust (add, delete or modify) the variables. Moreover, the requirements can be unwrapped when it is necessary to develop a specific test scenario in a reference set. For example, the requirement: "The tool shall be able to accurately recover *inaccessible recovery objects*" may be unwrapped and instantiated as "The tool shall be able to accurately recover *deleted JPG files*" or "The tool shall be able to accurately recover *hidden data in file slack*."

A digital forensic tool has the following eight requirements with respect to the data recovery function:

- The tool shall operate in at least one *operational environment*.
- The tool shall operate under at least one *operating system*.
- The tool shall operate on at least one type of *storage media*.
- The tool shall be able to accurately render *system data*.
- The tool shall be able to accurately recover *inaccessible (recovery) objects*.
- The tool shall be able to accurately recover *hidden (recovery) objects*.
- If there are unresolved errors when reconstructing data, then the tool shall report the *error types* and *error locations*.
- The tool shall report the *attributes* of the recovered data.

## 5. Reference Set Development and Testing

A reference set consists of test scenarios (cases) against which a digital forensic tool or its individual function is validated. The development of test scenarios is based on the specification of function requirements. Using the requirements specification, it is possible to establish a reference set for testing the data recovery function of various digital forensic tools. Since the function requirements are specified in an extensible manner, the corresponding reference set is also extensible. This would enable practitioners, tool developers and researchers to identify critical needs and to target deterministic reference sets.

We have identified eight requirements for the data recovery function. Since each requirement has several variables, multiple test scenarios have to be designed for each requirement. Each scenario represents a single instantiation of each variable. The following are some pilot samples of the reference set for the data recovery function:

- A deleted JPG file in a FAT32 file system on an IDE hard disk.
- A deleted JPG file in an NTFS file system on a SCSI hard disk.
- A deleted WAV file in a UDF file system on a CD.
- A deleted Microsoft Word file in an FFS file system on flash memory.

- A deleted compressed HTML file in an NTFS file system on an IDE hard disk.
- An encrypted MP3 file in a FAT32 file system on an ATA hard disk.

Thus far, we have completed the function mapping, requirements specification and reference set development. We now know what needs to be tested and what the expectations are. Validating a digital forensic tool that professes to have a search function is now as simple as testing the tool against the reference set and applying metrics (accuracy and precision) to determine the quality of the results.

## 6. Conclusions

Mapping the fundamental functions of the digital forensic discipline is a powerful approach for creating a function-oriented validation and verification paradigm for digital forensic tools. The utility of the approach is demonstrated in the context of the data recovery function via the specification of data recovery requirements and a reference set for testing tools that implement the data recovery function. Validating a digital forensic tool is reduced to testing the tool against the reference set. Compared with traditional testing methods, this testing paradigm is extensible, and neutral and transparent to specific tools and tool versions.

More work remains to be done to complete the validation paradigm. Although the methodology holds promise, it needs to be tested extensively to evaluate its utility and identify potential weaknesses and shortcomings. Tests would have to be implemented against popular tools such as EnCase and FTK. A quantitative model is also required to evaluate the results of validation and verification. Metrics are needed to measure the accuracy and precision of testing results, and it is necessary to specify rules for judging the validity of digital forensic tools. Is a tool validated only when it passes all the test cases? Or is a tool validated when it passes the test cases for certain scenarios?

It is important to recognize that numerous variables are involved in function requirements specification and that the corresponding reference set can be very large. Indeed, the number of possible combinations for validating a single function in a digital forensic tool may well be in the thousands (even discounting the different versions of the tool). Interestingly, this problem is also faced by the Computer Forensics Tool Testing (CFTT) Program [7] created by the National Institute of Standards and Technology (NIST) to validate and verify digital forensic tools. This problem will be examined in our future work.

## References

[1] J. Beckett and J. Slay, Digital forensics: Validation and verification in a dynamic work environment, *Proceedings of the Fortieth Annual Hawaii International Conference on System Sciences*, p. 266, 2007.

[2] Y. Guo, J. Slay and J. Beckett, Validation and verification of computer forensic software tools – Searching function, *Digital Investigation*, vol. 6(S1), pp. S12–S22, 2009.

[3] E. Huebner, D. Bem and C. Wee, Data hiding in the NTFS file system, *Digital Investigation*, vol. 3(4), pp. 211–226, 2006.

[4] D. Hurlbut, Orphans in the NTFS world, AccessData, Lindon, Utah (www.accessdata.com/media/en_US/print/papers/wp.NT_Orphan_Files.en_us.pdf), 2005.

[5] R. McKemmish, What is forensic computing? *Trends and Issues in Crime and Criminal Justice*, no. 118 (www.aic.gov.au/publications/tandi/ti118.pdf), 2002.

[6] G. Mohay, A. Anderson, B. Collie, O. de Vel and R. McKemmish, *Computer and Intrusion Forensics*, Artech House, Norwood, Massachusetts, 2003.

[7] National Institute of Standards and Technology, Computer Forensics Tool Testing Program, Gaithersburg, Maryland (www.cftt.nist.gov).

[8] A. Pal and N. Memon, The evolution of file carving, *IEEE Signal Processing*, vol. 26(2), pp. 59–71, 2009.