Designation: E3016 – 15$^{\varepsilon 1}$

Standard Guide for Establishing Confidence in Digital Forensic Results by Error Mitigation Analysis[1]

This standard is issued under the fixed designation E3016; the number immediately following the designation indicates the year of original adoption or, in the case of revision, the year of last revision. A number in parentheses indicates the year of last reapproval. A superscript epsilon (ε) indicates an editorial change since the last revision or reapproval.

ε[1] NOTE—Editorial changes were made throughout in September 2016.

1. Scope

1.1 This guide provides a process for recognizing and describing both errors and limitations associated with tools used to support digital forensics. This is accomplished by explaining how the concepts of errors and error rates should be addressed in digital forensics. It is important for practitioners and stakeholders to understand that digital forensic techniques and tools have known limitations, but those limitations have differences from errors and error rates in other forensic disciplines. This guide proposes that confidence in digital forensic results is best achieved by using an error mitigation analysis approach that focuses on recognizing potential sources of error and then applying techniques used to mitigating them, including trained and competent personnel using tested and validated methods and practices.

2. Referenced Documents

2.1 *ISO Standard:*[2]

ISO/IEC 17025 General Requirements for the Competence of Testing and Measurement Laboratories

2.2 *SWGDE Standards:*[3]

SWGDE Model Quality Assurance Manual for Digital Evidence

SWGDE Standards and Controls Position Paper

SWGDE/SWGIT Proficiency Test Program Guidelines

SWGDE/SWGIT Guidelines & Recommendations for Training in Digital & Multimedia Evidence

3. Significance and Use

3.1 Digital forensics is a complex field that is heavily reliant on algorithms that are embedded in automated tools and used to process evidence. Weaknesses or errors in these algorithms, tools, and processes can potentially lead to incorrect findings. Indeed, errors have occurred in a variety of contexts, demonstrating the need for more scientific rigor in digital forensics. This guide proposes a disciplined approach to mitigating potential errors in evidence processing to reduce the risk of inaccuracies, oversights, or misinterpretations in digital forensics. This approach provides a scientific basis for confidence in digital forensic results.

3.2 Error rates are used across the sciences to explain the amount of uncertainty or the limitation of a given result. The goal is to explain to the reader (or receiver of the result) the confidence the provider of the result has that it is correct. Many forensic disciplines use error rates as a part of how they communicate their results. Similarly, digital forensics needs to communicate how and why there is confidence in the results. Because of intrinsic difference between the biological and chemical sciences and computer science, it is necessary to go beyond error rates. One difference between chemistry and computer science is that digital technology is constantly changing and individuals put their computers to unique uses, making it infeasible to develop a representative sample to use for error rate calculations. Furthermore, a digital forensic method may work well in one environment but fail completely in a different environment.

3.3 This document provides a disciplined and structured approach for addressing and explaining potential errors and error rates associated with the use of digital forensic tools/processes in any given environment. This approach to establishing confidence in digital forensic results addresses *Daubert* considerations.

4. Background

4.1 Digital forensic practitioners are confident in the ability of their methods and tools to produce reliable conclusions; however, they often struggle to establish their confidence on a scientific basis. 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 actually come from the same source. But in digital forensics, there are fundamental differences in the nature of

[1] This guide is under the jurisdiction of ASTM Committee E30 on Forensic Sciences and is the direct responsibility of Subcommittee E30.12 on Digital and Multimedia Evidence.

Current edition approved May 1, 2015. Published June 2015. DOI: 10.1520/E3016-15E01.

[2] Available from American National Standards Institute (ANSI), 25 W. 43rd St., 4th Floor, New York, NY 10036, http://www.ansi.org.

[3] Available from the Scientific Working Group on Digital Evidence (SWDGE), https://www.swgde.org.

many processes that can make trying to use statistical error rates inappropriate or misleading.

4.2 The key point to keep in mind is the difference between random errors and systematic errors. Random errors are based in natural processes and the inability to perfectly measure them. Systematic errors, in contrast, are caused by imperfect implementations. Digital forensics – being based on computer science – is far more prone to systematic than random errors. Additionally, the rapid change in technology including the innumerable permutations of hardware, software and firmware makes it close to impossible to address all situations.

4.3 One fundamental difference between digital forensics and other forensic disciplines is that many forensic disciplines try to determine whether or not two artifacts are a match (for example, from the same source), whereas digital forensics predominantly endeavors to find multiple artifacts that may show or imply actions by an individual. An error rate for a matching task focuses on establishing how often a false positive or a false negative occurs. Error rates for matching tasks are often statistical in nature and may derive from taking a measurement or sample from a population. Conversely, in digital forensics, there is often a series of tasks, any one of which could introduce error of a systematic rather than statistical nature. Even though there are errors, the errors in digital forensic tasks/processes are not always characterized in a useful or meaningful way by an error rate.

4.4 For each digital forensic task, there is an underlying algorithm (how the task should be done) 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. Characterizing hashing algorithms with an error rate is appropriate because the algorithms assume a file selected at random for the population of all possible files.

4.5 Once an algorithm is implemented in software, in addition to the inherent error rate of the algorithm, the implementation may 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 software containing the hash algorithm may be badly written and may produce the same hash every time an input file starts with the symbol "$".

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

4.6.1 *Incompleteness*—All the relevant information has not been acquired or found by the tool. For example, an acquisition might be incomplete or not all relevant artifacts identified from a search.

4.6.2 *Inaccuracy*—The tool does not report accurate information. Specifically, the tool should not report things that are not there, should not group together unrelated items, and should not alter data in a way that changes the meaning. Assessment of accuracy in digital forensic tool implementations can be categorized as follows:

4.6.2.1 *Existence*—Are all reported artifacts reported as present actually present? For example, a faulty tool might add data that was not present in the original.

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

4.6.2.3 *Association*—Do all items associated together actually belong together? A faulty tool might incorrectly associate information pertaining to one item with a different, unrelated item. For instance, a tool might parse a web browser history file and incorrectly 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.

4.6.2.4 *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.

4.6.3 *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 digital forensic tools present information.

4.7 The basic strategy to develop confidence in the digital forensic results is to mitigate errors, including known error rates, by applying tool testing and sound quality control measures as described in this document including:

4.7.1 *Tool Testing:*

4.7.1.1 Determine applicable scenarios that have been considered in tool testing.

4.7.1.2 Assess known tool anomalies and how they apply to the current case.

4.7.1.3 Find untested scenarios that introduce uncertainty in tool results.

4.7.2 *Sound Quality Control Procedures:*

4.7.2.1 Tool performance verification.

4.7.2.2 Personnel training, certification and regular proficiency testing.

4.7.2.3 Follow written procedures and document any necessary deviations/exceptions.

4.7.2.4 Laboratory accreditation.

4.7.2.5 Technical/peer review.

4.7.2.6 Technical and management oversight.

4.7.2.7 Use multiple tools and methods.

4.7.2.8 Maintain awareness of past and current problems.

4.7.2.9 Reasonableness and consistency of results for the case context.

4.8 A more formalized approach to handling potential sources of error in digital forensic processes is needed in order to address considerations such as those in *Daubert*.

4.9 The error mitigation analysis process involves recognizing sources of potential error, taking steps to mitigate any errors, and employing a quality assurance approach of continuous human oversight and improvement. Rather than focusing only on error rates, this more comprehensive approach takes into account all of the careful measures that can be taken to ensure that digital forensics processes produce reliable results. When error rates can be calculated, they can and should be included in the overall error mitigation analysis.

5. Procedures

5.1 Mitigating errors in a digital forensics process begins by answering the following questions:

5.1.1 Are the techniques (for example, hashing algorithms or string searching) used to process the evidence valid science?

5.1.2 Are the implementations of the techniques (for example, software or hardware tools) correct and appropriate for the environment where they are used?

5.1.3 Are the results of the tools interpreted correctly?

5.2 Considering each of these questions is critical to understanding errors in digital forensics. The next three sections explain the types of error associated with each question. In the first section, *Techniques* (5.3), the basic concept of error rates is addressed along with a discussion of how error rates depend on a stable population. The second section, *Implementation of Techniques in Tools* (5.4), addresses systematic errors and how tool testing is used to find these errors. The third section, *Tool Usage and Interpreting Results* (5.5), summarizes how practitioners use the results of digital forensic tools. This overall approach to handling errors in digital forensics helps address *Daubert* considerations.

5.3 *Techniques*—In computer science, the techniques that are the basis for digital processing includes copying bits and the use of algorithms to search and manipulate data (for example, recover files). These techniques can sometimes be characterized with an error rate.

5.3.1 *Error Rates*—An error rate has an explicit purpose – to show how strong the technique is and what its limitations are. There are many factors that can influence an error rate including uncertainties associated with physical measurements, algorithm weaknesses, statistical probabilities, and human error.

NOTE 1—*Systematic and Random Errors:* Error rates for many procedures can be treated statistically, however not all types of experimental uncertainty can be assessed by statistical analysis based on repeated measurements. For this reason, uncertainties are classified into two groups: the random uncertainties, which can be treated statistically, and the systematic uncertainties, which cannot.[4] The uncertainty of the results from software tools used in digital forensics is similar to the problems of measurement in that there may be both a random component (often from the underlying algorithm) and a systematic component (usually coming from the implementation).

5.3.1.1 Error rates are one of the factors described in *Daubert* to ascertain the quality of the science in expert testimony.[5] The underlying computer techniques are comparable to the type of science that is described in *Daubert*. Are the underlying techniques sound science or junk science? Are they used appropriately? In computer science, the types of techniques used are different from DNA analysis or trace chemical analysis. In those sciences, the technique or method is often used to establish an association between samples. These techniques require a measurement of the properties of the samples. Both the measurements of the samples and the associations have random errors and are well described by error rates.

5.3.1.2 Differences between digital and other forensic disciplines change how digital forensics uses error rates. There are error rates associated with some digital forensic techniques. For example, there are false positive rates for cryptographic hashing; however, the rate is so small as to be essentially zero. Similarly, many algorithms such as copying bits also have an error rate that is essentially zero. See Appendix X1, X1.2 and X1.3, for a discussion of error rates associated with hashing and copying.

5.3.2 *Error Rates and Populations*—There are other major differences between digital forensics and natural sciences-based forensic disciplines. In biology and chemistry-based disciplines, the natural components of a sample remain fairly static (for example, blood, hair, cocaine). Basic biology and chemistry do not change (although new drugs are developed and new means of processing are created). In contrast, information technology changes constantly. New types of drives (for example, solid-state drives) and applications (for example, Facebook) may radically differ from previous ones. There are a virtually unlimited number of combinations of hardware, firmware, and software.

5.3.2.1 The rapid and significant changes in information technology lead to another significant difference. Error rates, as with other areas of statistics, require a "population." One of the key features of a statistical population is that it is stable, that is, the composition remains constant. This allows predictions to be made. Since IT changes quickly and unpredictably, it is often infeasible to statistically describe a population in a usable way because, while the description may reflect an average over the entire population, it may not be useful for individual situations. See Note 2 for an example of this.

NOTE 2—*Deleted File Recovery Example:* File fragmentation is significant to the performance of the deleted file recovery algorithm. If some file systems have low fragmentation, many deleted files will be recoverable. However, if there is a large amount of fragmentation, the recovered files will tend to be mixtures of multiples files and therefore harder to recover. So the error rate will be low for the algorithm applied to a drive with low fragmentation and high for a drive with high fragmentation. If one tries to look at a large number of drives to derive a single error rate, it would not be applicable for a particular drive because each drive is very likely to be different from the average. (The average will not address drives with either high or low fragmentation.) Furthermore, the error rate would not apply to solid-state drives or other file systems.

5.3.2.2 In examining these two differences – *(1)* the virtually infinite number of combinations, and *(2)* the rapid pace of change – it can be seen that error rates for digital forensics are

[4] Taylor, John R., *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*, University Science Books, Sausalito, CA, 1997, p. 93.

[5] Daubert v. Merrell Dow Pharmaceuticals (92-102), 509 U.S. 579, 1993.

different from other forensic disciplines. It is apparent that the error rate for many techniques being close to zero would imply that the topic of errors is of no concern to the digital forensics profession. This is clearly not the case. Similarly, it is not useful to say that potential sources of error cannot be addressed because of the lack of a meaningful population.

5.3.2.3 In order to understand error meaningfully, it is necessary to look at digital forensic tools. The tools implement a variety of computer science techniques and are "where the rubber hits the road" in digital forensics. Errors in tools and their use can have a much more significant negative impact on a digital forensic process. The next section discusses these types of errors.

5.4 *Implementation of Techniques in Tools*—The kinds of errors that occur in tools are systematic errors, not the random errors generally associated with measurements. See Note 1 for an explanation of random and systematic errors. Digital forensic tools (for example, software, hardware, and firmware) are implementations of techniques. Tools are known to contain bugs of varying impact. Bugs are triggered by specific conditions and result in an incorrect output. For example, a tool may have a bug that causes it to underreport the size of a hard drive leading to a partial acquisition.

5.4.1 Because software bugs are logic flaws, the tool will produce the same result if given the same inputs. (In some rare cases, it may be that not all inputs are known or reproducible, in which case the program output can vary from run to run.) The output is not random, even though it is wrong. These are the systematic errors. The appendix has digital forensics-based examples showing the difference between the error rate of a technique and systematic errors of tool.

5.4.2 In order to address systematic errors in tools, one must draw on computer science and software engineering. Software engineering provides methods for testing software to ascertain if it does what it is supposed to do. *Software testing and validation is the primary method for mitigating the risk of errors in tools.* Software testing can never prove that a tool is always functioning correctly; however, good testing can lead to confidence that the tool is unlikely to fail within the situations for which it has been tested.

5.4.3 There is another situation – primarily within forensic imaging of hard drives – that may cause tools to give different, but acceptable, results when processing the same drive. While imaging a hard drive, tools may not be able to read bad sectors on a drive. Tools may skip varying amounts of readable sectors that surround the bad sector for performance reasons. The resulting forensic images of a given drive made by different tools can be different and will have different hash values. Neither the tools' differing strategies for imaging a hard drive with bad sectors, nor the resulting images that differ are errors. They are, instead, the result of basic limitations with reading failing hardware.

5.4.4 When searching for something, such as a keyword or type of file, it is possible that the tool will find things that are not relevant (false positive) or fail to find things that are (false negative). These are not errors in the colloquial sense of a mistake, but are a method to describe the limitations of the tool. Digital forensic tools are designed to report only information that actually exists on the original drive, and not to report anything that does not exist. One of the goals of tool testing is to verify that this holds true.

5.5 *Tool Usage and Interpreting Results*—Even when a technique is properly implemented in a tool, the tool can be used improperly, leading to errors. Furthermore, misinterpretation of what certain information means can result from a lack of understanding of the underlying data or from ambiguities in the way digital forensic tools present information.

5.5.1 Another significant consideration related to the interpretation of results is assessing the quality of data that was reconstructed from deleted material or recovered in an unusual manner. Such data may be incomplete, may mix data from multiple original sources, or have other problems. Technical/peer review and use of a second method are often needed to address the limitations of reconstruction and recovery.

5.5.2 The errors associated with the improper tool usage, misinterpretation of results, and human factors errors are beyond the scope of this document. They can best be addressed by sound management practices including training, proficiency testing, peer review, and best practices. Additional information is available in the SWGDE-SWGIT Guidelines and Recommendations for Training and the SWGDE Model Quality Assurance Manual for Digital Evidence Laboratories, Sections 5.2 and 5.9.

6. Error Mitigation Techniques

6.1 The field of digital forensics requires an approach to error analysis that goes beyond error rates, and addresses the broader scope of errors that are relevant to digital forensics. Digital forensics is best served by a framework that guides practitioners to state the sources of potential errors and how they were mitigated in a disciplined manner. This document presents an error mitigation analysis process that addresses each discrete digital forensic task/process in order to accomplish this. The analysis must be flexible enough to address the wide range of evidence types and sources. Mitigation techniques will not be able to address every potential situation and the resulting error mitigation analysis should clearly state this.

6.1.1 An error mitigation analysis must address the potential sources of error for each major process and document the mitigation strategies that were employed. A list of common mitigation strategies is described below. Three approaches for applying these as part of an Error Mitigation Analysis Report are included in Section 8. Many of these activities are discussed in ISO/IEC 17025, Requirements for the Competency of Test and Calibration Laboratories. Effective implementation of these activities will reduce the risk of errors.

6.2 *Tool Testing*—Tool Testing focuses on how the tool performs in situations that it was designed to handle. Evaluation of a tool is usually conducted by testing it against known data to provide confidence that a given tool is working as expected. If a tool is used in other situations, additional testing or verification will be needed. Testing has been demonstrated in computer science to be an effective method for revealing errors in tools. Testing provides confidence in multiple situations by eliminating known sources of systematic error.

6.2.1 The primary limitation of testing is that no amount of testing can prove that the tool is functioning correctly in all instances of its use. Even if all tests produce the expected results, a new test scenario may reveal unexpected results. In practice, the more testing of diverse test scenarios, the more confidence you have that the software works correctly.

6.2.2 Another limitation of testing is that each version of a tool could have flaws that are unique to that version operating in a particular environment. As new operating systems, hardware, software, and protocols evolve and new applications emerge, tools are updated to address these new developments in IT. Tool testing is further challenged by the large number of variables related to the tool and environment in which it is used.

6.2.3 These issues relate directly to the discussion of populations (see 5.3.2) and deciding how much testing is enough is an active area of research in computer science. The amount of testing often depends on the application of the software. For example, safety control systems for nuclear power stations are tested more rigorously than other non-life critical systems. Tools and functions that address the integrity of the evidence need to be tested more rigorously than functions that can be verified by alternative methods, including manual inspection.

6.3 *Performance Verification*—Performance verification refers to checking a specific tool in the environment in which it is used to ensure it can perform its given function. This is not a repetition of the in-depth tool testing already performed, but rather a quick check that the hardware has not failed, that a piece of software can interact with the environment in which it is run, or that new copies of tools that have been received are working. This may consist of running a subset of the tests from in-depth tool testing. See also SWGDE Standards and Controls Position Paper.

6.4 *Training*—Training in forensic processes in general and in the specific tool used mitigates the risk that the tool is used incorrectly. In accordance with SWGDE-SWGIT Guidelines and Recommendations for Training, forensic practitioners should be trained on the tools they are using. Formal training can include classes. Informal training can include review of tool documentation and on the job training. See also SWGDE/SWGIT Proficiency Test Program Guidelines.

6.5 *Written Procedures*—Having written procedures mitigates risk by documenting the correct procedures so forensic practitioners can more easily follow them. Procedures can be updated to keep current with industry best practices, and to state the limitations of specific tools and in what situations they are unsuitable for use.

6.6 *Documentation*—Documentation mitigates errors by allowing for review of work performed and for supporting reproducibility. A forensic practitioner's work must be reviewable in a meaningful way, including repetition of the process to assess the reliability of the results. Following written procedures and documenting significant outcomes should cover the majority of a practitioner's work. It is also important to retain and review audit/error logs of digital forensic tools in order to assess whether they functioned properly or encountered problems. Thorough documentation is especially critical for situations not fully covered by standard operating procedures. When such exceptions occur, detailing the situation and how it was handled is essential for error mitigation analysis.

6.7 *Oversight*—Technical and management oversight of digital forensic processes mitigates errors by ensuring that practitioners are trained in the tools they are using, that tools are tested, that documentation is produced and that procedures are followed.

6.8 *Technical/Peer Review*—Technical/peer review mitigates error by having another qualified forensic practitioner look for errors or anomalies in digital forensic results. This is especially important if there are novel techniques used or outcomes or findings are outside of expected results.

6.9 *Use of Second Method*—The use of a second method by the forensic practitioner mitigates errors by verifying results. Common second methods include:

6.9.1 After acquiring a forensic image of a hard drive with a tested hard drive imager and write blocker, forensic practitioner uses cryptographic hashes to verify that evidence is unchanged.

6.9.2 Manual review of reconstructed files, such as from deleted file recovery or file carving.

6.9.3 Manual review of files identified by a hash as being part of a contraband collection.

6.9.4 Use of multiple tools such as virus scanners, which while providing similar functionality, work differently.

6.9.5 *Use of Multiple Tests*—Since most digital forensic processes are non-destructive, it is possible to repeat most forensic processes as many times as necessary without "using up" the evidence. The forensic practitioner can use multiple techniques or repeat specific processes (including peer review) on copies of the evidence because the copies can be verified to be identical to the original.

6.10 *Awareness of Past and Current Problems*—Digital forensics is a rapidly moving field. Forensic practitioners can mitigate errors by staying current with problems discovered in their laboratory and elsewhere. There are several sources including vendor blogs, conferences, listservs, forums, professional publications, and peer reviewed journals. Before relying on a particular source, forensic practitioners should carefully consider the reliability of the information and, when feasible, verify the problem for themselves.

6.11 *Error Rates*—The use of error rates can mitigate errors by showing the limits of a technique. Many digital forensics techniques, such as copying and cryptographic hashing, have very small error rates.

6.11.1 Other techniques, such as file recovery, have error rates that are dependent on multiple conditions present on the media, which are often unique to that piece of media. Therefore, it is not advisable to state an error rate for such techniques as it not likely to be relevant. There are cases where an error rate can be determined but techniques require a method to establish a baseline and may only be able to be applied in specific circumstances.[6] Error mitigation for these situations must employ other techniques, such as use of a tested tool (that reveals the tools limitations) or use of a second method.

6.12 *Context/Consistency of Data Analysis*—Context/Consistency Analysis mitigates error by checking that recovered or identified material makes sense. Does the data make sense in context? Is it in the expected format? For example, the tool purports to recover a JPEG file that further examination reveals is actually a PDF file.

6.13 *Other*—This is not an all-inclusive list of error mitigation strategies. Forensic practitioners should document and explain other strategies they employed.

7. Summary

7.1 Many processes in digital forensics have fundamental differences from those in other forensic disciples that make them unsuitable for error rate evaluations. As a result, relying solely on error rates is insufficient and potentially misleading as a method to address the quality of the science when applying *Daubert*-type factors to digital forensics. In general, assessing the reliability of scientific testimony goes beyond error rates to include whether results are the product of sound scientific method, whether empirical testing was performed, and whether standards and controls concerning the process have been established and maintained. Therefore, when applying *Daubert*-type factors to digital forensics, it is necessary to go beyond merely stating an error rate – it is necessary to perform a comprehensive error mitigation analysis that addresses potential sources of error and how they have been mitigated. Mitigation techniques will not be able to address every potential situation and the resulting error mitigation analysis should clearly state this.

7.2 Digital forensics is best served by a framework that guides practitioners to state the sources of potential errors and how they were mitigated in a disciplined manner. This document provides a disciplined and structured approach to recognizing and compensating for potential sources of error in evidence processing. This error mitigation analysis process involves recognizing sources of potential error, taking steps to mitigate any errors, and employing a quality assurance approach of continuous human oversight and improvement. This more comprehensive process for addressing error is more constructive to establishing the scientific rigor and quality of digital forensic results than merely seeking out an error rate.

7.3 In the face of ever changing technology, digital forensic practitioners can provide reliable results by continuing to apply and develop best practices that provide guidance for how to perform forensic processes across disparate technology landscapes. Best practices may include implementing an array of error mitigation strategies such as those listed above, the foundation of which includes competent personnel implementing tested and validated tools and procedures, and employing a quality assurance approach of continuous human oversight and improvement.

8. Report

8.1 The following are three examples for what an error mitigation report might look like, each quite different from one another. The purpose is to provide sample language and sample structures for the reports. The first is quite comprehensive and shows the full breadth of applying the error mitigation strategies. The second example addresses a more specific situation and has a more focused error mitigation report. The third is focused on addressing the use of a new technique within a forensic process.

8.2 It is expected that the reader will select from the examples to create a template that works well within their laboratory and is appropriate for the type of forensic process performed. The goal is to document and communicate the steps taken to reduce errors and expose areas where there is still a significant source of error. For example, the use of a non-tested tool should be obvious from an error mitigation report and would require additional explanation for why untested tools were used.

8.3 *Example Report One*—The case involves intellectual property theft and includes web-based e-mail and cell phone analysis.

Report:

Confidence in the results from the cell phone analysis, including conspirator's contacts from the address book and text messages with conspirators that included references to new product development is based on:

- Use of a tested tool: The tool, MobileImager version XYZ, was tested by NIST and by the lab; however NIST tested an earlier version and neither NIST nor the laboratory tested the model of phone in question, but both the NIST and the laboratory tests included other models from the same manufacturer. Testing showed that the tool could retrieve contact information and text messages. Anomalies found during testing were not relevant to this examination.
- Context Analysis: The tool returned well-formatted data.
- It is possible that not all contact information was recovered.
- Text message recovery is limited to what was still stored on the phone.
- Lab-based procedures, including training, documentation, and oversight, were followed.

Confidence in the results of the web-based e-mail analysis, including identification of e-mails that contained company intellectual property being directed outside the company, is based on:

- Internet Tool ABC and Other Internet Tool DEF were used to acquire the e-mail have been tested within the lab.
- Context analysis showed that the returned data was well formatted consistent with web-based e-mail.
- Or: Context analysis showed that attachments were not returned. Only header information and the e-mail message itself were returned but they were well formatted.
- It is possible that not all e-mails were discovered.
- Lab-based procedures, including training, documentation, and oversight, were followed.

[6] For an example of an error rate for a specific situation see: Garfinkel, S. L., et al., "An Automated Solution for the Multiuser Carved Data Ascription Problem," *IEEE Transactions on Information Forensics and Security*, Vol 5, No. 4, December 2010. Available online: http://simson.net/clips/academic/2010.TFIS.Ascription.pdf, 11 June 2014.

8.4 *Example Report Two*—During the course of a forensic examination, a new technique is developed to address a particular aspect of the examination. The technique may be developed in-house or brought in from outside. This example addresses error mitigation strategies appropriate to this situation.

In this case, files had been deleted using a known wiping program. Normally, not only are the files not recoverable, but the wiping program removes any trace of the deleted files, file names, and of the tool's activity. The laboratory develops a technique to recover the deleted file names based on a journaling capability of the file system. In this example, it is important to determine what files the suspect possessed and then deleted. The resulting tool is called Zombie Resurrection.

Step 1—Zombie Resurrection was used on a copy of the evidence and was able to find 50 file names for files that were not present on the drive.

Step 2—Since it appears that Zombie Resurrection might be useful for finding deleted file names, Zombie Resurrection was tested.

A controlled test data set was created with known content. The controlled test data set used the same operating system as the evidence.

The known wiping tool was used on the controlled test set to delete 100 files.

Zombie Resurrection was used on the controlled test set. The result was that Zombie Resurrection produced a list of 75 file names that had been on the system, but the list did not include 25 file names. There were no file names included on the list that had not been on the system.

Zombie Resurrection was deemed to be effective for finding deleted file names but cannot be used to claim that the list provided is complete.

Step 3—Documentation was written for Zombie Resurrection for both the use of the tool and for the testing performed.

Step 4—Zombie Resurrection and its documentation were given to a colleague to test on a similar system. The colleague got consistent results as the initial test. Because Zombie Resurrection uses a straightforward technique, the colleague was able to understand how it works and was able to conclude that it was unlikely for there to be errors in the implementation using the tool for this situation.

Error Mitigation Report—The novel tool, Zombie Resurrection, was developed and tested in-house, documentation written and peer reviewed in-house by a competent forensic practitioner familiar with digital forensic tools and techniques. It is best practice to have tested tools that produce repeatable and reproducible results and to have peer review for new techniques.

Other error mitigation strategies will be needed if the tool is applied more broadly. Additional testing will increase confidence in the reliability of the results and its applicability to other environments.

8.5 *Example Report Three*—In this case, digital forensics was used to find information about a criminal plot. One drive was imaged and deleted files were recovered. This example uses a table to be filled in by the forensic practitioner to document the relevant error mitigation strategies that were employed. A brief discussion of the fields in the table is provided along with a table that has been filled in.

Fields:

Mitigation strategies that apply throughout should be noted up front. Only when there are exceptions should these overall strategies be discussed for each process. For example, if the operator were trained on six of the seven tools used, that would only need to be noted when the seventh tool is discussed.

Techniques: Describe the underlying computer science techniques or algorithms employed.

Tool: List the tools used including all relevant versioning information

Techniques Mitigation strategy: Techniques may have relevant error rates. NIST will be providing analysis of error rates for common forensics techniques. Check www.cftt.nist.gov. Other sources of error rate information are valid to cite. If an unusual technique is employed, refer to relevant documentation and literature.

Since testing is a primary mitigation strategy, list what relevant test reports are available. Be sure that any referenced test reports are reviewed for problems or limitations encountered during tool testing that are related to the current forensic examination. If the specific version has not been tested, be sure to clear about this. The other mitigation strategies that were used should also be listed. It will be helpful to take the generic strategies and state how they were applied in this examination. It will probably be helpful to state that the tool was or was not used according to its documentation and is appropriate for the given situation.

Findings: List facts that show that the examination produced relevant findings and summarize any key issues related to error mitigation.

TABLE 1 Forensic Practitioner Documentation

Techniques	Technique Mitigation Strategy	Tools	Tool Mitigation Strategy	Findings
Write Blocking	The ability to block commands is well established in literature. See NIST report on write blocking	Writeblocker ABC, version 1.2.3	Drive type is XYZ, which Writeblocker ABC supports. Tool has been tested by NIST and this version (including firmware) by our lab. The lab testing included the relevant operating environment. Hashing was used as a secondary verification.	Confidence is based on use of tested tools, secondary verification, and adherence to lab-based mitigation strategies.
Drive Imaging	The ability to copy content from drives is well established in literature. See X and Y. See NIST report on hard drive imaging.	Driveimager DEF, version 5.6	Drive type is XYZ, which Driveimager DEF supports. Drive had HPA, which Driveimager DEF can acquire. Tool has been tested by NIST and this version by our lab. Hashes were verified. Operator has not been trained on Driveimager DEF, but is familiar with several other hard drive imaging programs.	Confidence is based on use of a tested tool and verification of hashes.
Deleted File Recovery (DFR)	The ability to recover files using metadata based tools is established. See NIST report on DFR testing.	Deleted File Recovery Tool GHI, version 7	Drive contained NTFS file systems, which Deleted File Recovery Tool GHI can recover. Tool tested by NIST (provide reference) and found to be able to recover files if there is little fragmentation. There is a possibility that the tool will join file fragments from different files to recreate a recovered file.	Confidence is based on use of a tested tool and manual inspection of the files that contained relevant search terms to eliminate incorrectly recovered files and adherence to lab-based mitigation strategies.

9. Keywords

9.1 computer forensics; Daubert; digital forensics; error mitigation; error rates

APPENDIX

(Nonmandatory Information)

X1. EXAMPLE ERROR ANALYSIS FOR SELECTED TECHNIQUES

X1.1 The purpose of this appendix is to show the relationship between the error rate of a technique and the systematic errors of an implementation. Several examples are presented. An error rate is stated for an algorithm and an analysis of possible implementation errors with strategies for mitigation of the implementation errors. The topics covered are:

X1.1.1 Hashing (X1.2).

X1.1.2 Hard Drive Imaging (X1.3).

X1.1.3 Hard Drive Write Blocking (X1.4).

X1.1.4 Deleted File Recovery (X1.5).

X1.2 *Hashing*—Use of hashing in a forensic context is usually used to determine if a file has changed (for example, image of a hard drive) or if a given file is exactly the same as some known file.

X1.2.1 *Hashing Algorithm Error Rates:*

X1.2.2 Two types of errors that are possible are:

X1.2.2.1 Two files are the same but produce different hashes (false negative).

X1.2.2.2 Two files are different but produce the same hash value (false positive).

X1.2.3 The design of the algorithm is such that it always produces the same result for the same input, so the false negative rate for the algorithm is zero.

X1.2.4 Hash algorithms have a false positive error inherent in the algorithm design. The size (number of digits) of the hash value determines the false positive error rate. For example, consider a (not very useful) hash algorithm that computes a two decimal digit hash value. If 101 unique files are hashed then there must be at least two files with the same hash value. In practice, hash algorithms are designed to have a vanishingly small false positive rate near zero. The MD5 algorithm computes a 128-bit hash value, that is, 1 chance in 2^{128} of a given file having the same hash as another file chosen at random. The SHA1 algorithm is 160 bits with an even lower false positive rate.

X1.2.5 *Errors Implementing Hash Algorithms:*

X1.2.5.1 The implementation of a typical hash algorithm has several sections, including a section to input the data to hash and a section to compute the hash value. Some possible errors and implications include:

X1.2.5.2 Computer code to do the hash calculation may be incorrect. This type of error is readily apparent by software testing with a few files with known hashes. Most likely all the hashes will be incorrect. Such a tool is defective and a different tool should be used. An error rate for this implementation would be 100 %.

X1.2.5.3 The input section may change the data before passing the data to the program section that calculates the hash value. An example is that under certain conditions extra characters may be added by the operating system to the end of each line of text for text files. Such a tool incorrectly computes hashes for text files, but correctly computes hashes for other file types. This can be detected by software testing using a variety of file types including text files. Such a tool should not be used. An error rate for this tool could be calculated as a proportion of the text files relative to the total number of files. However, such a calculation would not be useful for any other case.

X1.3 *Hard Drive Imaging*—Hard drive imaging is the acquisition of the digital contents of a secondary storage device.

X1.3.1 *Hard Drive Imaging Algorithm Error Rates:*

X1.3.2 The basic algorithm for imaging a hard drive is:

X1.3.2.1 Determine the size of the target device.

X1.3.2.2 Read all readable data and save.

X1.3.3 The algorithm for reading data and saving it incorporates error correcting codes, which prevent reading data incorrectly. It is called a miscorrection when the error correcting codes do not produce the correct data. In accordance with *The PC Guide*: "A typical value for this occurrence is less than 1 bit in 1021. That means a miscorrection occurs every trillion gigabits read from the disk—on average you could read the entire contents of a 40 GB drive over a million times before it happened!"[7] In other words, the algorithm has an error rate that is zero for all practical purposes.

X1.3.4 *Errors Implementing the Hard Drive Imaging Algorithm:*

X1.3.5 Implementation of hard drive imaging tool is vulnerable to many systematic errors. Some examples:

X1.3.5.1 The size of the hard drive is determined incorrectly by the operating system or storage device reporting a smaller than actual size to the tool. The tool then stops the acquisition before all data has been read. This error is usually a consequence of a change in storage device technology. Tool testing can be used to detect this problem by using test drives that are the most recent available in addition to a mix of older drives.

X1.3.5.2 The size of the hard drive is determined incorrectly if the tool ignores hidden areas. This is often an intentional tool design decision and not really an error. Tool testing can detect this behavior by including test drives that contain hidden areas. This behavior can be mitigated by checking for a hidden area before imaging; if hidden sectors are present, another tool or technique can be used to reconfigure the drive to unhide the hidden areas.

X1.3.5.3 Some imaging tools offer a feature to restore a previously acquired drive image to another drive. Some operating systems under report the size of hard drives to the tool. In such a situation, the tool will stop the restore before the entire image has been restored. Tool testing will detect this error by testing with a restore drive exactly the same size drive that was imaged. This can be mitigated by always using a restore drive larger by the underreported amount than the original.

X1.4 *Hardware Write Blocker*—A hardware write blocker is a device used to connect a storage device to a computer that allows access to data storage device without altering the content of the device.

X1.4.1 *Write Block Algorithm*—The basic write block algorithm is:

X1.4.1.1 Intercept each command sent from the host to the storage device.

X1.4.1.2 Examine the command function.

X1.4.1.3 If the command could change content of the storage device, do not pass the command on to the storage device.

X1.4.1.4 For other commands, pass the command on to the storage device.

X1.4.2 The algorithm prevents any commands that can alter the content of the storage device being passed to the device. The error rate of the algorithm is zero; that is a perfect implementation would have no errors.

X1.4.3 *Errors implementing Write Blocking*—Some errors that can occur are:

X1.4.3.1 Not all possible write commands are blocked. Such a device may appear to protect a device as long as the host computer uses one of the blocked commands and then silently fail if the host computer uses one of the other commands that are not blocked. Tool testing can detect such errors by transmitting all known commands from the host to the storage device through the write blocker. The commands not blocked will always write to the storage device. This allows identification of a potentially unsafe write blocker and selection of a safe write blocker.

X1.5 *File Recovery:*

X1.5.1 Recovery of deleted files presents a tool user with a collection of recovered files, possibly with file sizes, names, MAC times, and other recovered metadata. Some of many possible recovery results are the following:

X1.5.1.1 A deleted file is recovered completely along with the file name and other metadata. This is the ideal case.

X1.5.1.2 A deleted file is recovered completely, but the file name and other metadata is not recovered. One situation when this happens is when some tools recover files from a Linux[8] ext2 file system.

X1.5.1.3 A deleted file is partially recovered sequentially from the first data block.

X1.5.1.4 A deleted file is partially recovered sequentially not including the first data block.

[7] Kozierok, C. M., *The PC Guide*, site version: 2.2.0, version date: April 17, 2001, available online: http://pcguide.com.

[8] A trademark of the Linux Foundation, Linus Torvalds, San Francisco, CA.

X1.5.1.5 A deleted file is recovered with some data blocks skipped. This scenario can lead to misinterpretation of results.

X1.5.1.6 A deleted file is recovered with some data blocks assembled out of order. This scenario can lead to misinterpretation of results.

X1.5.1.7 A recovered file contains data that was not present anywhere on the original drive. This would be a serious flaw in a tool; the tool has invented data.

X1.5.1.8 A recovered file contains data that was not ever present in a file, active, or deleted. This would be another flaw in a tool; the tool has included data that may not have been created or used by the drive owner.

X1.5.1.9 A recovered file contains data from multiple deleted files. This scenario can lead to misinterpretation of results.

X1.5.2 These results occur as a result of the interaction of the data available, the recovery algorithm, and the algorithm implementation. Before an error rate can be discussed, the error to be measured must be defined. There are many possible errors that can be defined and usually more than one way to define an error in the context of deleted file recovery. Many of the results listed above are really the best that can be done under the limitations imposed on tools by the data available. For this discussion, all the results other than the first result are treated as errors in the sense that the result is not a complete, accurate reconstruction of the original deleted file.

X1.5.3 Some examples of possible errors that can be defined:

X1.5.3.1 *Multiple Source Error*—Recovered file is constructed from multiple sources.

X1.5.3.2 *Size Error*—Recovered file is the wrong size. (The definition of the right size is not relevant for this example.)

X1.5.3.3 *Gap Error*—There are one or more missing blocks between two recovered blocks.

X1.5.4 Recovery is usually accomplished either by metadata based file recovery or by file carving. The algorithms used for each method are very different.

X1.5.5 *Metadata Based File Recovery*—Metadata based deleted file recovery exploits storage device characteristics, operating system behaviors, and file system behaviors that do not overwrite file data and may leave intact enough metadata to locate at least some of the file data.

X1.5.6 The actual deleted file recovery algorithm implemented by a given tool is often proprietary and not available for examination or analysis. However, the general approaches are well known and can be considered in light of known operating system behavior and limitations. A typical algorithm looks for metadata describing deleted files and then uses the metadata to locate the deleted data. As an example, consider the file allocation table (FAT) file system:

X1.5.7 *FAT*—When a file is deleted from a FAT file system, some metadata is immediately overwritten. The file entry is marked with a hex value of 0 × E5. This overwrites the first character of one copy of the file name. (However, there may be two copies of a file name: a disc operating system (DOS) 8.3 name and a long file name. The first character of the DOS 8.3 file name is overwritten, but the long file name remains intact.) The metadata that locates the first block of data and the file size is preserved, but the metadata to locate the remainder of file blocks is cleared to zero. This establishes limits that any algorithm recovering files from a FAT file system:

X1.5.8 The first block, the file name and the file size can be recovered immediately after a file is deleted.

X1.5.9 The actual location of the remainder of the file is unknown. However, it is possible to make a guess about the location of the remainder of the file because the operating system tries to avoid file fragmentation by allocating file blocks contiguously. Consider four layouts of deleted files at the time of data acquisition:

X1.5.9.1 The file data blocks are contiguously allocated.

X1.5.9.2 A file is fragmented such that the fragments are sequential and separated only by blocks from allocated files.

X1.5.9.3 A file is fragmented such that the fragments are sequential and separated by blocks from either allocated files or other deleted files.

X1.5.9.4 Once other file system activity occurs, overwriting of both metadata and file data may occur.

X1.5.10 *Some Simple Recovery Algorithms*—Here are three possible simplified algorithms for locating the remainder of file blocks when recovering files from a FAT file system:

X1.5.10.1 Include enough unallocated blocks following the first file block until the recovered file is the same size as in the deleted file metadata entry.

X1.5.10.2 Include enough blocks, regardless of allocation state, following the first file block until the recovered file is same size as in the deleted file metadata entry.

X1.5.11 Stop recovering after the first block.

X1.5.12 The following table describes algorithm behavior in terms of the multiple source error defined above on each of the four data layouts.

Algorithm	Layout			
	Contiguous	Frag/Active	Frag/Deleted	Overwritten
A	No error	No error	Multi source	Unknown*
B	No error	Multi source	Multi source	Unknown*
C	No error	No error	No error	No error

NOTE X1.1—If the original source were completely overwritten, from a single source, then the recovered file would be from a single source. If the original source were partially overwritten, then the recovered file would be from multiple sources.

X1.5.13 An error rate for each algorithm can be defined, but calculating the error rate is not really practical. For algorithm A, none of the files recovered from Layouts 1 or 2 have the multiple source error and all files from Layout 3 have the multiple source error. (Ignoring Layout 4), an error rate for a particular drive can be calculated by counting the number of occurrences of each layout. An estimate of the error rate could be estimated if a large corpus of drives were examined where the layouts were accurately known. However, there is not a practical way to know what the actual layouts are. The same considerations apply to Algorithm B. As for Algorithm C, the multiple source error never occurs. However, Algorithm C has the limitation that only the first block is recovered.

X1.5.14 Tool testing can give a general indication for what the deleted file recovery algorithm does for specific conditions and file systems.

X1.5.15 *File Carving*—File carving algorithms depend on the following characteristics of certain file types to determine the beginning and end of a file for carving:

X1.5.16 File types have a unique structure including a beginning marker (or signature) and an ending marker:

X1.5.16.1 File systems try to allocate file space contiguously.

X1.5.16.2 Files are allocated in cluster size units (multiples of 512).

X1.5.17 A typical file carving algorithm includes the following steps:

X1.5.17.1 Scan through unallocated space for paired file beginning marker and ending marker.

X1.5.17.2 Check for reasonableness.

X1.5.17.3 Collect the clusters between the two markers into a recovered file.

X1.5.18 For some file types, for example, pictures and videos, a visual examination can identify most incomplete or incorrectly recovered files. The picture does not display, the content is not recognizable or some similar result. For other file types, care must be used to examine the recovered file if data could be missing or come from multiple sources.

X1.5.19 For example, suppose a file is recovered that tracks web sites visited and the number of times a site has been visited. The format of the file is as follows:

Web site URL
';'
Unspecified other data
';'
Visit count
';'

X1.5.20 The following table displays that the original file has the following content:

Cluster Number	Content
0	Beginning marker
1	www.alpha.com;aaaaaaaaaaaa;5;
	www.beta.net;bbbbb;7;
	. . .
	www. How-to-chloroform.com;hhh
2	Hhhhh;1;
	www.irs.gov;xxxx;20;
	. . .
	www.trees.edu;tttttttttt;60;
	www.biology.edu;bbbbb;30;
	www.how-to-chlorophyll.com;ccccc
3	Cccc;74;
	www-movies.com;mmmm;8;
	. . .
	Ending marker

X1.5.21 If this file is carved and Cluster 2 is omitted, an incorrect inference about the interests of the user might be made.

X1.5.22 *Summary*—It is difficult to have a meaningful error rate for deleted file recovery tools. Tool testing can reveal the quirks of tool behavior and guide the tool user in areas where additional detailed examination can mitigate misinterpretation.