Reproducibility of Digital Evidence in Forensic Investigations

Lei Pan Lynn M Batten

School of Information Technology, Deakin University,
221 Burwood Highway, Burwood, Victoria 3125, Australia

{ln, lmbatten}@deakin.edu.au

Abstract

We present a three-component model of a digital investigation which comprises: determination of input-output layers, assignment of read and write operations associated with use of forensic tools, and time-stamping of read and write operations. This builds on work of several authors, culminating in the new model presented here which is generic, scalable and compatible with all functions in the system, and which is guaranteed to produce a high quality of reproducibility.

1 Introduction

With the development of modern computers and networks, computer-related crime has become a threat to society because of the immense damage it can inflict while at the same time it has reached a level of sophistication which makes it difficult to track it to its source. However, any computer crime leaves a trail of evidence in the form of digital information stored or transmitted on electronic components. In order to be usable as evidence in a court of law, such information needs to be captured in a systematic way without altering it in so doing. Thus, the process of identification and handling of the evidence is of prime concern in a forensic investigation.

In 2001, several organizations came together in the Digital Forensic Research Workshop (DFRWS) to establish a generic approach to the investigative process as applied to digital systems and networks. One of the outcomes was a proposal for a digital investigative process that comprises "the use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the purpose of facilitating or furthering the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations." [Palmer, 2001]

In determining the effectiveness of a scientific method approach, one of the key factors is 'reproducibility', that is, the ability to achieve a consistent level of quality throughout the investigative process, no matter how many times it is repeated under the same conditions. With respect to reproducibility, the forensic tools applied in the procedure must be of reliable, and measurable, quality, and the investigative process itself must be well formulated and logically laid out to achieve a best outcome.

In the present paper we pull together old and new ideas in establishing a model of input-output which is suitable for digital investigations where proof of reproducibility of the results is required. In developing this model, we draw on work of Gerber and Leeson [Gerber and Leeson, 2004] on the Hadley model of input-output layers evolved from the standardized Open Systems Interconnection (OSI) layers, on the work of Carrier [Carrier, 2003], who argues for the freedom to choose input-output layers appropriate to the requirements of the investigations, and of Hosmer [Hosmer, 2002] who proposes

the use of digital time-stamps to prove the integrity of digital evidence. Our model comprises a three-stage process including the determination of input-output layers, the assignment of read and write operations, and the time-stamping of those operations during the investigation.

In the next section, we detail the work of the authors mentioned above and describe how their work impacts on the goal of reproducibility in forensic investigation. In section 3, we describe our model of the digital forensic process in detail, explaining how it improves on past models concerning reproducibility. In section 4, we provide a forensic investigation example, applying several models to it and comparing the results in section 5. Finally, in section 6, we summarize the paper and propose future work which can be developed from it.

2 Reproducibility

To ensure a high quality of reproducibility, one has to reproduce, first of all, the necessary experimental conditions each time when a process needs to be repeated. In addition, during a case, a digital forensic investigator may have to deal with different types of digital devices and systems, and has many forensic investigatory tools to choose from, which further increases the complexity of the whole forensic setting. A high degree of formalization both of the system and of the process is therefore necessary to successful reproduction of results. However, as Gerber and Leeson [Gerber and Leeson, 2004] point out, digital investigation deals almost entirely with input-output, while no existing model for input-output formally proves the correctness of results to the level required for forensic investigations. They go on to state that "[f]orensic tools are, as a rule, presently tested based solely on their predictability: a full source code level audit of any tool, let alone any operating system component, to ensure precise and correct operation is basically impossible."

The Hadley model, introduced by Gerber and Leeson in [Gerber and Leeson, 2004], is an attempt to analyze input and output in both the hardware and software contexts in order to completely explain each data transformation of interest to the investigation. Although their starting point is the seven layer OSI model (physical, data link, network, transport, session, presentation, application), they argue that this choice of layers is not prescriptive, but should be flexible, depending on the investigation, and bound only by the following objectives:

1. "A layer should be created where a different level of abstraction is needed.

2. Each layer should represent a well-defined location or abstraction.

3. Each layer should be chosen with an eye toward representing well-recognized extant components of peripheral I/O.

4. The layer boundaries should be chosen to keep the information flow across interfaces between layers well ordered.

5. The number of layers should be large enough that distinct locations or abstractions need not be thrown together in the same layer out of necessity, and small enough that the model does not become unwieldy." [Gerber and Leeson, 2004]

In Figure 1 below, the hardware version of the Hadley model is presented with input denoted as a read operation and output as a write operation. Gerber and Leeson assume that the system sends no write commands whatsoever while in the forensic investigation mode. (We return to this issue in the description of our model.)

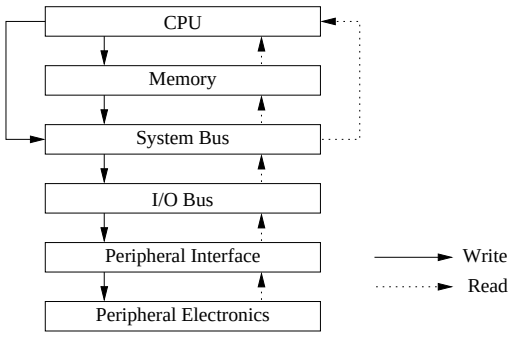

Figure 1: the Hadley Hardware Model

The intentions of the Hadley model are to "give computer forensic investigators access to a tool that

completely explains each of its salient transformations of data; ... (to) give computer scientists a complete, constructive, verifiable model of I/O; ... (and to) define translations between hardware ..." ([Gerber and Leeson, 2004]) and software layers of input-output. Only the first of these is relevant to the goal of reproducibility.

With the move away from the standardized OSI layers, Carrier [Carrier, 2003] argues for a completely open approach to the choice of layer and uses the phrase 'layer of abstraction' to indicate a generic layer chosen on the basis of several variables, including those of Gerber and Leeson listed above. Others might depend on the skill of the investigator, the investigation requirements, the tools available, and the system to be investigated.

Carrier's approach [Carrier, 2003] does not focus on the input-output (or read-write) operations, although these still play a role in his concept, but rather on the change in each layer as the data moves in and out of it. He addresses the reproducibility problem by introducing the idea of a margin of error on each layer which, according to him, can result either from tool implementation error or from the simplification to abstraction layers. Like Gerber and Leeson, Carrier also notes that 'read-only' mode is appropriate during an investigation.

Figure 2 below depicts Carrier's layer structure – the layer inputs consist of the input data and a rule set, and the layer outputs consist of output data and a margin of error.

Figure 2: Single Abstraction Layer in Carrier's Model

As noted in the Hadley model (point 4 above), it is important to preserve the correct order of read and write operations in order to obtain reproducibility of results in a computer system. For instance, a read operation returns an invalid result if it is executed before all the necessary information is up-to-date; a data transaction is untraceable if a write operation erases the content committed by the previous write operation before any read operation fetches the data. In a distributed system, the correct order of occurrence of each operation relies on accurate and precise time-synchronization between all entities. Hosmer [Hosmer, 2002] recognizes this as an issue for the forensic investigator and suggests the use of secure and auditable digital time-stamps when handling digital evidence in an investigation. He argues that such time-stamps can supply accuracy, authentication, integrity, nonrepudiation and accountability. While his work focuses on the sources of secure and auditable time, it does not examine the issues of where and on what time-stamps should be placed. Moreover, it is technically challenging to synchronize all clocks in a distributed system; however, we assume in this article that throughout an investigation, a trusted and reliable server is available through which the investigator is able to synchronize time.

In the next section, we introduce a framework for digital investigations which combines the above ideas into a more sophisticated model than seen previously.

3 A New Model

We argue that any model of a system for the purposes of forensic investigation should allow for reproducibility of conditions, but should also take into account the fact that the investigation may need to be scaled up to a larger system over time. The larger system may include devices not tested or searched in the earlier one, and so the following three properties of any model are crucial:

Generic. A model should cover all entities with potential use for storing or transmitting digital evidence, such as computers, cell phones, PDAs, firewalls, image display systems, routers.

Scalable. A model should scale well in incorporating new forensic tools into the investigation; existing tools should retain former behavior; when a new piece of digital evidence is identified, updates should be easily accomplished.

Compatible. A model should be compatible with all functions that the tools, system and network offer. Options for handling digital evidence should not be constrained by the model.

Description of the Model

In our model, we make the following assumptions:

1. a system process can always contact a trusted time server throughout the investigation,
2. the events of every operation of every process are uniquely time-stamped,
3. read and write operations are transacted without error of any kind,
4. every action performed by the investigator is safely and faithfully recorded.

It is also assumed that a forensic investigator begins an investigation with knowledge of the relevant procedural guidelines. Requirements and procedures may vary in different investigative settings. In any event, it is generally acknowledged that transient data are considered to be 'read' first in order to achieve high reproducibility, and so read operations are normally the starting point of an investigation process. Furthermore, the behavior of forensic tools in the system delineates paths of 'read' and 'write' operations.

I. Choice of the abstraction layers
We set up appropriate layers of abstraction on the basis of the following criteria, incorporating most of the Gerber-Leeson points:

a) a layer should be created only when it is needed,

b) each layer should represent a well-defined location or function,

c) each layer should be chosen to represent existing components based on input-output,

d) the number of layers should be large enough to separate essentially different locations and functions, but small enough to be manageable.

Although we drop point 4 of the Gerber-Leeson list, we describe a method of ordering information flow relative to a forensic investigation in the following two sections.

Additionally, we concur with Carrier's approach to the choice of necessary abstraction layers [Carrier, 2003]. Factors to take into consideration include:

- the expertise of the investigator
- the tools available
- the structure of the digital systems being analyzed
- any special requirements of the investigation.

At the beginning of evaluating an investigation, the points above are used in determining the abstraction layers.

II. Assignment of read and write operations
Once a set of layers is constructed, we define two operations 'read' and 'write'. A 'read' operation is performed when a digital sequence is accessed and the content is extracted; a 'write' operation is performed when a digital sequence is created, copied, or modified. A 'read' operation is denoted pictorially as an arrow starting from the layer in which the digital sequence resides and pointing towards the layer to which the content of the sequence is extracted. Similarly, each 'write' operation is represented as an arrow starting from the layer upon which the digital sequence resides and pointing to the layer on which the sequence is finally stored.

In order to obtain the highest reproducibility and so as not to damage digital evidence, the investigator should assign each operation based on the knowledge of how the forensic tool functions in the digital systems in which it is operating. Specifically, there are two points worth noticing:

- A 'read' operation is non-intrusive to any layers, but the information extracted will affect any future 'write' operation relying on the content retrieved by this 'read' operation.
- A 'write' operation alters the content of digital sequences in one or more layers. The content written should always come from a trustworthy and reliable source so that a back-track of information is always possible. When a 'write' operation occurs, we must take account of all the layers affected by it.

III. Time-stamping
Hosmer proposes in [Hosmer, 2002] to apply globally-synchronized time-stamps in order to maintain the integrity of digital evidence in a forensic investigation. Applying this idea, we define a

timing algorithm on 'read' and 'write' operations which allows us to linearly order time throughout the investigation as follows:

- Trusted time-stamps are obtained, recorded and applied during the entire investigation so that every forensic action is associated with an accurate and precise time.
- A 'read' operation is time-stamped at the beginning of a command to read information; a 'write' operation is time-stamped at the conclusion of the 'write' operation.
- If a time-stamped 'read' operation fails to start due to data-synchronization with some 'write' operations, then we delay dispatching this 'read' operation until the last 'write' finishes.

Applying the above algorithm, a sequence of time-stamps is generated on 'read' or 'write' operations. In an investigative scenario consisting only of 'write' operations, the reproducibility is compromised if a piece of digital evidence is written multiple times; but when 'read' operations are also involved in the scenario, time sequences assist in distinguishing the valid operations from invalid ones.

With respect to operations on each piece of digital evidence, there are four possible sequence pairs to consider:

1. Read-After-Write (RAW) means a 'write' operation finishes before we start to read the content. The content retrieved by the 'read' operation is what the 'write' performed.
2. Read-After-Read (RAR) means two 'read' operations retrieve information from the same version of the source.
3. Write-After-Read (WAR) means a 'write' operation changes the content what was read before.
4. Write-After-Write (WAW) means a 'write' operation happens after a previous 'write' operation. So the content of the previous write is erased.

Only two combinations have a negative impact on the reproducibility. The reproducibility of a WAR sequence is not maintained if the 'write' operation in the sequence depends on the content retrieved by the 'read' operation. For a WAW sequence, the reproducibility is not maintained if a 'write' operation deletes the content committed by a former 'write'. The reproducibility of an investigative process is guaranteed if neither of these two situations ever occurs. However, reproducibility is still possible when allowing all four possibilities by using time-stamps as described above.

Our model is designed to be generic, well scaled and highly compatible with existing system functions. Based on the ubiquitous existence of abstraction layers in digital data, forensic tools and digital systems, our model gives the freedom to the investigator to choose a suitable layered structure to cover all digital entities in his/her investigative case. The model achieves good scalability because newly introduced elements in the model do not incur any changes to operations performed by the investigator on existing ones. Finally, our model is compatible with all system functions as it is based on input and output.

4 An Example

We suppose that Jo, a forensic investigator, has received basic training on UNIX-like systems. Her forensic toolkit includes several statically compiled GNU programs: `lsof`, `ps`, `ls`, `fdisk`, `dd`, `nc`, `mkisofs` and `cdrecord`. She may use any of these programs to acquire digital evidence from a target machine named `pie`. This host is running a GNU Linux 2.6 system with a CD recorder and one IDE hard drive. To simplify the situation, we assume an ideal forensic environment.

The hard drive in the target system has two ext3 partitions: `/dev/hda1` and `/dev/hda2/`, where the disk usage shows:

```
Filesystem     Used Available Use% Mounted
/dev/hda1   3810876   5314868  42% /
/dev/hda2  26041508  36914748  42% /home
```

Jo identifies some suspicious files in a folder `/home/tunna/.pic/hidden/` which is mounted to `/dev/hda2`. In this example, her mission is limited to acquire an image file from the evidence hard disk, so she decides to use disk duplicate program `dd`.

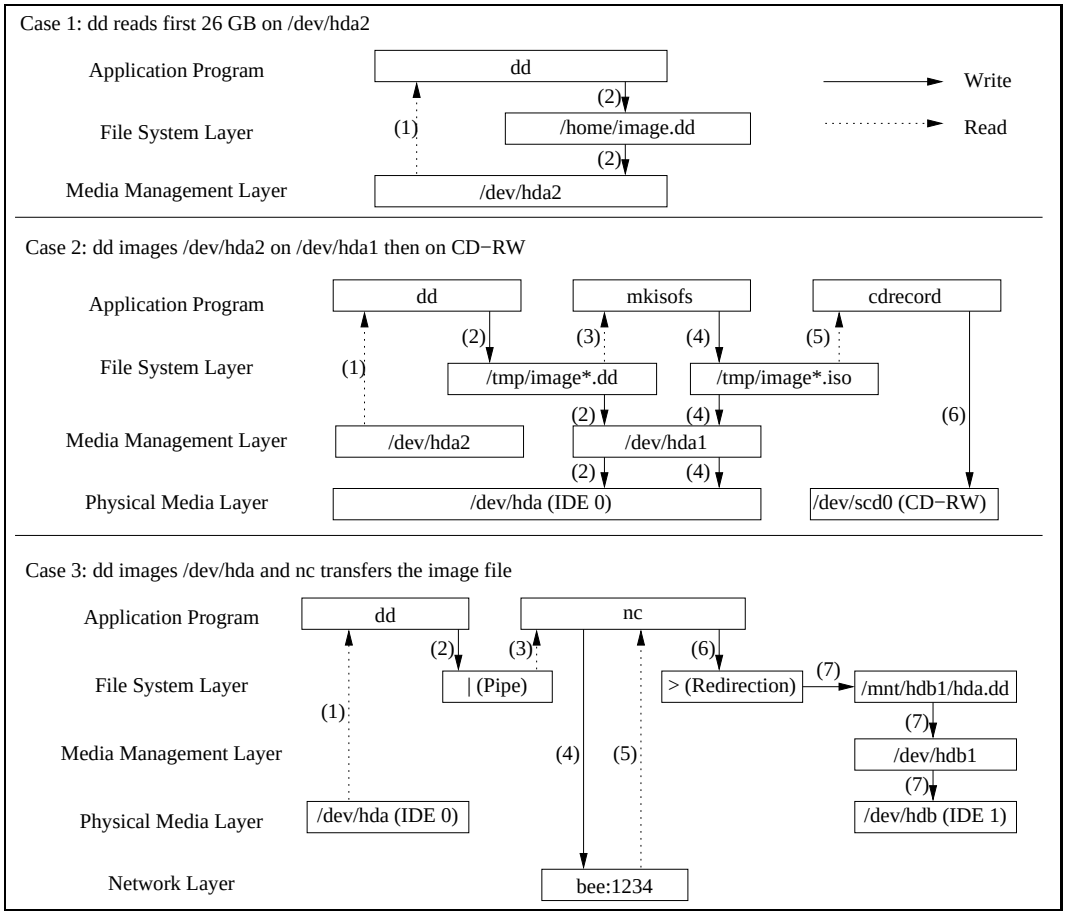

Figure 3: Our Model Applied to the Cases

Case 1 Suppose she mistakenly thinks that the evidence files are located in the first 26GB of `/dev/hda2`. Then she attempts to image this part of the partition and save the image file back to the same partition. The available free space will enable her to finish the following command.

```
pie #> dd if=/dev/hda2 of=/home/image.dd \
  bs=1024 count=26041508
```

In this case, the execution of program `dd` involves a disk partition as input and an image file as output. Therefore, Jo sets up 3 layers accordingly: the application program layer, the media management layer and the file system layer. The investigation does not require any other layers.

Then, she assigns a 'read' operation from `/dev/hda2` directly to `dd` and assigns a 'write' operation from `dd` to `/home/image.dd` and `/dev/hda2`.

As assumed in our model, all of Jo's actions should be faithfully recorded. According to the documented history, the 'read' operation of `dd` starts before the 'write' operation finishes. And the program `dd` starts reading immediately after the command line is typed in the console; thus two 'write' operations happen simultaneously. So Jo retains the order of operations as they are documented, which makes the 'read' operation the first and the 'write' the second (Figure 3).

Case 2 In this case, Jo realizes that the suspicious files are possibly scattered in the entire partition (`/dev/hda2`). Then she might image the partition piece by piece and burn the split image files to CDs. She uses some free disk space on the root partition to cache CD images before they are recorded.

```
pie #> dd if=/dev/hda2 of=/tmp/image1.dd \
```

```
  bs=1024 count=700000; \
  mkisofs -o /tmp/image1.iso \
  /tmp/image1.dd; cdrecord -v speed=4 \
  dev=0,0,0 -data  /tmp/image1.iso
....
(ignore other 89 steps)
```

In this case, `dd` behaves in the same way as in the first case; the input of program `mkisofs` are partition image files (`*.dd`) and the output are CD image files (`*.iso`); the program `cdrecord` has CD image files as input and CD recorder device `/dev/scd0` as output. Hence, Jo adds the physical media layer to describe disk devices in addition to the previous three layers chosen. The investigation does not need additional layers.

Then, she assigns a 'read' operation from `/dev/hda2` directly to `dd` and assigns a 'write' operation from `dd` to `/tmp/image*.dd`, `/dev/hda1` and `/dev/hda`. Program `mkisofs` 'reads' directly from `/tmp/image*.dd` and 'writes' to `/tmp/image*.iso`, `/dev/hda1` and `/dev/hda`. And program `cdrecord` 'reads' from `/tmp/image*.iso` and 'writes' to `/dev/scd0`.

As in the first case, every action should be associated with a unique time. Then Jo sorts them based on the time values. Since every 'read' operation starts immediately, she preserves the chronological order of all the operations (Figure 3).

Case 3 If the suspect user **tunna** has tampered with the system partition `/dev/hda1`, then the second approach might destroy evidence of this. Jo therefore decides to image the entire disk, and she needs a place to store the disk image file. Suppose a trusted Linux file server (`bee`) is running on the same LAN with enough free storage space on its second IDE hard disk (`/dev/hdb1`) formatted with ext2 file system. Then Jo executes the following commands on both machines respectively.

```
jo@bee $> nc -l -p 1234 > /mnt/hdb1/hda.dd
pie #> dd if=/dev/hda | nc bee 1234
```

In this case, the input of `dd` becomes the entire disk and its output is redirected to the input of program `nc` through a pipe (|). On the host `pie`, `nc` takes piped data as input and generates network packets as output; on the host `bee`, the input of `nc` is network packets and the output is redirected to a file. Thus, Jo introduces the network layer in addition to the four layers of the second case. These five layers are now sufficient to describe her actions.

Jo assigns a 'read' operation from `/dev/hda` directly to `dd` and a 'write' operation from `dd` to (|). Program `nc` 'reads' firstly from `bee:1234` and 'writes' to (>), and then to `/mnt/hdb1/hda.dd`, `/dev/hdb1` and `/dev/hdb`; the second `nc` 'reads' from (|) and 'writes' to `bee:1234`.

Time-stamps give the chronological order as `nc` 'reads' from `bee:1234`, `dd` 'reads' from `/dev/hda`, `nc` 'reads' from (|), `dd` 'writes' to (|), `nc` 'writes' to `bee:1234`, `nc` 'writes' to (>), (>) 'writes' to `/dev/hdb`.

In fact, the Linux kernel synchronizes 'read' and 'write' operations on the pipe, so Jo moves the 'read' of `nc` on (|) after the 'write' of `dd`. Similarly, program `nc` blocks 'read' on port 1234 of host `bee`. Therefore, Jo moves the 'read' operation of `nc` appearing in the first command line on `bee:1234` after the 'write' of the second `nc`, and gets the reordered operations (Figure 3).

5 Analyzing the Example

The example of the previous section demonstrates the acquisition of digital evidence from a Linux machine. In the first case, the 'read' operation on the partition `/dev/hda2` happens before the 'write' operation which relies on the content from the 'read'. Hence, this indicates a WAR sequence on the evidence partition and reproducibility cannot be guaranteed. In the second case, two 'write' operations wipe out a large amount of unallocated disk space which possibly contains the remnants of earlier 'write' operations committed by the suspect before the investigation; so we find two WAW sequences at the physical media layer and the media management layer in each command line. In the third case, no WAR or WAW sequence is performed and therefore, our investigator knows that commands she executed do not affect the reproducibility of the investigation process.

Neither the Carrier model nor the Hadley model guarantees reproducibility in the above examples. Carrier's model lacks a mechanism to combine the behavior of several programs as required in the last two cases. The Hadley model permits consideration of programs, digital sequences and read-write operations within one system, but does not preserve

the reproducibility of the investigation process.

6 Conclusions and Future Work

We have introduced a new model of a forensic investigation which determines the layers on which the investigation takes place, sets up a linear path of 'read' and 'write' operations and arranges these operations providing a linear time flow. The new model provides a method of substantiating reproducibility of the investigation. This new model is generic, scalable and compatible with current computer and digital device systems.

Forensic analysts, who have solid background knowledge of forensic tools and underlying systems, could quickly discover an erroneous forensic action damaging the reproducibility by using our model.

In future work, we plan to develop this model to include the possibility of error-capture.

The authors wish to thank the referees for useful comments.

References

[Carrier, 2003] Carrier, B. (2003). Defining Digital Forensic Examination and Analysis Tools Using Abstraction Layers. *International Journal of Digital Evidence*, 1.

[Gerber and Leeson, 2004] Gerber, M. and Leeson, J. (2004). Formalization of computer input and output: the Hadley model. *Digital Investigation*, 1:214 – 224.

[Hosmer, 2002] Hosmer, C. (2002). Proving the Integrity of Digital Evidence with Time. *International Journal of Digital Evidence*, 1.

[Palmer, 2001] Palmer, G. (2001). A Road Map for Digital Forensic Research. Technical Report DTR - T001o -01, Digital Forensic Research Workshop.