NISTIR 8354-DRAFT

Digital Investigation Techniques: *A NIST Scientific Foundation Review*

James R. Lyle
Barbara Guttman
John M. Butler
Kelly Sauerwein
Christina Reed
Corrine E. Lloyd

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8354-draft

National Institute of
Standards and Technology
U.S. Department of Commerce

NISTIR 8354-DRAFT

Digital Investigation Techniques: *A NIST Scientific Foundation Review*

James R. Lyle
Barbara Guttman
Software and Systems Division

John M. Butler
Kelly Sauerwein
Christina Reed
Corrine E. Lloyd
Special Programs Office

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8354-draft

May 2022

U.S. Department of Commerce
Gina Raimondo, Secretary of Commerce

National Institute of Standards and Technology
Laurie E. Locascio, NIST Director and Under Secretary of Commerce for Standards and Technology

National Institute of Standards and Technology Internal Report 8354-DRAFT
(May 2022)

Acknowledgments: Gregory Haber and Will Guthrie of the NIST Statistical Engineering Division performed the capture recapture analysis in Sec. 2.7.2. Helpful feedback was provided by Simson Garfinkel, Eoghan Casey, Brandon Epstein, Bill Eber, and Tracy Walraven. The Scientific Working Group on Digital Evidence and the Organization of Scientific Area Committees for Forensic Science Digital Evidence Subcommittee served as valuable sources of information about current practices in digital forensics. Members of the NIST Computer Forensics Tool Testing team, Rick Ayers, Craig Russell, Jenise Reyes-Rodriguez, and Mary Laamanen, provided needed expertise.

Public comment period: May 9, 2022 through July 11, 2022

The initial release of this report is a draft document, and we welcome comments and feedback from readers. All relevant submitted comments will be made publicly available and will be considered when finalizing this report. Do not include personal information, such as account numbers or Social Security numbers, or names of other individuals. Do not submit confidential business information, or otherwise proprietary, sensitive, or protected information. We will not post or consider comments that contain profanity, vulgarity, threats, or other inappropriate language or like content. During the 60-day comment period, comments may be sent to scientificfoundationreviews@nist.gov.

All comments, including commenter name and affiliation, will be published at https://www.nist.gov/topics/forensic-science/interdisciplinary-topics/scientific-foundation-reviews.

National Institute of Standards and Technology
Attn:Special Programs Office – Scientific Foundation Review
100 Bureau Drive Stop 4701
Gaithersburg, MD 20899-4701

Email: scientificfoundationreviews@nist.gov

National Institute of Standards and Technology Interagency or Internal Report 8354-draft
Natl. Inst. Stand. Technol. Interag. Intern. Rep. 8354-draft, 84 pages (May 2022)

https://doi.org/10.6028/NIST.IR.8354-draft

Preface

Forensic science plays a vital role in the criminal justice system by providing scientifically based information through the analysis of physical and digital evidence. The National Institute of Standards and Technology (NIST) is a non-regulatory scientific research agency within the U.S. Department of Commerce with a mission to advance measurement science, standards, and technology and has been working to strengthen forensic science methods for almost a century. In recent years, several scientific advisory bodies have expressed the need for reviews of the scientific basis of forensic methods and identified NIST as an appropriate agency for conducting them. A scientific foundation review, also referred to as a technical merit evaluation, is a study that documents and assesses the foundations of a scientific discipline, that is, the trusted and established knowledge that supports and underpins the discipline's methods. Congress has appropriated funds for NIST to conduct scientific foundation reviews in forensic science. These reviews seek to answer the question: "What established scientific laws and principles as well as empirical data exist to support the methods that forensic science practitioners use to analyze evidence?" Background information on NIST scientific foundation reviews is available at https://doi.org/10.6028/NIST.IR.8225.

Abstract

This document is an assessment of the scientific foundations of digital forensics. We examined descriptions of digital investigation techniques from peer-reviewed sources, academic and classroom materials, technical guidance from professional organizations, and independently published sources. Digital investigation techniques are based on established computer science methods and when used appropriately are considered reliable. The process of evaluating, for example, the contents of a computer hard drive does not create information that was not there before the investigation started. However, because the field is rapidly changing there are limitations that practitioners and stakeholders need to be aware of: (1) as with any crime scene not all evidence may be discovered; (2) when recovering deleted files, the results may include extraneous material; (3) examiners need to understand that as software (operating systems and applications are revised) the meaning and significance of digital artifacts created by the software can change over time.

In addition, because there are often multiple ways to search for information, two examiners may find different information, and both can be correct. The methods used in digital investigations are often not peer-reviewed in a formal process, but trustworthiness is established by members of the digital forensic community trying out proposed methods, testing, and updates circulated within the community. This process strengthens an examiner's awareness of the capabilities and limitations of their techniques.

Key words

digital forensics, digital evidence, computer forensics, digital investigation, scientific foundations

Table of Contents

Glossary and Acronyms

Terms and Acronyms	Definition of Term
AAFS	American Academy of Forensic Sciences.
Advanced Format	Created to address technical issues with the 512-byte storage device sector size by changing storage device sector size from 512-bytes to a multiple of 512-bytes such as 4096-bytes, i.e., storage devices with a sector size larger than 512-bytes.
Algorithm	A sequence of steps for solving a problem or accomplishing a task.
Anti-forensics	Active measures and techniques taken by a computer user to mislead or obstruct an examiner. Common methods include deleting relevant files, creating bogus artifacts, modifying time stamps, log file alterations, creation of file system artifacts that can disrupt operation of common forensic tools and other measures.
APFS	Apple File System. One of the file systems supported on Macintosh Computers. APFS was introduced in 2017.
Artifact	A digital artifact is a singular unit of interpretable data that can be extracted from a given data source that is useful for addressing questions in forensic investigations.
ASCII	American Standard Code for Information Interchange is a character encoding standard for electronic communication.
ATA	Stands for AT Attachment, also known as PATA (Parallel ATA) or IDE (Integrated Drive Electronics). ATA is a protocol for connecting storage devices to a host computer. Note: AT is an IBM PC model name, not an acronym.
Binary	A base-2 representation for numbers that uses a sequence of 1's and 0's to write a number. See *Place Value Notation*.
BIOS	Basic Input Output System. PC computer firmware to perform hardware initialization during the PC power-on startup process and provide other services to the operating system.
CFReDS	Computer Forensics Reference Data Sets. A repository at NIST of community created test data sets for testing digital forensic tools, including CFTT test data sets.

Terms and Acronyms	Definition of Term
CFTT	Computer Forensic Tool Testing. A project at NIST for testing digital forensic tools.
Chip-Off	A destructive method of acquiring digital data from a device by removing memory chips from a printed circuit board and then directly copying the data from the chip.
CRC	Cyclic Redundancy Check is an error-detecting code commonly used to detect accidental changes to transmitted data.
Data Acquisition	The general process of making a copy of digital data. This can be an entire digital device, just a partition from a storage device, or selected files from a file system.
DC3	Defense Cyber Crime Center.
DCO	Device Configuration Overlay. Used to change the features offered by a storage device to present a subset of the available features and change the apparent storage capacity of a storage device to a smaller size.
DE	Digital Evidence. Digital evidence or electronic evidence is any probative information stored or transmitted in digital form that a party to a court case may use at trial.
DFRWS	Digital Forensics Research Workshop
DFRWS-EU	Digital Forensics Research Workshop Europe
DHS	Department of Homeland Security.
Disk Imaging	The process of acquiring the digital contents of a storage device (fixed disk, removable disk, flash drive, etc.). This acquires all the data on a device including files, metadata, and contents of unallocated areas of the device.
DOJ	Department of Justice.
EBCDIC	Extended Binary Coded Decimal Interchange Code is a character encoding used on older IBM mainframe computers.
ECC	Error Correcting Code. A method to ensure accurate detection and correction of transmission errors when data is moved from one

Terms and Acronyms	Definition of Term
	place to another, e.g., memory to memory transfers, storage device to memory transfers.
Encode	In computing, to represent information as numbers. For example, text can be encoded by assigning each letter a unique number.
Encrypt	To encode information in a way that prevents unauthorized access. For example, decryption with a key is required to access the information.
ExFAT	Extensible File Allocation Table. A revised implementation of the FAT file system introduced in 2006 that addresses some shortcomings in the FAT file system, e.g., allows files larger than 4GB, and faster performance.
Exif	A standard metadata format employed in specific digital still camera file formats, e.g., JPEG. While EXIF is a specific type of metadata, the term is used colloquially in reference to a variety of metadata embedded in audio and image files describing the file content. Audio files may have metadata such as artist, copyright, creation date, and more. An image file may have camera make, model, exposure settings, geolocation and more.
Ext4	Fourth Extended File System. The default file system for many Linux distributions as of this writing. Ext4 was introduced in 2008 as a replacement for the earlier ext2 and ext3 Linux file systems.
FAT	File Allocation Table (file system). A file system developed for Microsoft computers introduced in 1977 and revised and extended over the years. Versions include FAT12 (12-bit addresses), FAT16 (16-bit addresses) and FAT32 (32-bit addresses).
File System	A method for organizing files on a storage device. Common file systems on Windows systems are NTFS, ExFAT and FAT. LINUX systems use ext4 and FAT. Apple Macs use HFS+, APFS, FAT and ExFAT.
Fixed media	A storage device that is physically installed in a computer.
Hash	A mathematical technique that computes a hash value (short, fixed length) from a possibly much longer set of data. Hashes can be designed to exhibit several useful properties depending on the

Terms and Acronyms	Definition of Term
	intended application. In digital forensics, *cryptographic hashes* are usually used that have the following properties: • The same input always produces the same output. • The original input data cannot be reconstructed from the output hash. • Hash values from files with small differences have hash values with large differences. • Chance of two different files selected at random having the same hash value is so small that it is essentially zero. These hashes can be used to verify that a file, e.g., an acquisition from a device, has not changed, or find copies of known contraband. Some cryptograph hashes in current use include Message Digest 5 (MD5), Secure Hash Algorithm (SHA-1, SHA-2 & SHA-3). SHA-2 and SHA-3 come in several variants.
Hexadecimal	Hexadecimal is a base 16 number system than uses in addition to the digits 0-9, 6 letters (A through F) rather than the traditional base 10 decimal system. See *Place Value Notation*.
HFS Plus	Hierarchical File System Extended. Apple file system introduced in 1998, replaced by APFS in 2017.
HPA	Host Protected Area. A hidden area that can be configured on a storage device.
HTCIA	High Technology Crime Investigation Association.
IDEMA	International Disk Drive Equipment and Materials Association is a trade organization that represents the disk drive industry.
JTAG	Joint Test Action Group. An industry standard for verifying designs and testing printed circuit boards after manufacture.
LBA	Logical Block Address. A scheme to locate data on a storage device. An LBA of 0 is the first block of data on the storage device, LBA of 1 is the next block of data and so on.
MAC Times	Time stamp metadata maintained by a file system to track events in the life cycle of a file. The exact events recorded depends on the operating system and the file system. The usual meanings are Modify, Access, and Create with slight differences in meaning for

Terms and Acronyms	Definition of Term
	each type of file system and differences in meaning for files and directories.
MD5	Message Digest 5. A commonly used cryptographic hash algorithm.
Metadata	Metadata is a description of stored data. Categories of metadata include: (1) application metadata (in a document this could be author, organization, etc., in a database such as SQLite there is metadata to describe the layout of the stored data within the database), (2) file system metadata (placement of the file within the file system, owner, permissions, MAC times, etc.), (3) partition metadata that identifies the type of file system the partition contains and global file system parameters, and (4) device metadata describes the layout of partitions on a device.
NTFS	New Technology File System. Microsoft Windows file system introduced in 1993, revised several times over the years.
NW3C	National White Collar Crime Center.
Operating System	The software that creates the digital environment for running software on a computer or other digital device. Most operating systems are variants of either MS Windows (95, 98, 2000, Vista, XP, 10, etc.) or UNIX (BSD, Linux, Mac OS, iOS, etc.).
OSAC	Organization of Scientific Area Committees for Forensic Science.
Partition	A contiguous area of a storage device used to contain a formatted file system.
Partition Table	A table describing the layout of a physical storage device that has been divided into partitions, each partition contains a separate file system.
PhotoDNA	A hashing technique that creates similar hashes for similar image files. The calculation of the hash is based on image content and not the binary representation of the image file. It also addresses reformatting of an image from one format to another, e.g., JPG to PNG, since the image content stays the same even though the binary representation changes significantly.

Terms and Acronyms	Definition of Term
Place Value Notation	A method for representing numbers using a sequence of symbols selected from a fixed set of symbols that are assigned value based on the relative position within the sequence.
Removable media	A storage device that is either (1) a data container that is inserted and removed from a data reader or (2) a storage device that can be connected or removed from a computer while the computer is running.
SATA	Serial ATA. A protocol for connecting storage devices to a host computer.
SCSI	Small Computer System Interface is a protocol for connecting storage devices to a host computer.
SHA	Secure Hash Algorithm. A family of cryptographic hash algorithms approved by NIST for security applications. Includes SHA-1, SHA-2 and SHA-3.
SIM card	A Subscriber Identity Module is another, older, name for a UICC card.
Storage Device	An electronic or optical device that can store data for later retrieval. A storage device usually has some type of *file system* to organize the stored data as files. There are several types: • Fixed media physically installed in a computer. The computer must be powered off to install or remove the storage device. • Removable media. Can be installed or removed while the computer is running. Small storage devices are called *flash drives* or *thumb drives* (they are about the size of a human thumb). These devices are usually connected via a USB interface. • Memory card. One of several digital storage media types that can be inserted into a compatible card reader, e.g., SD card. • Optical disk. A CD or DVD in one of several formats.
SWGDE	Scientific Working Group on Digital Evidence.
UICC card	A Universal Integrated Circuit Card (also called a SIM card) contains phone number and account information for mobile

Terms and Acronyms	Definition of Term
	devices. An integrated circuit card that securely stores the international mobile subscriber identity (IMSI) and the related cryptographic key used to identify and authenticate subscribers on mobile devices.
Volatile	Data stored on a device that is lost when power is removed from the device. Removing power usually resets all binary digits to zero. For example, computer memory is lost when power to the computer is turned off.
Write Blocking	Techniques designed to prevent any modification to digital media during acquisition or browsing.

Executive Summary

Every interaction with a digital device has the potential to leave a trail of what we did, who we did it with, where we were, and when the event took place. This trail is made up of digital artifacts, which are created in the routine operation of a digital device. This trail can assist an investigator to discover and explain what happened. Computers generate many artifacts, most of which do not contribute to understanding what happened. The challenge is finding useful information and separating it from irrelevant information. Digital investigation techniques can extract this information and construct a narrative of the events. The analysis of digital devices for investigative purposes is widely practiced and, as this report shows, there are at least 11,000 digital forensic laboratories in the United States.

In recent years, several scientific advisory bodies have expressed the need for scientific foundation reviews of forensic disciplines and identified NIST as an appropriate agency for conducting them. The purpose of a scientific foundation review is to document and consolidate information supporting the methods used in forensic analysis and identify knowledge gaps where they exist. In addition to this report on digital investigation techniques, the initial scientific foundation reviews conducted by NIST include DNA mixture interpretation, bitemark analysis, and firearm examination (Butler et al. 2020).

To address the question of the scientific basis of digital investigation, NIST examined the scientific literature on digital forensics as well as multiple other sources (see Sec. 2.8 and Sec. 3). The review was led by a senior computer scientist and a multidisciplinary team from various areas at NIST. The team identified seven categories of digital forensic activities which were studied.

Obtaining input from experts outside of NIST is an integral component of a NIST scientific foundation review. As described in Chapter 3, the NIST team followed the process outlined in NISTIR 8225 (Butler et al. 2020) for conducting this review in terms of obtaining input from the community including:

- collecting and evaluating the peer-reviewed literature,
- assessing publicly available data from interlaboratory studies, proficiency tests, and laboratory validation studies,
- exploring other available information including position statements and non-peer reviewed literature, and
- obtaining input from members of the relevant community through interviews, workshops, working groups, and other formats for the open exchange of ideas and information.

The overall finding of this report is that digital evidence examination rests on a firm foundation based in computer science. Several of the techniques had already been extensively studied and documented in the peer-reviewed literature. Others are documented more informally through community discussion forums. The application of these computer science techniques to digital investigations is sound, only limited by the difficulties of keeping up with the complexity and rapid pace of change in IT.

There are many ways to organize tasks performed in digital investigations; for this report, the following grouping of tasks is used:

1. Protect data from modification. This is usually accomplished, by write blocking, i.e., monitoring access to a storage device for any data modifying attempts and suppressing the attempt at modification. This is discussed in Sec. 4.1.
2. Acquire digital data. This is accomplished by copying data to make an image file of the acquired digital data. Copying digital data accurately is based on established engineering techniques such as error detecting and correcting codes to ensure that data is copied accurately. This is discussed in Sec. 4.2.
3. Ensure integrity of acquired data. Cryptographic hashing is used to ensure that if acquired digital data is changed inadvertently or deliberately, the change can be detected. This is discussed in Sec. 4.3.
4. Recover deleted data. In some situations, recovery and reconstruction of deleted data makes it possible to bring back deleted files (in whole or in part) or internal records from within an application file. Recovering deleted data has several risks including missing data and conflating unrelated data. Any recovered item must be evaluated by the examiner for indications of problems. This is discussed in Sec. 4.4.
5. Navigate the acquired digital data. This is accomplished by unraveling, i.e., parsing the layout of the acquired data. This is best performed using a software tool. There is the risk that an incorrect implementation will not correctly interpret the structure of a particular file system, e.g., not showing all acquired active files. This is discussed in Sec. 4.5.
6. Identify and extract data artifacts. Items of interest are identified so they can be located and extracted by navigating the acquired data to find artifacts that meet criteria of interest such as, data that contains a specific text string, or association of an event with a specific date and time. This is discussed in Sec. 4.6.
7. Analyze. Examination of extracted artifacts can help develop a narrative or reconstruction of relevant events for inclusion in a final written report. This is discussed in Sec. 4.7.

The following 12 key takeaways have been identified in this report. Their number (#x.y) corresponds to which chapter they are located in (x) and their sequence within that chapter (y).

1. **KEY TAKEAWAY #2.1**: In routine operations computers store much more data than what is presented to the user. Examples include storing time and location data on photos, extra copies of data, and data about system activities.

Forensic tools and techniques can reveal this data to provide a window into activities that have taken place on a computer or other digital device.

2. **KEY TAKEAWAY #2.2**: Digital forensics is dependent on an understanding of computers and how they work. Any activity that is performed by a computer can potentially be a target for a forensics tool or technique.

3. **KEY TAKEAWAY #2.3**: Computer technology evolves rapidly but sporadically. Some attributes of computers last for decades and some only for a few weeks.

4. **KEY TAKEAWAY #2.4**: The forensic examiner needs to be aware of key changes in computing technology relevant to the examination being performed. Frequent changes in digital technology introduces the possibility for incomplete analysis or for misunderstanding of the meaning of artifacts.

5. **KEY TAKEAWAY #2.5:** Not every digital forensic technique undergoes a peer review, formal testing, or error rate analysis. In general, the digital forensics community performs an informal review by providing feedback about the usefulness of techniques. This general acceptance process allows for techniques to be quickly evaluated and revised.

6. **KEY TAKEAWAY #4.1:** When using techniques to recover deleted or hidden artifacts the examiner must determine the relevance of the recovered information as it may be incomplete or improperly merged with irrelevant information.

7. **KEY TAKEAWAY #4.2:** Searching tools have limitations based on the multiple ways that computers store information. Limitations include the type of files, types of encoding, and many other parameters. In general search tools are very effective at finding information, but there is a possibility that data will be missed because a tool does not have the capability to find it.

8. **KEY TAKEAWAY #4.3:** If someone has taken steps to change information in digital evidence to mislead an examiner, it may be difficult to detect the changes. Depending on the sophistication of the manipulation, identification of the changes relies on the skill of the examiner.

9. **KEY TAKEAWAY #4.4:** Digital processes tend to have systematic errors rather than random errors. Therefore, an error mitigation analysis provides more information and is the correct way to manage uncertainty. Asking for an error rate is only useful where there are random errors.

10. **KEY TAKEAWAY #4.5:** When error rates are provided, it is important for the user to understand the context of the numbers. Errors in computer science techniques tend to be so small as to be negligible. For some forensic techniques,

the error rates may vary significantly based on attributes of the technology and usage patterns.

11. **KEY TAKEAWAY #4.6:** It is not feasible to test all combinations of tools and digital evidence sources.

12. **KEY TAKEAWAY #4.7:** Extensive tool testing of over 250 widely used digital forensic tools showed that most tools can perform their intended functions with only minor anomalies.

In addition to addressing the scientific foundation of digital investigation, it is critical that digital findings are communicated clearly. Because of the breadth of digital evidence tools and techniques, it is challenging to properly communicate the results of a digital examination. Some of the basic topics are familiar to most lay people, but the more advanced topics can be rather difficult to understand. Hopefully this report will be helpful in communicating the underlying science and its limitations.

1 Chapter 1: Introduction

Digital devices have become ubiquitous in our lives. Many of the tasks of everyday lives are intertwined with mobile digital devices such as cell phones and tablets, personal computers, embedded digital devices and other digital devices. Every interaction with a device has the potential to leave a trail of what we did, who we did it with, where we were and when the event took place. Digital forensics is the application of the scientific method to make sense of the trail left by the interaction with a digital device. All scientific methods have limitations. One must understand those limitations to use a method appropriately. This is especially important in forensic science as critical decisions impacting life and liberty are often based on the results of forensic analysis.

This document is a review of the scientific foundations of digital forensics. We are asking what empirical data exists to support the methods that digital forensic practitioners use to identify and characterize evidence and associate it with people, places, and things from past events. Our approach is to identify and classify the methods and techniques used by the digital examiner and locate relevant literature validating the reliability of the method and to determine whether the scientific approaches, and practices for digital forensics are well-supported and suitable for use. Knowledge gaps and areas needing further improvement will also be discussed in this report.

1.1 Scope

Due to the wide breadth of potential topics, the scope of this document is limited to techniques to examine digital data stored in an active computer, mobile device memory or on secondary storage, such as a hard drive, or flash drive, etc. Other digital forensics topics such as network analysis and multimedia (video, audio) forensics are not discussed.

1.2 Who Conducted This Review?

The review team consisted of six individuals from the National Institute of Standards and Technology (NIST) whose diverse expertise permitted examination of the issues from many perspectives, including lessons learned in other fields. Table 1-1 lists members of the NIST review team, their NIST operating unit, and their expertise. Assistance in finalizing this report was also provided by several additional NIST employees or contractors as noted in the acknowledgements. Early drafts of this report were also sent to several members of the digital investigation community to seek their input and reaction.

Table 1-1 NIST review team and their areas of expertise

Name	NIST Operating Unit	Areas of Expertise
James R. Lyle	Software & Systems Division	Computer Scientist
Barbara Guttman	Software & Systems Division	Digital Forensics Research Management
John M. Butler	Special Programs Office	Forensic DNA and Scientific Literature
Kelly Sauerwein	Special Programs Office	Forensic Anthropology
Christina Reed	Special Programs Office	Communication and Science Writing
Corrine E. Lloyd	Special Programs Office	Management Analyst

1.3 Related Work

NIST also performed an interlaboratory study (Guttman et al. 2022) as part of its work on the scientific foundation of digital forensics. The study did not attract enough participants to draw meaningful conclusions but did demonstrate that digital forensic examiners could answer difficult questions related to the analysis of mobile phones and personal computers. Responses to the study underscored the size, variety, and complexity of the field.

1.4 How is This Report Structured?

This report contains six chapters. Following this introductory chapter, Chapter 2 provides information on the history of digital forensics and background concepts related to computer science. Chapter 3 lists and describes the data sources used and how they were located. Chapter 4 discusses the reliability of specific tasks critical to digital investigations. Chapter 5 provides conclusions and thoughts on the future directions for the field.

The initial release of this report is a draft document, and we welcome comments and feedback from readers. All relevant submitted comments will be made publicly available and will be considered when finalizing this report. Do not include personal information, such as account numbers or Social Security numbers, or names of other individuals. Do not submit confidential business information, or otherwise proprietary, sensitive, or protected information. We will not post or consider comments that contain profanity, vulgarity, threats, or other inappropriate language or like content. During the 60-day comment period, comments may be sent to scientificfoundationreviews@nist.gov.

1.5 Comparison of Non-Digital to Digital Investigation

Digital investigation techniques are based in computer science. The computer science world is often daunting to the uninitiated as a significant investment is required to learn obscure technical concepts and terminology. However, understanding the process of a forensic examination of digital data is not as difficult as one might first suspect and is analogous to many elements of a non-digital investigation. This section relates tasks in a digital investigation to a non-digital investigation to illustrate their analogous similarities.

Consider a search of an office or residence to find something relevant to an event of interest, possibly a crime, an accident or other event that needs to be better understood. After obtaining proper authorization and warrant for a search then a search can proceed. Digital evidence differs from physical evidence in the concept of search and seizure. For a physical search, the authorization covers searching the location and the seizure of objects of possible evidentiary value. In digital forensics an entire digital storage device, e.g., hard drive or flash drive is taken to then search it for evidence.

Just as in a non-digital investigation, the digital investigation seeks to create a timeline of events (to identify what actions occurred), reconstruct fragmented artifacts, identify a suspect (who committed the crime), means (how the crime was committed), establish opportunity to commit the crime and to find other relevant evidence. The object of the search could be records of nefarious economic activity, possession of contraband, weapons or tools used in a crime or indications of movement of a suspect. The location searched could be anything from a small apartment (a small computer) to a large farm (a server farm with many computers and removable devices) with barns and outbuildings (offline storage and archives), vehicles (mobile devices) and out of the way hiding places (box of CDs/DVDs in a closet). A search of a large property may uncover a skeleton in an unmarked grave, and an examination of the bones[1] may reveal relevant details about the person (deleted file recovery and file carving[2]).

In both digital and non-digital circumstances, the examiner is interested in learning more details about some event of interest and a search of the property is expected to uncover evidence that can be used to inform decision makers such as a judge. Likewise, search of a digital device (computer, mobile phone, removable storage, cloud, or other digital device) seeks to find relevant evidence related to an event. Non-digital investigations are guided by the principle that "Forensic science seeks to establish connections (or lack thereof) between evidence and its source . . . we consider the probability of the evidence in light of competing hypotheses"(Inman and Rudin 2000). In like manner, a digital investigation generates hypotheses, and the investigator searches for data artifacts, e.g., files, logged events with a time stamp, emails, etc., that can be used in evaluating observed evidence in light of alternative (opposing) hypotheses.

Examples of items relevant to a non-digital investigation and possible corresponding items relevant to a digital investigation are presented in Table 1-2.

Table 1-2 Examples correlating elements of a non-digital versus digital investigation

Correspondence of Real (non-digital) World to Digital World Evidence	
Real-World	**Digital-World**
Crime scene or a place to search for evidence: could be a small site like an apartment or a large site like a farm or business.	Computer, mobile device, storage device: a device to be examined; a server farm with many computers.
An item of evidence that is fragmented: shredded document, buried body.	Deleted data: evidence that isn't apparent with the usual computer user tools and can't be examined without some reassembly.
On site records such as a filing cabinet or desk.	Files stored on the computer hard drive, removable media.
Offsite records such as at a business branch office, a summer home, or a storage locker.	Files stored on a cloud server, or off-line on removable media.
Burglar tools or weapons.	Hacking tools.
Names, phone numbers and addresses from a list of contacts, e.g., address book on paper.	Contact list from a mobile device.

It is important to recognize that the goals of both a digital and a non-digital investigation are the same. Both types of investigations revolve around questions critical to identifying the actors and their actions involved in the events under consideration.

---

[1] The examination may require a specialist to do the examination.
[2] The deleted data recovery may require use of an additional tool for the data recovery.

There are general principles of forensics (OSAC 2018) that guide the examination of evidence, building on principles developed earlier (Inman and Rudin 2000):

- Authentication – Is there sufficient confidence that a claim is true?
- Identification – Is there sufficient confidence that something is what it is claimed to be?
- Classification – Is there sufficient confidence that something has been assigned to the appropriate category?
- Reconstruction – Have the elements of the case been organized in the most likely grouping of capabilities, patterns in time and linkages among entities?
- Evaluation – Is there enough information to provide input into a decision process?

Note: in digital forensics authentication is defined by the SWGDE Digital & Multimedia Evidence Glossary as “the process of substantiating that the data is an accurate representation of what it purports to be” (SWGDE 2016c).

In applying these principles, a non-digital investigation may require a variety of forensic tasks such as:

- Surveying the crime’s location.
- Identifying items found at the crime scene, e.g., blood, a bullet, or something dropped by someone present.
- Attempting to identify the source a particular item.
- Extracting useful DNA from biological samples that might be a single source or a mixture.
- Identifying the owner of an item or determining who used the item last.
- Determining what discrete events occurred and their order.

Other more detailed examples of investigative tasks, both digital and non-digital are available (OSAC 2018). A digital investigation usually involves a slightly different, but similar, set of tasks. Some example tasks are:

- Acquiring (or gaining access to) the digital data.
- Ensuring the integrity of the data.
- Reconstructing and recovering deleted artifacts.
- Identifying relevant artifacts.
- Extracting relevant artifacts.
- Classifying relevant artifacts.
- Assembling a narrative of what happened.

In a digital investigation there can be a long list of tasks associated with the analysis each with a different technique required to obtain a resolution. The tasks considered are context sensitive to the type of crime, type of information needed from digital evidence, and types of digital evidence that are available. A digital investigation can encompass many apparently unrelated artifacts that need to be assembled to make a more complete narrative of events.

It is important to recognize that digital evidence is generally a part of a larger investigation. The following example shows how digital evidence can be used as part of an investigation using the hierarchy of propositions (source, activity, offense) from the hypothetico-deductive method (Cook et al. 1998a, 1998b).

Rancher Alejandro reports that his favorite horse, an Appaloosa named Spunky appears to have been stolen last Saturday. Alejandro notes that there is a boot print in the ground by the door next to Spunky's stall. A ranch hand, Big Jake, has been identified as a suspect. There are three levels to consider:

- Level I: Source: The boot print was made by Big Jake's boot versus alternatives such as some other boot left the impression.
- Level II: Activities: Big Jake took Spunky from his stall versus alternatives such as someone else took Spunky.
- Level III: Offense (to be considered by the trier of fact): Big Jake stole Spunky from his stall.

Often both evidence from the physical world and the digital world are combined to get a complete picture of events. For example, an examination (after proper authorization is obtained) of Big Jake's mobile device yields the following items:

- A picture of an Appaloosa horse with a pattern of markings consistent with Spunky was found on Big Jake's mobile device with a time/date stamp of Sunday, after Spunky had been reported stolen and geolocation data for the picture was Big Jake's brother's farm.
- Text messages to a livestock market asking about selling an Appaloosa.

Together the physical and digital evidence paint a basic picture of the events, but additional case work must be done, of course. This example illustrates how elements of both the physical world and the digital world fit together to build a case and how statements about sources, activities, and offences form a hierarchy for consideration.

Much of the burden of accomplishing these digital tasks is carried out by software tools that interpret the bit patterns of digital objects and implement the underlying algorithms designed to accomplish each task. In the end, the job of the digital examiner is to use tools to find relevant information from digital evidence. The questions that the tools can answer range from very general (e.g., show the actual bits stored in a specific location) to very specific (e.g., display the email sent to John Smith on January 1, 2020).

2 Chapter 2: Computer Science Background and History of Digital Forensics

2.1 Computer Background

Before discussing specific tasks, it is helpful to review some background about how computers work, encode and organize digital data.

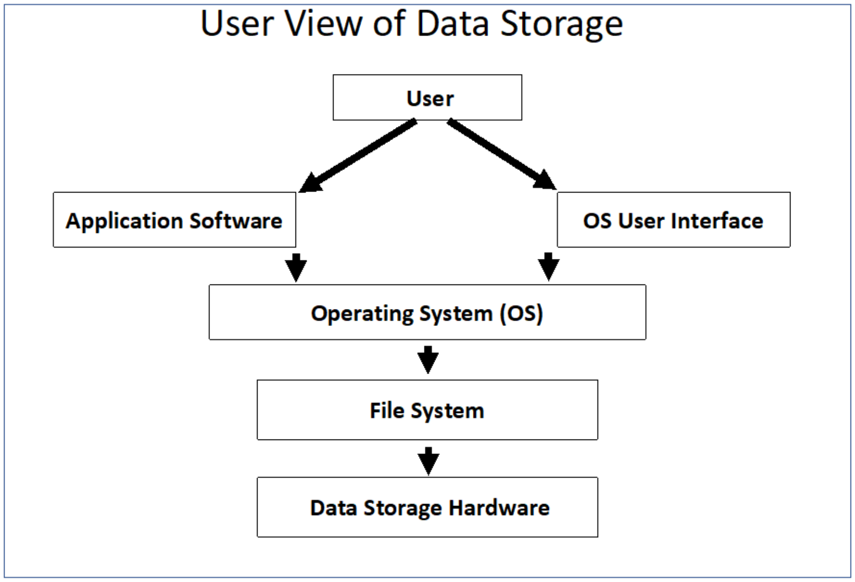

Figure 2–1 User View of Data Storage

Computer software is what makes a digital device useful. A user interacts with a digital device through an operating system and application software. The application software (apps) uses the environment provided by the operating system to do tasks requested by the user and any interaction (storing, changing, or deleting data) with a digital storage device happens through the file system. As illustrated in Fig. 2-1, the user views data stored on data storage hardware through these layers of software.

Computers automate many kinds of tasks, such as mathematical calculations, record keeping, machine tool control, etc. The task might be tedious, time consuming or just very detailed. The computer accomplishes an assigned task by following a list of instructions called a program, also known as computer software or computer code. The instructions describe the task in fine detail with steps such as “move this data item over there” and “add (multiply, subtract, divide) two data items” or “if these two data items are the same, skip the next instructions and continue running from another part of the program.” Most program instructions are a variation of these three types of instructions (move data, do math with data and if-condition-is-true-go-to alternative set of instructions). Since the program instructions are themselves just data elements, a program can produce a new program or modify an existing program or itself. This capability to generate a new program or modify an existing program gives computer software enormous flexibility in solving problems. These instructions are too detailed for a person to write a program quickly, so usually an easy to understand programming language is used that is then translated into the machine language of the computer.

Early in the development of digital computers the need for reliability was recognized and the means to ensure reliable data transfer was developed. Transferring data within a computer has to be extremely reliable because "in a digital computer … a single failure usually means the complete failure [that] . . . if it escapes detection then it invalidates all subsequent operations of the machine." (Hamming 1950) The reliability of copying data within a computer system is ensured by error correcting codes (ECC) incorporated in the actual representation of data. These codes protect a block of data from changes introduced by random noise that can change data as it is moved from one location to another (possibly from one memory location to another within the same device, or from a transmission from one physical location, e.g., satellite in orbit to a receiving station on Earth). These codes are implemented by computing a signature for a block of data to be protected and then transmitting the code (the ECC) with the protected block of data. A function is applied to the received data and compared to the transmitted ECC and the result indicates if the received data is error free or if an error occurred in transmission. The ECC may be designed to indicate which bits of the transmission has been modified and can therefore be corrected.

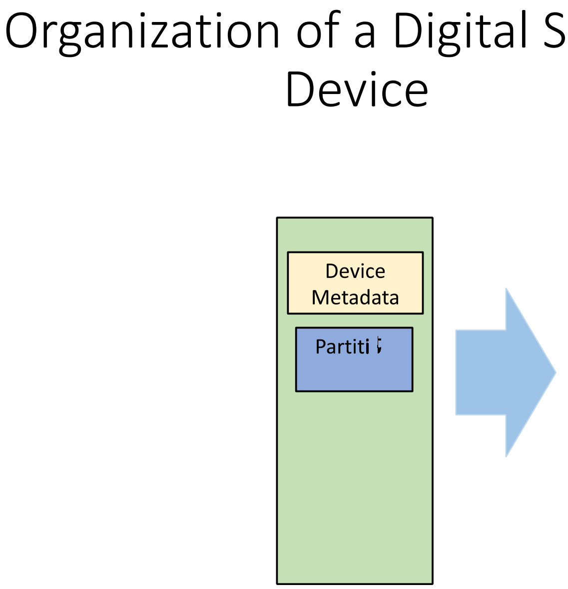

Figure 2–2 Storage Organization for a Single Device

Data storage has evolved with frequent changes to the details of how things are done, but the basic organization has stayed the same, as illustrated in Figure 2–2: a raw unformatted storage device contains a sequence of bits (or bytes) with no meaning, after formatting the device has a map of the layout of partitions on the device (the device metadata), and each partition is formatted with a selected file system including layout of file placement on the device in partition metadata, and any stored data. At each level metadata keeps descriptive

1. Protect data from modification. This is usually accomplished, by write blocking, i.e., monitoring access to a storage device for any data modifying attempts and suppressing the attempt at modification. This is discussed in Sec. 4.1.
2. Acquire digital data. This is accomplished by copying data to make an image file of the acquired digital data. Copying digital data accurately is based on established engineering techniques such as error detecting and correcting codes to ensure that data is copied accurately. This is discussed in Sec. 4.2.
3. Ensure integrity of acquired data. Cryptographic hashing is used to ensure that if acquired digital data is changed inadvertently or deliberately, the change can be detected. This is discussed in Sec. 4.3.
4. Recover deleted data. In some situations, recovery and reconstruction of deleted data makes it possible to bring back deleted files (in whole or in part) or internal records from within an application file. Recovering deleted data has several risks including missing data and conflating unrelated data. Any recovered item must be evaluated by the examiner for indications of problems. This is discussed in Sec. 4.4.
5. Navigate the acquired digital data. This is accomplished by unraveling, i.e., parsing the layout of the acquired data. This is best performed using a software tool. There is the risk that an incorrect implementation will not correctly interpret the structure of a particular file system, e.g., not showing all acquired active files. This is discussed in Sec. 4.5.
6. Identify and extract data artifacts. Items of interest are identified so they can be located and extracted by navigating the acquired data to find artifacts that meet criteria of interest such as, data that contains a specific text string, or association of an event with a specific date and time. This is discussed in Sec. 4.6.
7. Analyze. Examination of extracted artifacts can help develop a narrative or reconstruction of relevant events for inclusion in a final written report. This is discussed in Sec. 4.7.

The following subsections discuss more details of what is done in each of the seven steps. Sec. 4.9 discusses requirements for testing and validation of the techniques in each category.

4.1 Protecting Data by Write Blocking

Before any identified data can be copied (acquired) from a storage device, the device must be attached to a computer (This includes special purpose hardware devices that only makes a copy of the data on an attached device. These devices often have a built-in write blocker.) to access the data and make the copy. This can be a problem if the computer makes any changes to the device content before the copy operation takes place. There are several reasons this can happen, for example, as part of the startup process the operating system may examine several files and thereby change the file access times. The solution has been to introduce some sort of monitor on the connection to the device containing the data to be copied. This can be done with either software or hardware that monitors all commands sent to a device and suppresses any commands that might make a change to the device content(Lyle, Mead, and Rider 2007).

There are some situations where write blocking is not feasible, e.g., acquisition from a running system, acquisition of active memory or acquisition of data from a mobile device. In these situations, digital data is acquired imperfectly in that there are small differences between the actual data present on the device and the acquired data. Some examples:

- When acquiring a running system, other user activity may change file content during the acquisition.
- To acquire a mobile device, write blocking technology cannot usually be used; in addition a small tool might need to be loaded to the device to enable the acquisition. Of course, the tool overwrites the memory where it is loaded. See Sec. 4.2.2 for more details on acquiring a mobile device.
- For computer memory acquisition, as with mobile devices, write blocking technology does not apply and computer memory might require loading an acquisition tool into memory (overwriting a small portion of existing memory).

4.2 Acquisition of Digital Data

In the early days of digital forensics, the acquisition of digital data focused on acquiring the contents of computer hard drives, floppy disks, and CD-ROMs. The process was referred to as disk imaging. As digital storage devices have evolved it is more correct to refer to this process as digital data acquisition. This is the most fundamental task of digital forensics. The basic technique is to make a copy of the data to be examined. The copying of data is a straightforward reliable process ensured by error correcting codes (Hamming 1950) performed constantly by computers with safeguards to ensure that a complete and accurate copy is produced without modifying the original data.

The acquired data is placed into a container file that represents the acquired data. There are more than 30 different container file formats in use to contain digital data(Kim 2012). The most widely used image formats are raw images (dd format) and e01 (Expert Witness)(Vandeven 2014), but a number of other formats (usually specific to a tool vendor) are sometimes used. The need for a standard format has been recognized(Adelstein et al. 2006) and a standard, Advanced Forensic Format, has been proposed and is offered by some tool vendors(Garfinkel et al. 2006; Cohen, Garfinkel, and Schatz 2009; Cohen and Schatz 2010; Schatz 2015).

The procedures followed for an acquisition of digital data are vary slightly for different types of devices such as hard drives, flash drives, mobile devices, remote data, and other devices. In addition, significant digital data can sometimes be acquired from social media.

4.2.1 Storage Device (Hard Drive & Flash Drive) Acquisition

One of the first commercial digital forensic tools was SafeBack, a tool to create a forensic image of a hard drive(Pollitt 2010). Various procedures were developed to attach a hard drive to a computer that can run the imaging tool in conjunction with a write blocker so that a copy can be made without modification.

There are many special cases in data acquisition based on hardware or the type of acquisition. Different computer hardware models that have unique features requiring special consideration. In situations where only part of the source data is desired, techniques for selective acquisition and management of the fragmented data have been developed (Turner 2006), along with other projects to implement different selective acquisition tools and techniques (Novak, Grier, and Gonzalez 2019).

Whenever possible, acquisition should be done in conjunction with either a hardware write blocking device or a software write blocking tool to avoid modification of the original data.

4.2.2 Mobile Device Acquisition

For mobile device forensics, there are many considerations and options for acquiring and analyzing data from a mobile device (SWGDE 2016a, 2016b, 2019b):

- Logical acquisition: Extraction of a set of supported digital artifacts from the device. This is generally the easiest method.
- Selective acquisition: Extraction of a subset of supported digital artifacts from the device memory. This can be used to target specific data such as photos or contacts.
- File system acquisition: Extraction of the file system structure and content from the device. This allows acquisition of all data that is visible to the user.
- Physical acquisition: A copy of the device physical memory. These methods are often either destructive e.g., remove the memory chips (called chip-off) or risk damaging a device, e.g., use an industry standard for accessing memory chips (JTAG). These methods are the most complete and allows recovery of deleted data. (SWGDE 2019b)
- Universal Integrated Circuit Card (UICC), also called a Subscriber Identity Module (SIM Card) acquisition: Extraction of the supported artifacts from a UICC.

Each type of acquisition has advantages and limitations. Selection of an acquisition method depends on available tools and capabilities along with the make and model of device.

4.2.3 Remote Acquisition

Remote acquisition of data over a live network has several unique challenges not found when acquiring from a single device that can be taken offline for examination. These include getting access to the remote computer and the infeasibility of using a write blocker in the acquisition.

4.2.4 Other Device Acquisition

The embedding of digital devices into a variety of everyday items such as kitchen appliances, automobiles, home security systems and other everyday items has given to the rise of forensics of the Internet of Things (IoT). Acquiring digital data from such devices is challenging and often requires destructive disassembly to acquire the data.

4.2.5 Social Media Acquisition

In addition to the acquisition of raw binary data from digital devices, a wealth of digital data can be harvested from social media. This can include contact lists, images and locations visited.

4.3 Integrity Verification

After digital data has been acquired to an image file it needs to be verified that the acquired data has not been changed. Cryptographic hashing is used to detect inadvertent or deliberate changes. Cryptographic hashing is a robust technique used in multiple high security applications. NIST publishes hashing standards as part of its cryptography program (NIST 2015a, 2015b). The basic requirements for a cryptographic hashing algorithm are:

- Hash value can be computed quickly.
- It requires an unreasonable amount of computation to find two different files with the same hash value computed by the hash algorithm. This is defined as collision resistance.
- The original message cannot be recovered or reconstructed from the hash value.
- Any change to the original file brings about changes in the hash output value. On the average, a one-byte change to the original file causes about half of the bytes in the hash output to change.

4.4 Recovery of Deleted Data

Since most operating systems do not overwrite deleted data, this data can often be at least partially recovered. A complete file might or might not be reconstructed with the original content. In the situation where the storage device has had more than one owner, it is possible to recover data from previous owners, not just the current owner. This is one situation where apparently incriminating evidence can be found that has nothing to do with the current owner of the storage device.

There are three commonly used techniques for recovery of deleted data:

- Metadata-based file recovery (Fellows 2005). This technique exploits one design feature of file systems previously mentioned, that data is often not removed or overwritten when it is deleted. Just a notation is made to indicate that the data should not be seen and the storage space that it occupies can be reused. There might be file system metadata that can help locate where the deleted data was stored.
- File Carving (Richard, Roussev, and Marziale 2007). This technique is invoked when there might not be any file system metadata to guide recovery, in which case deleted files are identified by searching for data patterns at the beginning and end of a file that are unique to a given application file type.
- Deleted Record Recovery (Sanderson 2018). Some applications (e.g., databases such as MySQL or SQLite or the Windows Registry) keep records (a set of related data values) that might be marked as deleted but not overwritten and have the potential for recovery. A recovery tool examines the internal data layout of an application file to

identify deleted or updated data. Over time an application, such as SQLite, adds new records, updates existing records, and deletes some records. The application implementation can be exploited to identify and recover deleted data.

There are several considerations that have an impact on the quality of recovered data:

- If the deleted data has been overwritten or allocated to a new object, the deleted data cannot be recovered.
- Deleted data might be completely overwritten or only partially overwritten. It may not be possible to determine what data is original and what has been overwritten. The data presented as recovered might be mixed from several sources. The examiner can sometimes use context, metadata, and other clues to separate sources.
- Some file systems only preserve the location of the first storage block (FAT) when a file is deleted while other file systems (e.g., NTFS) preserve more block locations and other file systems (e.g., APFS) do not preserve any locations.
- Solid state drives might replace storage blocks marked by the OS (via a TRIM command) with a new block that only contains zero values. This could happen any time after the computer user has deleted a file, but before acquisition of the device contents.

Sometimes for a recovered object with content from multiple sources the location of the shift to another source can be identified. Because storage is allocated in fixed size blocks content can only shift to a new source with the next block. Block size is usually a multiple of 512 bytes. How this manifests in recovered content varies with the type of recovered object. In an image there will be a coherent image that becomes a different image from some point (usually the end of a block from one source and the beginning of a block from another source). In a document, there will be a shift of topic or some other unlikely shift, often within a sentence.

> **KEY TAKEAWAY #4.1:** When using techniques to recover deleted or hidden artifacts the examiner must determine the relevance of the recovered information as it may be incomplete or improperly merged with irrelevant information..

4.5 Parsing and Navigation

Once data has been acquired, the examiner needs to examine the acquired data. This is almost always done with some sort of interactive tool that presents the acquired data as seen in the original environment. The tool must recognize and interpret, i.e., parse the data structures and meta data embedded in the acquired data so that the tool can navigate the file system to display content. Development of a parser for a file system frequently requires reverse engineering of the file system (Nordvik et al. 2021; Nordvik et al. 2019) and then verification of the implementation. The common files systems are NTFS, ExFAT, FAT, ext4, HFS+, APFS, FAT and ExFAT. In addition, the tool needs to distinguish among the older file system versions, e.g., FAT comes in at least three major versions, twelve-bit FAT, sixteen-bit FAT and thirty-two-bit FAT (Carrier 2005). The Linux file system also comes in ext2 and ext3.

This publication is available free of charge from: https://doi.org/10.6028/NIST.IR.8354-draft

Forensic tools may not support all file systems that might be encountered, e.g., the ExFAT is sometimes not supported. When new file systems are introduced by computer vendors there is usually a lag time before the new file system is supported by forensic tools. If an unsupported file system is encountered, tools often treat the file system as unallocated space. Sometimes the support is incomplete or faulty at first. One of the most common failures is to not show all the object types. For example, NTFS has a feature to in effect have a collection of files under one name, this is called a primary data stream with multiple alternative data streams. A parsing tool might display only the primary data stream and ignore the alternative data streams. Some file systems have a feature called a link that allows more than one path through the directory tree to reach file content. There is potential for a defectively designed file system parser to produce incorrect results. The most likely impact is that the examiner would not see everything in the file system or see files in the wrong location. For example, if alternate data streams were not shown to the forensic tool user then content within an alternative data stream would be overlooked.

Forensic tools must be designed to allow for application file data organization so that files representing complex objects such as documents, databases, or graphic files can be displayed. A faulty implementation can display the wrong data, not just fail to acquire some data.

4.6 Identification and Extraction of Artifacts

An examiner often follows an iterative process to answer questions arising in an investigation. The main assembly of a narrative to describe the events of interest of an investigation or answering questions that arise during an investigation involves identifying, finding, and extracting relevant artifacts. A question of interest might prompt an examiner to select a specific artifact for examination. The examiner then tries to locate the selected artifact and then extract the artifact for examination. Some methods to accomplish this are:

- Keyword search locates files that contain a specific string. Some files containing instances of a searched for keyword might not be identified. Some situations where the keyword might not be found if the target string is:
  - in an encrypted file,
  - in a compressed file if the tool does not recognize compression and fails to expand the file and then search for the keyword,
  - represented with a text encoding method not searched for, e.g., only search UTF-8 but not search UTF-16, or
  - if text is in an application format that inserts formatting tags within words, e.g., inside the text of a word is a formatting tag to switch to bold font.
- Document retrieval locates files that discuss a specific topic.
- Metadata attribute matching locates files with metadata matching given criteria, e.g., file updated on a given date.
- Matching a given file property such as, a cryptographic hash of known contraband.

- Examining files known to contain specific content can identify needed information, e.g., contact list.
- Examining recovered files or recovered data records.

KEY TAKEAWAY #4.2: Searching tools have limitations based on the multiple ways that computers store information. Limitations include the type of files, types of encoding, and many other parameters. In general search tools are very effective at finding information, but there is a possibility that data will be missed because a tool does not have the capability to find it.

Useful digital artifacts can be extracted in a variety of ways. The simplest way is to know where a needed artifact can be found and just go get the artifact. There are lists of artifacts and how to locate the artifacts with some guidance on interpretation of the significance of the artifact (Magnet Forensics 2021a, 2021b).

4.6.1 Example Locating Artifact Indirectly

Sometimes locating the desired artifact requires a more indirect approach. For example, consider a question such as “has a given mobile phone ever been connected to a specific vehicle?” If the examiner knows where this model mobile phone keeps this vehicle identification number (VIN) information for vehicles that have been connected, they can just examine the file where this information is stored. However, the examiner might not know where this file is located. In that case, if the VIN of the vehicle is known then a keyword search for the VIN might verify a relationship between the vehicle and the mobile phone. If the VIN is unknown a pattern search can find files with strings in the same format as a VIN might locate a file with a list of vehicles that the mobile device has been connected to.

4.6.2 Locating Contraband

Cryptographic hashes can be used to identify known files from libraries of hashes of known files. A known file could be of known innocuous content, known contraband (e.g., child sexual abuse material), or of an ambiguous, dual-use, nature such as software tools often used for system administration that are also useful for system hacking. A tool likely to be found in a system administrator or computer science researcher’s tool kit might indicate further investigation is warranted if possessed by someone else. This is like finding lock picking tools. If in the possession of a locksmith, it is to be expected. This is a routine check to make when investigating hacking cases.

Identification of contraband can be accomplished is a variety of ways:

- Use a cryptographic hash of known contraband files to identify the presence of contraband. This method has a limitation in that the files must be identical. It does not identify files that are close, but not exact matches.
- Use hashes of file fragments to identify isolated pieces of contraband files. This is sometimes able to detect deleted contraband.
- Use an approximate matching technique (Bjelland, Franke, and Arnes 2014).

- Use DigitalDNA and similar methods to detect contraband images of children (Cifuentes, Orozco, and Villalba 2021; Franqueira et al. 2018; Hayes 2015; Ferraro, Casey, and McGrath 2005).
- Use string searching to look for words, numbers or other text associated with the targeted contraband.

4.6.3 Other Examples of Locating Possibly Relevant Artifacts

When a user interacts with a computer system, the computer generates artifacts which can be useful in an investigation. There are many possible artifacts, and more are being created with each new computer program. Some of the more common locations where artifacts useful in an investigation might be found are:

- Memory. It is possible to retrieve artifacts from memory such as currently running programs and connections.
- Windows Registry. The Windows operating system keeps track of user activity and changes to hardware and software.
- File system metadata. File systems keep track of when files were created, opened, and modified.
- Email. Email contains not only messages, but attachments and timestamps for when email was sent and received and for the path it took.
- Internet activity. This includes browsing history and downloads.

There are several efforts to catalog artifact types. Some examples are the Artifact Genome Project at University of New Haven. (See https://agp.newhaven.edu/about/start/) and the AXIOM Artifact Reference at Magnet Forensics (Magnet Forensics 2021a, 2021b). Additional projects are in progress.

There are many types of artifacts. For each type, an examiner needs to know what to look for and what it means. This can become quite complex. For example, the Windows Registry is designed for the Windows operating system to keep track of activity, specify configurations, and other system information. The meaning and significance of each artifact needs to be understood in context.

4.7 Analysis of Results

There are several important considerations to evaluate results. as well as items that are likely to be overlooked. For example:

- Does the examiner understand the meaning of each artifact relevant to an investigation?
- What steps have been taken to identify and mitigate bias that might have crept into the work?
- Were anti-forensics employed to thwart any investigation?
- Are there issues with system times that need to be handled?

- Can artifacts and activities be linked to a source such as the user of the machine or an external actor such as a hacker or malicious code? This may be referred to as attribution.

4.7.1 Analysis Tools

There are several classes of analysis tools that can help an examiner obtain a comprehensive understanding of the case data. Some examples include:

- A time-line tool can be used to put events into a temporal sequence to allow the examiner to have an overview of the relationships among events.
- Link analysis can look for relationships between entities in an investigation such as who is communicating with whom.
- Artificial Intelligence (AI) tools use a technique called *deep learning* that can be used to uncover unseen relationships between case elements or search through data to recognize relevant items. Some AI applications have been controversial because of the introduction of unexpected, unintentional bias. Examples include facial recognition software exhibiting poor or misleading results for racial minority subjects (Grother, Ngan, and Hanaoka 2019).

AI tools are powerful, but not perfect and should be used with caution due to unexpected behaviors. What comes out depends on the data set used to train the AI and may not be relevant to the data at hand, and any results could be misleading and should be verified or confirmed. As with other techniques, examiner must use caution and check that AI based finding are used in the appropriate context.

4.7.2 Anti-Forensics

There are many active measures that can be taken by a computer user to mislead an examiner. The simplest method is to delete incriminating files. However, deleted files might be recoverable and more effective anti-forensic techniques may be employed, such as using secure delete features of an operating system to overwrite deleted files. File wiping applications can also be employed to remove file remnants. If the file wiping application is found on a computer, it may indicate an effort to remove incriminating data.

Another common technique is to directly modify timestamps or set the system clock to the wrong time; this could be an attempt to set-up an alibi or just create confusion for the examiner.

There are widely used methods to muddy the water (Harris 2006) of an investigation. These include deleting information from system log files (to remove a record of an event), changing file MAC times (perhaps to create an alibi), and many others. In addition, there are methods to hide data using some properties of file systems (Huebner, Bem, and Wee 2006). There are techniques for data hiding specifically in Linux filesystems (Piper et al. 2006) and even in the system BIOS (Gershteyn et al. 2006).

Another technique for data hiding is *steganography*, hiding one set of data within another set of data. For example, media files such as image files or audio files often have higher resolution than is perceivable when a person views an image or listens to an audio file. One technique is to use the pixels of an image file to carry a hidden message. Only the left most bits make a difference in what is seen when viewing an image (the significant bits). For a 16-bit pixel the left most digits make a difference when looking at an image, so the right most digits (the least significant bits) can be used to contain something hidden. There are many techniques for detection (Rodriguez and Peterson 2007) of the hidden data. For example, in an unmodified picture the color values of the pixels should cluster around the dominate colors in the picture. On the other hand, if the distribution of pixel values is uniform, i.e., flat, then something might be hidden within the picture, but not observable when viewing the picture.

> **KEY TAKEAWAY #4.3:** If someone has taken steps to change information in digital evidence to mislead an examiner, it may be difficult to detect the changes. Depending on the sophistication of the manipulation, identification of the changes relies on the skill of the examiner.

4.8 Verification of Techniques and Validation of Tools

When discussing tool testing, the forensic community needs to be aware of the usual meaning of the terms "validation" and "verification" within software engineering. In colloquial usage the terms verification and validation mean essentially the same thing: checking to see if something is correct. But, in a technical context there is an important difference thus leading to the potential for confusion.

For a forensic technique or method to be considered validated it should be shown to be fit for purpose otherwise defined as "the process of providing objective evidence that the method is good enough to do the job required by the end user". Validation can give a false indication of "fitness for purpose" that becomes apparent later.

Verification, on the other hand, is the demonstration that the implementation of the method correctly follows the tool design. It does not intend to show that the design is correct, but it may show that the design is incorrect.

Some examples to contrast verification and validation include:

- Consider building a tower. An engineer submits a design for a tower, it is reviewed and found to be like the design of other towers and approved for construction. That seemed good enough. A contractor is hired, and work begins. At each step the contractor's work is checked and found to agree with the design. The tower is finished, but after completion the new tower begins to lean to one side. The design is wrong, i.e., not fit for purpose at the building site. On deeper examination the soil conditions under one side of the tower are too weak to support the weight of the tower. The design failed to account for this condition and should have been rejected. Another way to look at this is the design requirements were incomplete and something was missed. The builders verified that the construction (implementation) conformed to the tower design, but since the tower design was not fit for building on the weak soil, the tower failed. This is one common way that the wrong tool gets built.

- Consider the scenario of selecting an algorithm for detecting if a digital object has changed (say, to verify image file integrity). This is an example of using validation to select an algorithm to implement that is fit for purpose. There are several candidates, e.g., CRC16, CRC32, MD4, MD5, SHA-1, SHA-2. The CRC algorithms have been used for decades to check if a block of data has been transmitted without an error and was used in early imaging tools to verify image integrity (Peterson and Brown 1961). The CRC is fit for detecting changes caused by random noise, however a malicious actor can easily modify a file in such a way that the CRC does not change. (This is called creating a hash collision.) Some additional requirements are needed for a hash algorithm to be fit for purpose in a forensic context:

  - Can be computed quickly.
  - It requires an unreasonable amount of computation to find two different files with the same hash value computed by the hash algorithm. This is defined as collision resistance.
  - Original message cannot be recovered or reconstructed from the hash value.
  - Any change to the original brings about changes in the hash output value.

  CRC does not meet all these criteria because CRC is not collision resistant. MD5 and SHA-1 were considered to meet these criteria until hash collision production algorithms were created for MD5 (Wang and Yu 2005) and SHA-1 (Wang, Yin, and Yu 2005). The work of Wang created concern about the use of MD5 and SHA-1 for digital forensic applications but, these collision creation algorithms are for a restricted context that is not relevant for digital forensic applications. (Thompson 2005)

  The SHA-2 and SHA-3 algorithms have been tested to meet these requirements and do not need to be further studied (NIST 2015a, 2015b). However, a tool that computes either SHA-3 or SHA-2 needs to be verified to ensure that the implementation correctly computes the hash value.

There have been several papers published on validation of digital forensics methods (Regulator 2020; Arshad, Jantan, and Abiodun 2018; Beckett and Slay 2007; Brunty 2011; Casey 2011a; Craiger et al. 2006; Guo, Slay, and Beckett 2009; Horsman 2018; Horsman 2019; Marshall and Paige 2018; Risinger 2018; SWGDE 2014; Wilsdon and Slay 2006). Some of these papers seem to confuse validation of a method and verification of a software tool and try to fold the two activities together instead of keeping them separate. The guidance from the UK Forensic Science Regulator (Regulator 2020) seems the most clear and includes consideration of risk assessment of the method, documentation of acceptance criteria and possible outcomes.

The general validation and verification for a given version of a tool can be done once. It does not need to be performed by every lab. The validation of the technique needs to be repeated as the implemented algorithm changes to address changes in method to solve the intended forensic task. The implemented tool needs to be studied whenever the tool is changed or related technology changes. Each lab should ensure that personnel understand the basic capabilities and limitations of a tool, especially the relationship between the tool and the fast-changing IT environment.

4.9 Requirements for Testing Forensic Techniques

This section discusses digital forensic methods from the perspective of validation and verification. There are several approaches to show the reliability of a technique or that it is fit for purpose.

- An analysis or inspection of the algorithm to see if the algorithm is sound and to identify potential limitations.
- The general intent of an algorithm may be known, but the details of the algorithm design may be unknown. In this case, a direct analysis of the actual algorithm is not feasible, but an implementation can be tested to evaluate conformance to the intent of the algorithm.
- Part of the validation process should include an analysis of what can go wrong. This gives guidance for prioritizing and constructing test cases to evaluate an implementation.
- Implementations need to be tested to look for mistakes in the implementation and anomalies that occur within a given run time environment (hardware and operating system version).
- Testing is sometimes done to show that a technique can work and at other times to identify conditions when it does not work.

4.10 Errors and Testing

This section discusses the meaning of error, error rates and tool testing.

4.10.1 Error Rates

Some forensic disciplines use an error rate to describe the chance of false positives, false negatives, or otherwise inaccurate results when determining whether two samples come from the same source. But in digital forensics, there are fundamental differences in many processes that can make trying to use statistical error rates inappropriate or misleading.

The key point to keep in mind is the difference between random errors and systematic errors. Random errors are characterized by error rates because they are based in natural processes and the inability to perfectly measure them. Systematic errors, in contrast, are caused by many different factors. In computer software, for example, an imperfect implementation can produce an incorrect result every time a particular condition, usually unknown, is met. Digital forensics – being based on computer science – is far more prone to systematic than random errors.

Digital forensics includes multiple tasks which, in turn, use multiple types of automated tools. For each digital evidence forensic tool, there is an underlying algorithm and an implementation of the algorithm (how the task is done in software by a tool). There can be different errors and error rates with both the algorithm and the implementation. For example, hash algorithms used to determine if two files are identical have an inherent false positive rate, but the rate is so small as to be essentially zero (NIST 2015c, 2015a, 2015b).

The classic concept of *error rate* as found in statistical hypothesis testing should apply to the intended algorithm of a statistical technique but does not usually apply to evaluating reliability of digital forensic tools (Lyle 2010; SWGDE 2018b). This is mostly due to the nature of the two activities. In hypothesis testing there is a simple binary decision. Something like “do two samples come from the same source with a given probability of a correct decision?” A digital forensics technique implementation in software may have multiple ways to fail with different risks associated with each failure mode ranging from significant to trivial. An error such as mislabeling a phone number as an email address could result in needed information not being found or be trivial since an examiner could easily correct this.

Once an algorithm is implemented in software, in addition to the inherent error rate of the algorithm, the implementation can introduce systematic errors that are not statistical in nature. Software errors manifest when some condition is present either in the data or in the execution environment. It is often misleading to try to characterize software errors in a statistical manner since such errors are not the result of variations in measurement or sampling. For example, the hashing software could be poorly written and may produce the same hash every time an input file name starts with the symbol “$.” We might be tempted to collect data on files with this property and compute an error rate based the observed frequency of this property. This works fine if we know what characteristic triggers the incorrect tool behavior. The problem with this approach is that we most likely do not know which characteristics trigger the incorrect tool behavior and do not know what characteristics to measure. There is also the problem of collecting enough data to compute meaningful statistics. In addition, the triggering characteristics may become irrelevant as technology evolves.

Another problem is that the properties and characteristics of digital data changes with the software environment as the technology evolves over time and an error rate valid at one point in time might not apply at any other point in time.

> **KEY TAKEAWAY #4.4:** Digital processes tend to have systematic rather than random errors. Therefore, an error mitigation analysis provides more information and is the correct way to manage uncertainty. Asking for an error rate is only useful where there are random errors.
>
> **KEY TAKEAWAY #4.5:** When error rates are provided, it is important for the user to understand the context of the numbers. Errors in computer science techniques tend to be so small as to be negligible. For some forensic techniques, the error rates may vary significantly based on attributes of the technology and usage patterns.

4.10.2 Observed Errors

The primary types of errors found in digital evidence forensic tool implementations are:

- Incompleteness: All the relevant information has not been acquired or found by the tool. For example, an acquisition might be incomplete, or a search does not identify all existing relevant artifacts.

- Inaccuracy: The tool does not report accurate information. Specifically, the tool should not report artifacts that do not exist, should not group together unrelated items, and should not alter data in a way that changes the meaning. Assessment of accuracy in digital evidence forensic tool implementations can be categorized as follows:

    o Existence: Do all artifacts reported as present exist? For example, a faulty tool might add data that was not present in the original.

    o Alteration: Does a forensic tool alter data in a way that changes its meaning, such as updating an existing date-time stamp (e.g., associated with a file or e-mail message) to the current date?

    o Association: For every set of items identified by a given tool, is each item truly a part of that set? A faulty tool might incorrectly associate information pertaining to one item with a different, unrelated item. For instance, a tool might interpret a web browser history file incorrectly and report that a web search on “how to murder your wife” was executed 75 times when in fact it was only executed once while “history of Rome” (the next item in the history file) was executed 75 times, erroneously associating the count for the second search with the first search. There are many techniques to detect such errors such as peer review of the tool.

    o Corruption: Does the forensic tool detect and compensate for missing and corrupted data? Missing or corrupt data can arise from many sources, such as bad sectors encountered during acquisition or incomplete deleted file recovery or file carving. For example, a missing piece of data from an incomplete carving of the above web history file could also produce the same incorrect association.

- Misinterpretation: The results have been incorrectly understood. Misunderstandings of what certain information means can result from a lack of understanding of the underlying data or from ambiguities in the way forensic tools present information (SWGDE 2018b).

4.10.3 Software Testing (Tool Verification)

Doing software testing is like doing science. Just as Popper's (Popper 1959) description of a scientific theory includes the idea that you cannot prove a theory is true, you can only disprove a theory or at least identify conditions where the theory does not apply. In keeping with the previous discussion of validation and verification is Sec 4.8 you cannot prove that a software program is correct by testing, just identify conditions where it fails.

Other articles about digital forensic tool testing (Anobah, Saleem, and Popov 2014; Beckett and Slay 2007; Brunty 2011; Casey 2011a; Craiger et al. 2006; Cusack and Homewood 2013; Cusack and Liang 2011; Flandrin et al. 2014; Garfinkel 2012; Garfinkel et al. 2009; Glisson, Storer, and Buchanan-Wollaston 2013; Grajeda, Breitinger, and Baggili 2017; Guo, Slay, and Beckett 2009; Guttman, Lyle, and Ayers 2011; Hibshi, Vidas, and Cranor 2011; Horsman 2018; Horsman 2019; James, Lopez-Fernandez, and Gladyhsev 2014; Marshall and Paige 2018; McKemmish 2008; SWGDE 2014, 2018b; Yates and Chi 2011) discuss various aspects of testing digital forensic tools.

> **KEY TAKEAWAY #4.6:** It is not feasible to test all combinations of tools and digital evidence sources.

4.10.4 NIST Tool Testing Results

NIST/CFTT (National Institute of Standards and Technology 2019a, 2020) develops tool specifications and test plans for testing various types of forensic tools, such as:

- Data Acquisition (National Institute of Standards and Technology 2005a, 2004a),
- Write Blocking (National Institute of Standards and Technology 2005b, 2004b, 2003),
- Media Preparation (National Institute of Standards and Technology 2009b, 2009c),
- File Carving (National Institute of Standards and Technology 2014; National Institute of Justice and National Institute of Standards and Technology 2014),
- Metadata based Deleted File Recovery (National Institute of Standards and Technology 2009a),
- Windows Registry (National Institute of Standards and Technology 2018b, 2018c),
- Text String Searching(National Institute of Standards and Technology 2008, 2018a),
- SQLite Deleted Record Recovery (National Institute of Standards and Technology 2021b) and
- Mobile Devices (National Institute of Standards and Technology 2019b).

DHS and NIJ have published forensic tool test reports for a variety of tool types.

NIST/CFTT has also published papers describing the testing techniques used by CFTT for write blocking (Lyle 2006a; Lyle, Mead, and Rider 2007), general disk imaging (Lyle 2002), and imaging hard drives with faulty sectors(Lyle and Wozar 2007).

There are NIST/CFTT digital forensic tool test reports for:

- Disk Imaging and secondary storage acquisition tools (Department of Homeland Security and National Institute of Standards and Technology 2016a, 2013a; National Institute of Justice and National Institute of Standards and Technology 2009a, 2008a;

Department of Homeland Security and National Institute of Standards and Technology 2013b; National Institute of Justice and National Institute of Standards and Technology 2008b, 2008c, 2009b, 2013a, 2008d, 2011a; Department of Homeland Security and National Institute of Standards and Technology 2013c, 2013d, 2014g, 2014h; National Institute of Justice and National Institute of Standards and Technology 2011b; Department of Homeland Security and National Institute of Standards and Technology 2014i, 2013e, 2016b, 2016c, 2016d, 2016e, 2016f, 2016g, 2016h, 2016i),

- Write Blocking tools (Department of Homeland Security and National Institute of Standards and Technology 2020a; National Institute of Justice and National Institute of Standards and Technology 2006a, 2006b, 2006c, 2007a, 2007b, 2006d; Department of Homeland Security and National Institute of Standards and Technology 2018a; National Institute of Justice and National Institute of Standards and Technology 2018; Department of Homeland Security and National Institute of Standards and Technology 2018b, 2018c; National Institute of Justice and National Institute of Standards and Technology 2006e; Department of Homeland Security and National Institute of Standards and Technology 2018d; National Institute of Justice and National Institute of Standards and Technology 2006f, 2009c; Department of Homeland Security and National Institute of Standards and Technology 2009, 2018e, 2018f; National Institute of Justice and National Institute of Standards and Technology 2007c; Department of Homeland Security and National Institute of Standards and Technology 2018g, 2018h, 2018i; National Institute of Justice and National Institute of Standards and Technology 2007d, 2007e; Department of Homeland Security and National Institute of Standards and Technology 2018j, 2018k, 2018l, 2018m; National Institute of Justice and National Institute of Standards and Technology 2007f, 2007g, 2008f, 2008g; Department of Homeland Security and National Institute of Standards and Technology 2018n, 2018o; National Institute of Justice and National Institute of Standards and Technology 2006g, 2006h, 2006i, 2006j, 2005a, 2005b, 2005c, 2004a, 2004b, 2004c, 2004d, 2008k, 2008l),
- File Carving tools (Department of Homeland Security and National Institute of Standards and Technology 2014j, 2014k, 2014l, 2014m, 2014n, 2014o, 2014p, 2014q, 2014r, 2014s, 2015h, 2014x, 2014y, 2014z, 2014aa, 2014ab, 2014ac),
- Metadata based Deleted File Recovery tools (Department of Homeland Security and National Institute of Standards and Technology 2014a, 2014b, 2014c, 2014d, 2014e, 2014f),
- Windows Registry tools, (Department of Homeland Security and National Institute of Standards and Technology 2019e, 2019f),
- Mobile Devices tools (Department of Homeland Security and National Institute of Standards and Technology 2019g, 2018aa; National Institute of Justice and National Institute of Standards and Technology 2010e, 2008j; Department of Homeland Security and National Institute of Standards and Technology 2014w, 2014v, 2015g, 2014u, 2015f, 2014t, 2015e, 2013f, 2015d, 2018z; National Institute of Justice and National Institute of Standards and Technology 2010d; Department of Homeland Security and National Institute of Standards and Technology 2018y, 2017h, 2016r, 2018x, 2018t, 2017g; National Institute of Justice and National Institute of Standards and Technology 2010c; Department of Homeland Security and National Institute of

Standards and Technology 2017i, 2016q, 2019d, 2019c, 2016p, 2017f, 2016o, 2015c; National Institute of Justice and National Institute of Standards and Technology 2013e; Department of Homeland Security and National Institute of Standards and Technology 2015b, 2018w, 2016n, 2018v; National Institute of Justice and National Institute of Standards and Technology 2011c; Department of Homeland Security and National Institute of Standards and Technology 2018u, 2017e, 2016m, 2016l, 2015a, 2017d, 2017c; National Institute of Justice and National Institute of Standards and Technology 2012b, 2013d, 2008i; Department of Homeland Security and National Institute of Standards and Technology 2018s; National Institute of Justice and National Institute of Standards and Technology 2013c; Department of Homeland Security and National Institute of Standards and Technology 2018r; National Institute of Justice and National Institute of Standards and Technology 2010b, 2008h; Department of Homeland Security and National Institute of standards and Technology 2019b, 2018q, 2017b, 2018p, 2016k; National Institute of Justice and National Institute of Standards and Technology 2013b, 2012a; Department of Homeland Security and National Institute of Standards and Technology 2017a, 2016j; National Institute of Justice and National Institute of Standards and Technology 2010a; Department of Homeland Security and National Institute of Standards and Technology 2019a), and

- Key Word String Searching tools (Department of Homeland Security and National Institute of Standards and Technology 2018ab, 2019h, 2020c, 2020d, 2020e, 2020b).

NIST does not use the terms validation and verification and calls the CFTT “tool testing” to avoid confusion about the terms. CFTT creates a requirements specification based on what tool vendors implement but considers it as descriptive of what the available tools do rather than a prescriptive specification of what they must do. CFTT tests for correct implementation against the CFTT created specification.

The NIST/CFTT project has been testing data acquisition of storage devices such as hard disk drives and removable *flash drives* since 2002. Volatile memory acquisition was not within the scope of these tests. In general, the tools performed well with minor behaviors that needed to be kept in mind. Some examples include:

- Data acquisition might stop before all data on a device had been acquired. This was usually not a problem because the omitted data was not being offered to the computer user by the operating system for use. Typically, the operating system would group the basic unit of space on a hard drive (the 512-byte sector) into larger fixed size blocks with some space left over. The left-over space could vary from a single sector (National Institute of Justice and National Institute of Standards and Technology 2002a) to over 5,000 sectors(National Institute of Justice and National Institute of Standards and Technology 2002b).
- The size, number of sectors, of a digital device can be reported several ways (BIOS size, visible size ignoring hidden sectors, visible sectors + size of host protected area (HPA), and visible sectors + HPA + device configuration overlay (DCO)). These four sizes can all be different. A forensic tool may choose any size to use as the size to acquire. Using BIOS reported size is obsolete now but in the late 1990s and early

2000s this was the preferred method since in allowed a BIOS based software write blocker to be invoked to protect the computer hard drive from modification. After development of hardware write blockers, direct acquisition without risk of modification of the data became possible.

- As hard drives age some sectors may fail and become unusable. Sometimes during data acquisition, a sector may be unreadable and is reported as a bad sector. CFTT was able to develop and use techniques for testing tool behavior on encountering bad sectors(Lyle and Wozar 2007). Tools often omit readable sectors surrounding a bad sector, usually related to how the file system blocks disk sectors for the interface (USB, SATA, Firewire, etc.) used to access the hard drive.
- EnCase Version 4.22a (National Institute of Justice and National Institute of Standards and Technology 2008e) test results are an example of a tool having a small problem that sounds worse than it actually is. Seven sectors are imaged incorrectly, and one sector is omitted from the image. When the imaging tool FTK Imager 2.5.3.14 was tested on the same data set (National Institute of Justice and National Institute of Standards and Technology 2008d), the tool omitted all eight sectors. The reason for omitting the eight sectors is that the NTFS file system does not use these sectors to store any user data and should not contain any evidence. EnCase was trying to omit the data but made a mistake on when to stop writing the image file. From the EnCase Test Report (National Institute of Justice and National Institute of Standards and Technology 2008e):
- “If a logical acquisition is made of an NTFS partition, a small number (seven in the executed test) appear in the image file twice, replacing other sectors (DA–07–NTFS).
- If a logical acquisition is made of an NTFS partition, the last physical sector of the partition is not acquired (DA–07–NTFS). “

The most serious failure ever observed was reported in 2003 for SafeBack Version 2.0 (National Institute of Justice and National Institute of Standards and Technology 2003). In one tool configuration the acquired data of a SCSI drive was not the expected content and incomplete. The tool gave no indication that there was a problem. A direct SCSI disk copy, using the Advanced SCSI Programming Interface (ASPI) driver for the SCSI adapter, copied only 2,097,270 sectors from a source disk with 17,921,835 sectors to an equal-size disk, leaving 15,824,565 sectors of the destination disk unchanged. SafeBack gave no indication of any problems and indicated a successful copy. An examiner might realize that the acquisition was corrupt but would not be able to continue the analysis. The vendor fixed the problem within two days of being notified.

- NIST has developed several techniques for testing both software and hardware write blockers(Lyle 2006b) that are widely used by digital forensic labs to verify operation of write blockers. The NIST testing uses several techniques to generate traffic to see if a write blocker intercepts commands or lets them pass. Most write blockers on the market were able to block commands that would have changed a drive. The few exceptions were for uncommon commands or, in one case, where a vendor was unaware of a change to a chipset (that was quickly fixed). An analysis of the testing performed by NIST showed that write blocking is an effective technique. Special purpose software write-blockers have been designed for situations such as virtual machines(Tobin, Le-Khac, and Kechadi 2016).

In general, the NIST test results often revealed minor anomalies. Typical results include:

- Sometimes acquisition tools miss data located in usually unused areas at the end of the device.
- Except for one model device, hardware write-block devices always blocked write commands. The firmware for the one blocker that allowed write commands was quickly fixed by the vendor.
- Deleted file recovery and file carving results need to be carefully examined and might contain missing data or data mixed from multiple sources. Any conclusions drawn from recovered files must be carefully evaluated.
- Mobile device results often have minor anomalies such as truncated strings and unsupported device models.
- Text string searching often misses some strings searched for, especially for text encoded in Unicode 16-bit schemes.

While the analysis of write blockers performed by NIST and documented through test reports by NIJ and DHS shows that the technique has been demonstrated to be effective (see Sec. 4.10.4 on NIST Tool Testing), implementation and usage are critical. The tool must match the technology it is intended to be used with and be set up and used correctly. For example, some write blocking devices use the same hardware for a write blocker as for a bridge used to switch between interfaces, if by mistake during a firmware upgrade the firmware for a bridge is uploaded to a write blocker then the device will no longer prevent changes to an attached storage device. This scenario is a good example of why it is a best practice to retest forensic tools after an upgrade.

The analysis of string searching performed by NIST and documented through test reports by NIJ and DHS shows that the technique has been demonstrated to be effective at finding items. The test results demonstrated some systematic missing of text, e.g., Unicode 16-bit representation for languages with diacritical marks, but no false positives were observed. Sometimes two tools gave different search results for pre-defined search targets if the tools defined the targets differently. Implementation, usage, and analysis of results based on an understanding of the capabilities and limitations of the technique are critical.

> **KEY TAKEAWAY #4.7:** Extensive tool testing of over 250 widely used digital forensic tools showed that most tools can perform their intended functions with only minor anomalies.

4.10.5 NIST Test Data Sets for Tool Testing

The CFReDS (**Computer Forensic Reference Data Sets**) project at NIST is a repository of digital storage device images. Examiners can use CFReDS in several ways including validating the software tools used in their investigations, checking that equipment is working properly, training examiners, and practicing using forensic tools. Some images are produced by NIST, often from the CFTT (tool testing) project, and some are contributed by other organizations. The CFTT project posted its first document for public comment in March

2001, a specification for disk imaging including requirements and test assertions. CFTT submitted the first forensic tool test report, Red Hat GNU fileutils 4.0.36 dd, based on the final version of the specification to the National Institute of Justice for publication in August of 2002.

The CFTT project approached forensic tool testing with the conformance testing model, often used to certify that a product conforms to a specific standard. The conformance testing model verifies that a product performs according to its specified standards. Because there were no published standards for forensic tools, CFTT took on the task of writing specifications for the tool functions they were tasked with testing. The tool specification included definitions of the function to be tested, a list of requirements the tool should meet, a list of test assertions to specify conformance to requirements and a set of test cases to be run. CFTT also created software to create test data, test data sets, software to evaluate test case results and procedures to follow when executing test cases. A formal test plan, test report and code review were published (Gavrila and Fong 2004) for the test support software used with the disk imaging tool tests. No significant anomalies were found.

Before CFTT creates test data sets, CFTT first needs a tool function specification and a test plan with test cases. The steps for creating a data set are:

1. With the help of law enforcement representatives that advise the CFTT project, a forensic tool function is identified for testing along with a list of candidate software or hardware tools.

2. CFTT examines the selected tools and produces two lists of tool features offered: core features that are offered by all tools and optional features that are offered by some tools. For example, all imaging tools are tested for acquiring an entire hard drive (a core feature), but only some tools support imaging of a single partition (an optional feature).

3. For each feature a list of requirements is created to specify what the feature is supposed to do.

4. A list of parameters that could impact tool behavior is created to help specify test cases. These may be tool settings that did not fit as a tool feature or run time environment factors such as type of file system to be examined.

5. Test cases are created based on test parameters.

After test cases are developed, test data sets must be created. Testing is all about getting a tool to fail. The more unique opportunities a tool is given to fail, the more confidence in the correctness of tool results when a tool is used for a real investigation.

CFTT uses two data set creation approaches: static and on-the-fly. The static data sets are created in such a way that it is convenient to make a disk image of the data and then the disk image can be imported into a forensic tool for examination. Creation of a test data set is often a combination of scripts and custom tools. The on-the-fly data sets are usually for some type of function that interacts directly with a device. Each test case has a set of procedures for preparing a device or test image for the test. In addition, there may be a set of custom tools to help evaluate the result for both test data creation approaches.

The on-the-fly testing usually follows this protocol:

1. Populate a device with test data designed to reveal anomalies by using custom tools and scripted user actions to set-up the device.

2. Run the tool under test.

3. Examine the results, using custom tools to help evaluate the results. If the test case does not modify the device (e.g., hard drive), the device can be reused for testing another tool. Precautions are taken to back up the device in case it is modified during running the test case and needs to be restored.

Tool functions at CFTT that use the on-the-fly approach include disk imaging, write blocking, forensic media preparation (drive wiping) and mobile device testing. Procedures for setting up test devices are posted on the CFTT web site along with a description of notable features of the data setup. For example, for disk imaging each sector of the device is given unique content that includes the LBA address of the sector. This allows easy diagnosis of misplaced sectors if a tool places an imaged sector in the wrong location in the image or places a given sector in the image more than once.

Sometimes writing the procedures to follow are challenging because an unusual condition would require additional steps to finish the procedure. For example, when testing disk wiping, for a tool that allows using the built-in security erase command it must be ensured that (1) the disk drive supports the security erase command and (2) the test computer BIOS does not disable the security feature set.

The static test data sets are usually provided as a set of small disk images for testing each tool function. The tool functions that use a static data set are file carving for graphic and video files, Metadata based deleted file recovery and string searching.

For some test data sets additional tools need to be available to evaluate the test results. For example, it is often suggested to hash (MD5 or SHA-1) the result of deleted file carving or metadata based deleted file recovery to see if the file is correctly recovered. However, this only gives a yes or no answer and does not measure if the recovered file is a total failure or a near miss. Rather than use an all or nothing measure it is more useful to measure the quality of the recovered files. For file carving CFTT uses two measures. First a visual evaluation to see if the returned file can be viewed. Second an examination of the data returned to see how much of the original file is returned, how much data is omitted and how much is not from the original file. Both measures were often revealing of important aspects of the tool that just one measure did not show. One tool, for example, that was given a certain graphic file format returned an image file that did not produce a viewable image when displayed, but an examination of the returned data file revealed that the tool returned all the data except for the last block.

Some data sets do not need an evaluation tool. For example, each string search test case has a list of expected string instances that should be returned. Evaluating each test case is just a matter of comparing the list of expected hits to the actual hits returned. With the string search test cases several test assertions are tested at the same time. For example, the string "DireWolf" appears 15 times in the test data set. Each instance of the target string is followed by a unique ID number so that an examination of a hit context confirms the actual instance returned. Some of the combined test assertions that can be tested:

- Find a string in an active file for each of 7 file systems (FAT, ExFat, NTFS, ext4, HFS+ ignore case, HFS+ case sensitive, & APFS).
- Find a string in a deleted file for each of 7 file systems.
- Find a string in unallocated space.

To give opportunities for testing to fail the CFTT data set also includes the strings "WOLF" (all caps), "Wolf" (mixed case), "wolf" (all lower case) and "WereWolf". These strings support several searching test assertions with enough strings that are almost matching to trigger some likely errors:

- Search for "wolf" with match case might fail by hitting "WOLF" or "Wolf" or "DireWolf."
- Search for "Wolf" as a whole word might fail by returning "WereWolf."

For testing UNICODE (UTF-8, UTF-16-BE & UTF-16-LE) 43 string instances are needed. CFTT also considered types of character sets and decided to include Latin based character sets with diacritic marks: Spanish, French, German and Italian; A non-Latin character set: Russian; a right-to-left presentation: Arabic; distinct Asian character sets: Chinese, Korean, and Japanese Kana. There were many other possibilities, but this covers most character set forms likely to be found.

Some considerations for constructing the CFReDS data sets include the following:

- It is often suggested that real-world data sets should be used. This has several advantages:
  - The data set is similar to the data that the forensic tool would encounter in investigative use.
  - The data set includes a large amount of noise, i.e., data that is not relevant to the investigation, that the tool must show that it can process successfully.
  - The actions of a computer user in the real-world are in a random order and produce a variety of layouts so that the data set may include a situation that

  would cause the tool to fail. However, a constructed data set might not consider or include such a data layout.

- A real-world data set also has disadvantages:

  - Data set ground truth is difficult to determine. The large amount of noise in the data is one factor in the difficulty.
  - Significant effort is required to obtain enough data sets so that there is coverage of all features included in the test plan.
  - Creating the data set takes significant effort.
  - Executing the test plan is time consuming when invoking the tool under test on several large image files.
  - The data sets are intended for sharing over the internet and large image files take significant time to download.

- Constructed data sets are able to address the disadvantages:

  - It is easy to create data sets with known ground truth.
  - A constructed data set can be focused on the features included in the test plan.
  - A created data set can be kept small.
  - Small data sets take much less time for a tool to scan and analyze.
  - Small data sets are quicker to download.

5 Conclusions

Digital investigation techniques are based on established computer science methods and are reliable when used with knowledge of how a tool functions and its limitations. The complexity and rapid change of the field do, however, introduce the possibility for incomplete analysis or for misunderstanding of the meaning of artifacts.

Practitioners and stakeholders need to be aware of the following limitations with digital investigations:

- As with any crime scene not all evidence may be discovered.
- When recovering deleted files, the results may include extraneous material.
- Examiners need to understand the meaning and significance of digital artifacts retrieved as they can change over time.

To reiterate, this analysis only addressed core digital evidence processes. It did not include several closely related areas such as network forensics, multimedia (audio, images, video) forensics, and hacking and malware analysis.

While developing this report, we encountered many areas that need further research and improved processes, including:

- Better sharing of forensic knowledge including new and changed artifacts, new techniques, tool limitations and workarounds, and other forensic insight. There are multiple blogs and other informal mechanisms, but a more structured approach would benefit the community.
- Better approaches to testing of forensic tools. Currently, digital forensics labs are each testing the same tools causing redundant work. A more structured approach could increase efficiency.
- Better sharing of forensic reference data. High quality data is expensive to produce but is vital for tool testing, training and education, and research and development of new tools and techniques.
- Better analysis of how digital evidence is used and whether there have been incorrect or misleading conclusions. Having this information centrally collected would benefit the field.
- Better understanding of bias. Because of the nature of most digital evidence case work, forensic examiners are exposed to knowledge about people involved in the case, such as seeing their photos and reading their text messages. In addition, the forensic examiner may need to interact with an investigator.

The overall finding of this report is that digital evidence examination rests on a firm foundation based in computer science. Several of the techniques had already been extensively studied and documented in the peer-reviewed literature. Others are documented more informally through community discussion forums. The application of these computer science techniques to digital investigations is sound, only limited by the difficulties of keeping up with the complexity and rapid pace of change in IT.

References

Adelstein, F., B. Carrier, E. Casey, S. L. Garfinkel, C. Hosmer, J. Kornblum, J. Lyle, M. Rogers, P. Turner, and CDESFW Grp. 2006. 'Standardizing digital evidence storage', *Communications of the Acm*, 49: 67-68.

Anobah, Maxwell, Shahzad Saleem, and Oliver Popov. 2014. 'Testing framework for mobile device forensics tools', *Journal of Digital Forensics, Security and Law*, 9: 221-34.

Aquilina, James M., Eoghan Casey, and Cameron H. Malin. 2008. *Malware forensics : investigating and analyzing malicious code* (Syngress Pub.: Burlington, MA).

Arshad, H., A.B. Jantan, and O.I. Abiodun. 2018. 'Digital forensics: Review of issues in scientific validation of digital evidence', *Journal of Information Processing Systems*, 14: 346-76.

Bar, Moshe. 2000. *Linux internals* (McGraw-Hill: New York ; London).

Beckett, Jason, and Jill Slay. 2007. "Digital forensics: Validation and verification in a dynamic work environment." In *40th Annual Hawaii International Conference on System Sciences (HICSS'07)*, 266-76. Waikoloa, HI.

Bejtlich, Richard. 2006. *Extrusion detection : security monitoring for internal intrusions* (Addison-Wesley: Upper Saddle River, NJ).

Bjelland, P. C., K. Franke, and A. Arnes. 2014. 'Practical use of Approximate Hash Based Matching in digital investigations', *Digital Investigation*, 11: S18-S26.

Britz, Marjie. 2009. *Computer forensics and cyber crime : an introduction* (Pearson Prentice Hall: Upper Saddle River, N.J.).

Brown, Christopher L. T. 2006. *Computer evidence : collection & preservation* (Charles River Media: Hingham, Mass.).

Brunty, Josh. 2011. 'Validation of Forensic Tools and Software: A Quick Guide for the Digital Forensic Examiner', *Forensic Magazine*.

Bunting, Steve. 2012. *EnCase computer forensics : the official EnCE : EnCase certified examiner study guide* (Wiley: Indianapolis, Ind.).

Burch, AM , MR Durose, and KA Walsh. 2016. "Publicly funded forensic crime laboratories: Resources and services, 2014." In.

Butler, John, Hari Iyer, Rich Press, Melissa K. Taylor, Peter M. Vallone, and Sheila Willis. 2020. "NISTIR 8225: NIST Scientific Foundation Reviews." In.: NIST.

Caloyannides, Michael A. 2001. *Computer forensics and privacy* (Artech House: Boston, MA).

———. 2004. *Privacy protection and computer forensics* (Artech House: Boston).

Carrier, Brian. 2005. *File system forensic analysis* (Addison-Wesley: Boston, Mass. ; London).

Carvey, Harlan A. 2005. *Windows forensics and incident recovery* (Addison-Wesley: Boston).

———. 2014. *Windows forensic analysis toolkit : advanced analysis techniques for Windows 8* (Syngress: Amsterdam ; Boston).

———. 2016. *Windows registry forensics : advanced digital forensic analysis of the Windows registry* (Elsevier: Amsterdam).

Carvey, Harlan A., and Eoghan Casey. 2009. *Windows forensic analysis : DVD toolkit* (Syngress Pub.: Burlington, MA).

Casey, E. 2011a. 'The increasing need for automation and validation in digital forensics', *Digital Investigation*, 7: 103-04.

Casey, Eoghan. 2001. *Handbook of computer crime investigation : forensic tools and technology* (Academic Press: San Diego, Calif.).

———. 2010. *Handbook of digital forensics and investigation* (Academic: Amsterdam ; Boston).

———. 2011b. *Digital evidence and computer crime : forensic science, computers and the Internet* (Academic Press: Waltham, MA).

Chao, A. 1987. 'Estimating the population size for capture-recapture data with unequal catchability.', *Biometrics*, 43: 783-91.

Chao, A., P.K. Tsay, S-H. Lin, W-Y. Shau, and D-Y Chao. 2001. 'The applications of capture-recapture models to epidemiological data.', *Statistics in Medicine*, 20: 3123-57.

Cifuentes, J., A. L. S. Orozco, and L. J. G. Villalba. 2021. 'A survey of artificial intelligence strategies for automatic detection of sexually explicit videos', *Multimedia Tools and Applications*.

Cohen, M., S. Garfinkel, and B. Schatz. 2009. 'Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow', *Digital Investigation*, 6: S57-S68.

Cohen, M., and B. Schatz. 2010. 'Hash based disk imaging using AFF4', *Digital Investigation*, 7: S121-S28.

Cook, R., I. W. Evett, G. Jackson, P. J. Jones, and J. A. Lambert. 1998a. 'A hierarchy of propositions: deciding which level to address in casework', *Science & Justice*, 38: 231-39.

———. 1998b. 'A model for case assessment and interpretation', *Science & Justice*, 38: 151-56.

Cowen, David. 2013. *Computer forensics : infoSec Pro guide* (McGraw-Hill: New York).

Craiger, Philip, Jeff Swauger, Chris Marberry, and Connie Hendricks. 2006. 'Validation of Digital Forensic Tools.' in, *Digital Crime and Forensic Science in Cyberspace* (IGI Global: Hershey, PA, USA).

Cusack, Brian, and Alain Homewood. 2013. 'Identifying bugs in digital forensic tools', *Australian Digital Forensics Conference*.

Cusack, Brian, and James Liang. 2011. 'Comparing the performance of three digital forensic tools', *Journal of Applied Computing and Information Technology*, 15.

Davis, Chris, David Cowen, and Aaron Philipp. 2005. *Hacking exposed computer forensics : secrets & solutions* (McGraw-Hill/Osborne: New York).

Department of Homeland Security, and National Institute of Standards and Technology. 2009. "Test results for hardware write block tool - T4 Forensics SCSI Bridge (USB Interface)." In.

———. 2013a. "Test results for digital daa acquisition tool - Paladin v2.06." In.

———. 2013b. "Test results for digital data acquisition tool - DCFLDD v1.3.4-1." In.

———. 2013c. "Test results for digital data acquisition tool - Image MASSter Solo-4 Forensic." In.

———. 2013d. "Test results for digital data acquisition tool - IXImager v3.0.nov.12.12." In.

———. 2013e. "Test results for digital data acquisition tool - X-Ways Forensics 16.2 SR-5." In.

———. 2013f. "Test results for mobile device acquisition tools - EnCase Smartphone Examiner v7.0.3." In.

———. 2014a. "Test results for deleted file recovery and active file listing tools - FTK v3.3.0.33124." In.
———. 2014b. "Test results for deleted file recovery and active file listing tools - ILooKIX v2.2.3.151." In.
———. 2014c. "Test results for deleted file recovery and active file listing tools - SMART for Linux v2011-02-02." In.
———. 2014d. "Test results for deleted file recovery and active file listing tools - the Sleuth Kit (TSK)/Autopsy v3.2.2/2.24." In.
———. 2014e. "Test results for deleted file recovery and active file listing tools - X-Ways Forensics v16.0 SR-4." In.
———. 2014f. "Test results for deleted file recovery and active file listing tools (revised) - EnCase Forensic v6.18.0.59." In.
———. 2014g. "Test results for digital data acquisition tool - MacQuisition v2013R2." In.
———. 2014h. "Test results for digital data acquisition tool - Paladin v4.0." In.
———. 2014i. "Test results for digital data acquisition tool - Tableau TD3 Forensic Imager v1.3.0." In.
———. 2014j. "Test results for graphic file carving tool - Androit Photo Forensics 2013 v3.1d." In.
———. 2014k. "Test results for graphic file carving tool - EnCase Forensic v6.18.0.59." In.
———. 2014l. "Test results for graphic file carving tool - EnCase Forensic v7.09.05." In.
———. 2014m. "Test results for graphic file carving tool - FTK v4.1." In.
———. 2014n. "Test results for graphic file carving tool - iLook v2.2.7." In.
———. 2014o. "Test results for graphic file carving tool - PhotoRec v7.0-WIP." In.
———. 2014p. "Test results for graphic file carving tool - R-Studio v6.2." In.
———. 2014q. "Test results for graphic file carving tool - Recover My Files v5.2.1." In.
———. 2014r. "Test results for graphic file carving tool - Scalpel v2.0." In.
———. 2014s. "Test results for graphic file carving tool - X-Ways Forensics v17.6." In.
———. 2014t. "Test results for mobile device acquisition tools - iOS Crime Lab v1.0.1." In.
———. 2014u. "Test results for mobile device acquisition tools - Mobile Phone Examiner Plus v5.5.3.73." In.
———. 2014v. "Test results for mobile device acquisition tools - UFED Physical Analyzer v3.9.6.7." In.
———. 2014w. "Test results for mobile device acquisition tools - viaExtract v2.5." In.
———. 2014x. "Test results for video file carving tools - iLook v2.2.7." In.
———. 2014y. "Test results for video file carving tools - PhotoRec v7.0-WIP." In.
———. 2014z. "Test results for video file carving tools - R-Studio v6.2." In.
———. 2014aa. "Test results for video file carving tools - Recover my Files v5.2.1." In.
———. 2014ab. "Test results for video file carving tools - Scalpel v2.0." In.
———. 2014ac. "Test results for video file carving tools - X-Ways v17.6." In.
———. 2015a. "Test results for mobile device acquisition tool - MOBILedit Forensic v7.8.3.6085." In.
———. 2015b. "Test results for mobile device acquisition tool - Phone Forensics Express v2.1.2.2761." In.
———. 2015c. "Test results for mobile device acquisition tool - Secure View v3.16.4." In.
———. 2015d. "Test results for mobile device acquisition tools - Device Seizure v6.8." In.

———. 2015e. "Test results for mobile device acquisition tools - EnCase Smartphone Examiner v7.10.00.103." In.

———. 2015f. "Test results for mobile device acquisition tools - Lantern v4.5.6." In.

———. 2015g. "Test results for mobile device acquisition tools - Oxygen Forensic Suite 2015 - Analyst v7.0.0.408." In.

———. 2015h. "Test results for video file carving tool - EnCase v7.09.05." In.

———. 2016a. "Test results for digital acquisition tool: Dc3dd v7.2.61." In.

———. 2016b. "Test results for digital data acquisition tool: Logicube Forensic Falcon v3.OU1RC13." In.

———. 2016c. "Test results for digital data acquisition tool: Guymager v0.8.1." In.

———. 2016d. "Test results for digital data acquisition tool: Logicube Forensic Falcon v2.4u1." In.

———. 2016e. "Test results for digital data acquisition tool: Paladin v6.08." In.

———. 2016f. "Test results for digital data acquisition tool: Paladin v6.09." In.

———. 2016g. "Test results for digital data acquisition tool: Tableau TD2u Firmware v1.1.2.3948-4270f9c." In.

———. 2016h. "Test results for digital data acquisition tool: WiebeTech Ditto Forensic FieldStation v2016Mar01a." In.

———. 2016i. "Test results for digital data acquisition tool: X-Ways Forensics v18.8." In.

———. 2016j. "Test results for mobile device acquisition tool - BlackLight v2016.1." In.

———. 2016k. "Test results for mobile device acquisition tool - Device Seizure v7.4 build 5921.15166." In.

———. 2016l. "Test results for mobile device acquisition tool - MOBILedit Forensic v8.6.0.20354." In.

———. 2016m. "Test results for mobile device acquisition tool - MOBILedit Forensic v8.6.0.20354 November." In.

———. 2016n. "Test results for mobile device acquisition tool - Oxygen Forensics v8.3.1.105." In.

———. 2016o. "Test results for mobile device acquisition tool - Secure View v4.1.9." In.

———. 2016p. "Test results for mobile device acquisition tool - UFED 4PC v4.2.6.5 - Physical Analyzer v4.2.6.4." In.

———. 2016q. "Test results for mobile device acquisition tool - UFED Touch v4.4.0.1 Internal Build 4.2.8.36." In.

———. 2016r. "Test results for mobile device acquisition tool - XRY v7.0.1.37853." In.

———. 2017a. "Test results for mobile device acquisition tool - Blacklight v2016.3.1." In.

———. 2017b. "Test results for mobile device acquisition tool - Electronic Evidence Examiner Device Seizure v1.0.9466.18457." In.

———. 2017c. "Test results for mobile device acquisition tool - Mobile Phone Examiner Plus v5.6.0." In.

———. 2017d. "Test results for mobile device acquisition tool - MOBILedit Forensic Express v3.5.2.7047." In.

———. 2017e. "Test results for mobile device acquisition tool - MOBILedit Forensics Express v4.2.1.11207." In.

———. 2017f. "Test results for mobile device acquisition tool - Secure View v4.3.1." In.

———. 2017g. "Test results for mobile device acquisition tool - XRY Kiosk v7.0.0.36568." In.

———. 2017h. "Test results for mobile device acquisition tool - XRY v7.3.1." In.

———. 2017i. "Test resutls for mobile device acquisition tool - Lantern v4.6.8." In.

———. 2018a. "Test results for hardware write block tool - Forensic ComboDock FCDv5.5." In.

———. 2018b. "Test results for hardware write block tool - Forensic LabDock U5." In.

———. 2018c. "Test results for hardware write block tool - Forensic UltraDock FUDv5.5." In.

———. 2018d. "Test results for hardware write block tool - Media WriteBlocker." In.

———. 2018e. "Test results for hardware write block tool - Tableau eSATA Forensic Bridge T35es-R2." In.

———. 2018f. "Test results for hardware write block tool - Tableau Forensic FireWire Bridge T9." In.

———. 2018g. "Test results for hardware write block tool - Tableau Forensic PCle Bridge T7u." In.

———. 2018h. "Test results for hardware write block tool - Tableau Forensic SAS Bridge T6es-B." In.

———. 2018i. "Test results for hardware write block tool - Tableau Forensic SAS Bridge T6u." In.

———. 2018j. "Test results for hardware write block tool - Tableau Forensic SATA/IDE Bridge T35u." In.

———. 2018k. "Test results for hardware write block tool - Tableau Forensic Universal Bridge T356789IU." In.

———. 2018l. "Test results for hardware write block tool - Tableau Forensic USB 3.0 Bridge T8u." In.

———. 2018m. "Test results for hardware write block tool - Tableau Forensic USB Bridge T8-R2." In.

———. 2018n. "Test results for hardware write block tool - UltraBlock USB 3.0 Forensic Card Reader." In.

———. 2018o. "Test results for hardware write block tool - USB WriteBlocker." In.

———. 2018p. "Test results for mobile device acquisition tool - electronic evidence examiner - device seizure (E3:DS) v1.7." In.

———. 2018q. "Test results for mobile device acquisition tool - Final mobile forensics v2018.02.07." In.

———. 2018r. "Test results for mobile device acquisition tool - Katana forensics Triage v1.1802.220." In.

———. 2018s. "Test results for mobile device acquisition tool - Magnet AXIOM v1.2.1.6994." In.

———. 2018t. "Test results for mobile device acquisition tool - MD-NEXT v1.75.20171226.28830D4,MD-RED v2.3.20171226.28828D6." In.

———. 2018u. "Test results for mobile device acquisition tool - MOBILedit forensics v9.1.0.22420." In.

———. 2018v. "Test results for mobile device acquisition tool - Mobilyze v2018.1." In.

———. 2018w. "Test results for mobile device acquisition tool - Oxygen Forensics v10.0.0.81." In.

———. 2018x. "Test results for mobile device acquisition tool - XRY Kiosk v7.8.0." In.

———. 2018y. "Test results for mobile device acquisition tool - XRY v7.8.0." In.

———. 2018z. "Test results for mobile device acquisition tool -UFED Touch/ Physical Analyzer v6/2/1/17/ v6.3.0.284." In.

———. 2018aa. "Test results for mobile device acqusition tool - Blacklight 2018 Release 1.1." In.

———. 2018ab. "Test results for string search tool - Autopsy version 4.6.0." In.

———. 2019a. "Test results for binary image tool - final mobile forensics v2019.07.05." In.

———. 2019b. "Test results for mobile device acquisition tool - GrayKey OS Version 1.4.2 App Bundle 1.11.2.5." In.

———. 2019c. "Test results for mobile device acquisition tool - UFED 4PC v7.8.0.942/Physical Analyzer v7.9.0.223." In.

———. 2019d. "Test results for mobile device acquisition tool - UFED InField Kiosk v7.50.0875." In.

———. 2019e. "Test results for Windows registry forensic tool - EnCase forensic 8.07.00.93 (x64)." In.

———. 2019f. "Test results for Windows registry forensic tool - forensic Toolkit (FTK) 7.0.0.163 (x64)." In.

———. 2019g. "Test results mobile device acquisition tool - E3-DS v2.2.118.12.15844." In.

———. 2019h. "Test resutls for string search tool - X-Ways Forensics Version 19.6-SR-4 x64." In.

———. 2020a. "Test results (federated testing) for hardware write block device - CRU Forensic UltraDock FUDv5.5 Firmware Version f3.01.0011." In.

———. 2020b. 'Test results for string search tool - Access Data Forensic Toolkit (FTK) Version 7.0.0.163'.

———. 2020c. "Test results for string search tool - BlackLight Version 2018-R4." In.

———. 2020d. 'Test results for string search tool - EnCase Version 8.09.00.192'.

———. 2020e. 'Test results for string search tool - Magnet Axiom Version 4.1.1.20153'.

ENFSI. 2015. "Best Practice Manual for the Forensic Examination of Digital Technology." In.

Fellows, G. H. 2005. 'The joys of complexity and the deleted file', *Digital Investigation*, 2: 89-93.

Ferraro, Monique Mattei, Eoghan Casey, and Michael McGrath. 2005. *Investigating child exploitation and pornography : the Internet, the law and forensic science* (Elsevier/Academic Press: Amsterdam ; Boston, Mass.).

Flandrin, Flavien, William J. Buchanan, Richard Macfarlane, Bruce Ramsay, and Adrian Smales. 2014. "Evaluating Digital Forensic Tools (DFTs)." In *7th International Conference: Cybercrime Forensics Education and Training*.

Franqueira, V. N. L., J. Bryce, N. Al Mutawa, and A. Marrington. 2018. 'Investigation of Indecent Images of Children cases: Challenges and suggestions collected from the trenches', *Digital Investigation*, 24: 95-105.

Gardner, Ross M. 2012. *Practical crime scene processing and investigation* (CRC Press: Boca Raton, FL).

Garfinkel, S. 2012. 'Lessons learned writing digital forensics tools and managing a 30TB digital evidence corpus', *Digital Investigation*, 9: S80-S89.

Garfinkel, S., P. Farrell, V. Roussev, and G. Dinolt. 2009. 'Bringing science to digital forensics with standardized forensic corpora', *Digital Investigation*, 6: S2-S11.

Garfinkel, Simson, David J. Malan, Karl-Alexander Dubec, Christopher C. Stevens, and Cecile Pham. 2006. 'Advanced forensic format: An open, extensible format for disk imaging.' in Martin Olivier and Sujeet Shenoi (eds.), *Advances in Digital Forensics II: FIP International Conferences on Digital Forensics* (Springer: New York).

Gavrila, Serban, and Elizabeth Fong. 2004. "Forensic Software Testing Support Tools Test Summary Report." In.: NIST.

Gershteyn, P., M. Davis, G. Manes, and S. Shenoi. 2006. 'Extracting concealed data from BIOS chips', *Advances in Digital Forensics*, 194: 217-+.

Glisson, W. B., T. Storer, and J. Buchanan-Wollaston. 2013. 'An empirical comparison of data recovered from mobile forensic toolkits', *Digital Investigation*, 10: 44-55.

Grajeda, C., F. Breitinger, and I. Baggili. 2017. 'Availability of datasets for digital forensics - And what is missing', *Digital Investigation*, 22: S94-S105.

Grother, Patrick, Mei Ngan, and Kayee Hanaoka. 2019. "Face Recognition Vendor Test (FRVT) Part 3: Demographic Effects." In.

Guo, Y. H., J. Slay, and J. Beckett. 2009. 'Validation and verification of computer forensic software tools-Searching Function', *Digital Investigation*, 6: S12-S22.

Guttman, Barbara, Mary T. Laamanen, Craig Russell, Chris Atha, and James Darnell. 2022. "Results from a Black-Box Study for Digital Forensic Examiners." In. Gaithersburg, MD: National Institute of Standards and Technology.

Guttman, Barbara, James R. Lyle, and Richard Ayers. 2011. 'Ten Years of Computer Forensic Tool Testing', *Digital Evidence and Electronic Signature Law Review*: 139-47.

Hamming, R. W. 1950. 'Error Detecting and Error Correcting Codes', *Bell System Technical Journal*, 29: 147-60.

Harris, R. 2006. 'Arriving at an anti-forensics consensus: Examining how to define and control the anti-forensics problem', *Digital Investigation*, 3: S44-S49.

Hayes, Darren Richard. 2015. *A practical guide to computer forensics investigations* (Pearson: Indianapolis, Indiana).

Hibshi, H., T. Vidas, and L. Cranor. 2011. "Usability of Forensics Tools: A User Study." In *2011 Sixth International Conference on IT Security Incident Management and IT Forensics*, 81-91.

Honeycutt, Jerry. 1996. *Using the Windows 95 registry* (Que: Indianapolis, IN).

———. 1998a. *Using the Windows 98 registry* (Que: Indianapolis, Ind.).

———. 1998b. *Windows 98 registry handbook* (Que: Indianapolis, Ind.).

———. 2000. *Microsoft Windows 2000 registry handbook* (Que: Indianapolis, Ind.).

———. 2003. *Microsoft Windows XP registry guide* (Microsoft Press: Redmond, Wash.).

Horsman, G. 2019. 'Tool testing and reliability issues in the field of digital forensics', *Digital Investigation*, 28: 163-75.

Horsman, Graeme. 2018. '"I couldn't find it your honour, it mustn't be there!" – Tool errors, tool limitations and user error in digital forensics', *Science & Justice*, 58: 433-40.

Huebner, E., D. Bem, and C. K. Wee. 2006. 'Data hiding in the NTFS file system', *Digital Investigation*, 3: 211-26.

IDEMA. 2022. 'Advanced Format Definitions, Abbreviations, and Conventions', IDEMA, Accessed January 4, 2022. http://idema.org/?page_id=2153.

Inman, Keith, and Norah Rudin. 2000. *Principles and practice of criminalistics : the profession of forensic science* (CRC Press: Boca Raton, Fla.).

James, Joshua I., Alejandra Lopez-Fernandez, and Pavel Gladyhsev. 2014. 'Measuring accuracy of automated parsing and categorization tools and processes in digital investigations.' in Pavel Gladyhsev, A. Marrington and I. Baggili (eds.), *Digital Forensics and Cyber Crime. ICDF2C 2013. Lecture Notes of the Instute for Computer Sciences, Social Informatics, and Telecommunications Engineering.* (Springer, Cham).

Jones, Keith J., Richard Bejtlich, and Curtis W. Rose. 2005. *Real digital forensics : computer security and incident response* (Addison-Wesley: Upper Saddle River, NJ).

Kim, Y. ;Ross, S. 2012. 'Digital Forensics Formats: Seeking a Digital PreservationStorage Container Format for Web Archiving', *The International Journal of Digital Curation*, 7: 21-39.

Kipper, Gregory. 2004. *Investigator's guide to steganography* (Auerbach Publications: Boca Raton, FL).

Knuth, Donald Ervin. 1968. *The art of computer programming* (Addison-Wesley Pub. Co.: Reading, Mass.,).

Lyle, J., S. Mead, and K. Rider. 2007. 'Disk drive I/O commands and write blocking', *Advances in Digital Forensic Iii*, 242: 163-+.

Lyle, J. R. 2006a. 'A strategy for testing hardware write block devices', *Digital Investigation*, 3: S3-S9.

———. 2006b. 'A strategy for testing hardware write block devices', *Digital Investigation*: S3-S9.

———. 2010. 'If error rate is such a simple concept, why don't I have one for my forensic tool yet?', *Digital Investigation*, 7: S135-S39.

Lyle, J. R., and M. Wozar. 2007. 'Issues with imaging drives containing faulty sectors', *Digital Investigation*, 4: S13-S15.

Lyle, James. 2002. "Testing Disk Imaging Tools." In *DFRWS*. Syracuse, NY.

Magnet Forensics. 2021a. "AXIOM Artifact Reference 50.0." In.

———. 2021b. "IEF Artifact Reference 6.48.0." In.

Malin, Cameron H., Eoghan Casey, and James M. Aquilina. 2012. *Malware forensics field guide for Windows systems : digital forensics field guides* (Syngress: Waltham, MA).

Malin, Cameron H., Eoghan Casey, James M. Aquilina, and Curtis W. Rose. 2014. *Malware forensics field guide for Linux systems* (Elsevier: Amsterdam ;).

Marcella, Albert J., and Doug Menendez. 2008. *Cyber forensics : a field manual for collecting, examining, and preserving evidence of computer crimes* (Auerbach Publications: New York).

Marshall, A. M., and R. Paige. 2018. 'Requirements in digital forensics method definition: Observations from a UK study', *Digital Investigation*, 27: 23-29.

McKemmish, Rodney. 2008. 'When is Digital Evidence Forensically Sound?' in Indrajit Ray and Sujeet Shenoi (eds.), *Advances in Digital Forensics IV. Digital Forensics 2008. IFIP - The International Federation for Information Processing* (Springer: Boston, MA).

Mohay, George M. 2003. *Computer and intrusion forensics* (Artech House: Boston).

National Institute of Justice, and National Institute of Standards and Technology. 2002a. "Test results for disk imaging tools: dd GNU fileutils 4.0.36, Provided with Red Hat Linux 7.1." In.

———. 2002b. "Test results for disk imaging tools: SafeBack 2.18." In.

———. 2003. "Partial results from prototype testing effors for disk imaging tools: SafeBack 2.0." In.
———. 2004a. "Test results for software write block tools - RCMP HDL VO.4." In.
———. 2004b. "Test results for software write block tools - RCMP HDL VO.5." In.
———. 2004c. "Test results for software write block tools - RCMP HDL VO.7." In.
———. 2004d. "Test results for software write block tools - RCMP HDL VO.8." In.
———. 2005a. "Test results for software write block tools - PDBLOCK v1.02 (PDB LITE)." In.
———. 2005b. "Test results for software write block tools - PDBLOCK Version 2.00." In.
———. 2005c. "Test results for software write block tools - PDBLOCK Version 2.10." In.
———. 2006a. "Test results for hardware write block tool - Digital intelligence Firefly 800 IDE (FireWire Interface)." In.
———. 2006b. "Test results for hardware write block tool - Digital Intelligence UltraBlock SATA (FireWire interface)." In.
———. 2006c. "Test results for hardware write block tool - Digital Intelligence UltraBlock SATA (USB Interface)." In.
———. 2006d. "Test results for hardware write block tool - FastBloc IDE (Firmware Version 16)." In.
———. 2006e. "Test results for hardware write block tool - ICS ImageMasster DriveLock IDE (Firmware Version 17)." In.
———. 2006f. "Test results for hardware write block tool - MyKey NoWrite (Firmware Version 1.05)." In.
———. 2006g. "Test results for hardware write block tool - WiebeTech FireWire DriveDock Combo (FireWire Interface)." In.
———. 2006h. "Test results for hardware write block tool - WiebeTech Forensic ComboDock (USB Interface)." In.
———. 2006i. "Test results for hardware write block tool - WiebeTech Forensic SATADock (FireWire Interface)." In.
———. 2006j. "Test results for hardware write block tool - WiebeTech Forensic SATADock (USB Interface)." In.
———. 2007a. "Test results for hardware write block tool - FastBloc FE (FireWire Interface)." In.
———. 2007b. "Test results for hardware write block tool - FastBloc FE (USB Interface)." In.
———. 2007c. "Test results for hardware write block tool - Tableau Forensic IDE Pocket Bridge T14 (FireWire Interface)." In.
———. 2007d. "Test results for hardware write block tool - Tableau Forensic SATA Bridge T3u (Firewire Interface)." In.
———. 2007e. "Test results for hardware write block tool - Tableau Forensic SATA Bridge T3u (USB Interface)." In.
———. 2007f. "Test results for hardware write block tool - Tableau T5 Forensic IDE Bridge (FireWire Interface)." In.
———. 2007g. "Test results for hardware write block tool - Tableau T5 Forensic IDE Bridge (USB Interface)." In.
———. 2008a. "Test results for digital data acquisition tool - DCCIdd (Version 2.0)." In.
———. 2008b. "Test results for digital data acquisition tool - EnCase Linen v5.05f." In.

———. 2008c. 'Test results for digital data acquisition tool - EnCase LinEn v6.01'.
———. 2008d. "Test results for digital data acquisition tool - FTK Imager v2.5.3.14." In.
———. 2008e. "Test results for digital data acquisition tool: EnCase 4.22a." In.
———. 2008f. "Test results for hardware write block tool - Tableau T8 Forensic USB Bridge (FireWire Interface)." In.
———. 2008g. "Test results for hardware write block tool - Tableau T8 Forensic USB Bridge (USB Interface)." In.
———. 2008h. "Test results for mobile device acquisition tool - guidance software Neutrino 1.4.14." In.
———. 2008i. "Test results for mobile device acquisition tool - Micro Systemation .XRY 3.6." In.
———. 2008j. "Test results for mobile device acquistion tool - Paraben Device Seizure 2.1." In.
———. 2008k. "Test results for software write block tools - Writeblocker Windows 2000 V5.02.00." In.
———. 2008l. "Test results for software write block tools - Writeblocker Windows XP V6.10.0." In.
———. 2009a. "Test results for digital data acquisition tool - BlackBag MacQuisition v2.2." In.
———. 2009b. 'Test results for digital data acquisition tool - EnCase v6.5'.
———. 2009c. "Test results for hardware write block tool - T4 Forensic SCSI Bridge (FireWire Interface)." In.
———. 2010a. "Test results for mobile device acquisition tool - BitPim - 1.0.6." In.
———. 2010b. "Test results for mobile device acquisition tool - iXAM Version 1.5.6." In.
———. 2010c. "Test results for mobile device acquisition tool - XRY 5.0.2." In.
———. 2010d. "Test results for mobile device acquisition tool - Zdziarski's Method." In.
———. 2010e. "Test results for mobile device acquition tool - Secure View 2.1.0." In.
———. 2011a. "Test results for digital data acquisition tool - Image MASSter Solo-3 Forensics; Software Version 2.0.10.23f." In.
———. 2011b. "Test results for digital data acquisition tool - Tableau TD1 Forensic Duplicator; Firmware v2.34 2/17/2011." In.
———. 2011c. "Test results for mobile device acquisition tool - Mobilyze v1.1." In.
———. 2012a. "Test results for mobile device acquisition tool - CellBrite UFED 1.1.8.6-Report Mgr 1.8.3/UFED Physical Analyzer 2.3.0." In.
———. 2012b. 'Test results for mobile device acquisition tool - Mobile Phone Examiner Plus (MPE+) 4.6.0.2'.
———. 2013a. "Test results for digital data acquisition tool - FTK Imager CLI 2.9.0 Debian." In.
———. 2013b. "Test results for mobile device acquisition tool - Device Seizure v5.0 build 4582.15907." In.
———. 2013c. "Test results for mobile device acquisition tool - Lantern v2.3." In.
———. 2013d. "Test results for mobile device acquisition tool - Micro Systemation XRY v6.3.1." In.
———. 2013e. "Test results for mobile device acquisition tool - Secure View 3v3.3.8.0." In.
———. 2014. "Forensic file carving tool test assertions and test plan v 1.0." In.
———. 2018. "Test results for hardware write block tool - Forensic ComboDock v5." In.

National Institute of Standards and Technology. 2003. "Software write block tool specification & test plan." In.

———. 2004a. "Digital data aquisition tool specification." In.

———. 2004b. "Hardware write blocker device (HWB) specification Version 2.0." In.

———. 2005a. "Digital data acquisition tool test assertions and test plan." In.

———. 2005b. "Hardware write blocker (HWB) assertions and test plan." In.

———. 2008. "Forensic string searching tool requirements specification." In.

———. 2009a. "Active file identification & deleted file recovery tool specification." In.

———. 2009b. "Forensic media preparation tool test assertions and test plan." In.

———. 2009c. "Forensic Storage Media Preparation Tool Specification." In.

———. 2014. "Forensic file carving tool specification Version 1.0." In.

———. 2018a. "Forensic string searching tool test assertions and test plan." In.

———. 2018b. "Windows registry forensic tool specification." In.

———. 2018c. "Windows registry forensic tool test assertions and test plan." In.

———. 2019a. 'Computer Forensics Tool Testing Program (CFTT)', Accessed 9 March 2020. https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt.

———. 2019b. "Mobile Device Forensic Tool Test Specification, Test Assertions and Test Cases V3.0." In.

———. 2020. 'CFTT Federatied Testing Project'. https://www.nist.gov/itl/ssd/software-quality-group/computer-forensics-tool-testing-program-cftt/federated-testing.

———. 2021a. 'OSAC Registry'.

———. 2021b. "SQLite data recovery specification, test assertions, and test cases." In.

Nelson, Bill. 2015. *Guide to computer forensics and investigations : processing digital evidence* (Cengage Learning: Boston, MA).

Nikkel, B. J. 2009. 'Forensic analysis of GPT disks and GUID partition tables', *Digital Investigation*, 6: 39-47.

NIST. 2015a. "FIPS 180-4 Secure Hash Standard." In.

———. 2015b. "FIPS 202 SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions." In.

———. 2015c. "Secure Hash Standard." In, edited by U.S. Department of Commerce. Washington, D.C.

Nordvik, R., H. Georges, F. Toolan, and S. Axelsson. 2019. 'Reverse engineering of ReFS', *Digital Investigation*, 30: 127-47.

Nordvik, R., R. Stoykova, K. Franke, S. Axelsson, and F. Toolan. 2021. 'Reliability validation for file system interpretation', *Forensic Science International-Digital Investigation*, 37.

Novak, Martin, Jonathan Grier, and Daniel Gonzalez. 2019. 'New approaches to digital evidence aquisition and analysis', *NIJ Journal*, 280: 1-8.

OSAC. 2018. "A Framework for Harmonizing Forensic Science Practices and Digital & Multimedia Evidence." In *OSAC Technical Series*.

Peterson, W. W., and D. T. Brown. 1961. 'Cyclic Codes for Error Detection', *Proceedings of the Institute of Radio Engineers*, 49: 228-&.

Philipp, Aaron, David Cowen, and Chris Davis. 2010. *Hacking exposed computer forensics* (McGraw-Hill/Osborne: New York).

Piper, S., M. Davis, G. Manes, and S. Shenoi. 2006. 'Detecting hidden data in Ext2/Ext3 file systems', *Advances in Digital Forensics*, 194: 245-+.

Pollitt, Mark. 2010. 'A History of Digital Forensics', *IFIP Advances in Information and Communications Technology*, 337: 3-15.

Popper, Karl R. 1959. *The logic of scientific discovery* (Basic Books: New York,).

Ramsland, Katherine M. 2016. *Confession of a serial killer : the untold story of Dennis Rader, the BTK killer* (ForeEdge: Hanover).

Regulator, Forensic Science. 2020. "Guidance: Method validation in digital forensics." In.

Reiber, Lee. 2019. *Mobile forensic investigations : a guide to evidence collection, analysis, and presentation* (McGraw-Hill Education: New York).

Richard, G., V. Roussev, and L. Marziale. 2007. 'In-place file carving', *Advances in Digital Forensic Iii*, 242: 217-+.

Risinger, D. 2018. 'The five functions of forensic science and the validation issues they raise: A piece to incite discussion on validation.', *Seton Hall Law Review*, 48: 719-32.

Rodriguez, B., and G. Peterson. 2007. 'Detecting steganography using multi-class classification', *Advances in Digital Forensic Iii*, 242: 193-+.

Rosenblatt, Kenneth S. 1995. *High-technology crime : investigating cases involving computers* (KSK Publications: San Jose, Calif.).

Sammes, A. J., and Brian Jenkinson. 2000. *Forensic computing : a practitioner's guide* (Springer: London ; New York).

Sammons, John. 2014. *The basics of digital forensics : the primer for getting started in digital forensics* (Elsevier: Waltham, MA).

Sanderson, P. 2018. *SQLite Forensics* (Independently Published).

Schatz, B. L. 2015. 'Wirespeed: Extending the AFF4 forensic container format for scalable acquisition and live analysis', *Digital Investigation*, 14: S45-S54.

Silberschatz, Abraham, Peter B. Galvin, and Greg Gagne. 2018. *Operating system concepts* (Wiley: Hoboken, NJ).

Slade, Robert. 2004. *Software forensics : collecting evidence from the scene of a digital crime* (McGraw-Hill: New York).

Solomon, Michael, Diane Barrett, and Neil Broom. 2005. *Computer forensics jumpstart* (Sybex: San Francisco).

Solomon, Michael, K. Rudolph, Ed Tittel, Neil Broom, and Diane Barrett. 2011. *Computer forensics jumpstart* (Wiley Publishing: Indianapolis, Indiana).

Steel, Chad. 2006. *Windows forensics : the field guide for conducting corporate computer investigations* (Wiley Pub.: Indianapolis, IN).

Stephenson, Peter. 2000. *Investigating computer-related crime* (CRC Press: Boca Raton, Fla).

Stephenson, Peter, and Keith Gilbert. 2013. *Investigating computer-related crime* (Taylor & Francis: Boca Raton).

SWGDE. 2014. "Recommended Guidelines for Validation Testing." In *Scientific Working Group on Digital Evidence, version 2.0.*

———. 2016a. "Best practices for collection of damaged mobile devices." In *Scientific Working Group on Digial Evidence, version 1.1.*

———. 2016b. "Best practices for mobile phone forensics." In *Scientific Working Group on Digital Evidence, version 2.0.*

———. 2016c. 'SWGDE Digital & Multimedia Evidence Glossary Version 3.0'.

———. 2017a. "Best practices for maintaining the integrity of imagery." In *Scientific Working Group on Digital Evidence v1.0*.

———. 2017b. "Best practices for the acquistion of data from novel digital devices." In *Scientific Working Group on Digital Evidence, version 1.0*.

———. 2018a. "Best practices for digital evidence collection." In *Scientific Working Group on Digital Evidence, version 1.0*.

———. 2018b. "Establishing confidence in digital forensic results by error mitigation analysis." In *Scientific Working Group on Digital Evidence, version 1.7*.

———. 2018c. "Minimum requirements for testing tools used in digital and multimedia forensics." In *Scientific Working Group on Digital Evidence, version 1.0*.

———. 2019a. "Best practices for digital evidence acquisition from cloud service providers." In *Scientific Working Group on Digital Evidence, version 1.0*.

———. 2019b. "Best practices for mobile device evidence collection & preservation, handling, and acquisition." In *Scientific Working Group on Digital Evidence, version 1.1*.

Thompson, E. 2005. 'MD5 collisions and the impact on computer forensics', *Digital Investigation*, 2: 36-40.

Tilling, K. 2001. 'Capture-recapture methods - useful or misleading?', *International Journal of Epidemiology*, 30: 12-14.

Tobin, P., N. A. Le-Khac, and M. T. Kechadi. 2016. 'A Lightweight Software Write-blocker for Virtual Machine Forensics', *2016 Sixth International Conference on Innovative Computing Technology (Intech)*: 730-35.

Turner, P. 2006. 'Selective and intelligent imaging using digital evidence bags', *Digital Investigation*, 3: S59-S64.

Vandeven, Sally. 2014. "Forensic images: For your viewing pleasure [White Paper]." In *Information Security Reading Room*. SANS Institute.

Wang, Xiaoyun, Yiqun Lisa Yin, and Hongbo Yu. 2005. "Finding Collisions in the Full SHA-1." In *CRYPTO 2005*, edited by V. Shoup, 17-36. Berlin, Heidelberg: Springer, Cham.

Wang, Xiaoyun, and Hongbo Yu. 2005. "How to break MD5 and other hash functions." In, 19-35. Berlin, Heidelberg: Springer Berlin Heidelberg.

Williams, Howard E. 2006. *Investigating white-collar crime : embezzlement and financial fraud* (Charles C. Thomas: Springfield, Ill.).

Wilsdon, T., and J. Slay. 2006. "Validation of forensic computing software utilizing black box testing techniques." In *4th Australian Digital Forensics Conference*. Edith Cowan University: Security Research Institute (SRI), Edith Cowan University.

Yates, Maynard, and Hongmei Chi. 2011. "A framework for designing benchmarks of investigating digital forensics tools for mobile devices." In *Proceedings of the 49th Annual Southeast Regional Conference*, 179–84. Kennesaw, Georgia: Association for Computing Machinery.