

# Selective Imaging

## Creating Efficient Forensic Images by Selecting Content First

Diploma Thesis by Johannes Stüttgen

March 31th, 2011

**First Examiner:** Prof. Dr.-Ing. Felix Freiling  
**Second Examiner:** Prof. Dr.-Ing. Wolfgang Effelsberg  
**Advisor:** Andreas Dewald



---

**Ehrenwörtliche Erklärung:** Ich erkläre hiermit, dass ich meine Diplomarbeit ohne Hilfe Dritter und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt und die den benutzten Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Mannheim, 31. März 2011

---

Johannes Stüttgen

---

## Abstract

In an increasingly computerized world, the amount of digital evidence in criminal investigations is constantly growing. In parallel, storage capacities of digital devices scale up every year and overwhelm forensic examiners with masses of irrelevant data, which has to be filtered and processed to identify applicable evidence. The acquisition of digital evidence is called imaging, the resulting evidence containers are named sector-wise images. Images are an exact copy of all data on a digital device and thus very large. To reduce the amount of data, which investigators have to cope with, the acquisition phase has to be improved.

Selective imaging is the creation of partial forensic images by selectively acquiring only relevant data from digital devices. While selective imaging is already practiced on a per-file basis in some cases, we work on increasing the granularity of the selection process to enable the application of this technique everywhere, even in cases where complex recovery has to be performed. The resulting evidence containers require accurate provenance documentation and precise verification procedures, which we develop in the scope of this thesis.

The principles and procedures developed in this thesis are implemented in a prototype, which is able to create, re-import and verify partial images. The prototype is used to evaluate the benefits and practicability of selective imaging, which we verify by interviewing forensic practitioners.

The methods and software, developed in the course of this thesis, allow examiners to create and work with partial images, having the same reliability as sector-wise images.

---

## Zusammenfassung

In einer Welt, die immer stärker von Computern durchdrungen ist, wächst die Menge an digitalen Beweisen in polizeilichen Ermittlungen ständig. Gleichzeitig entwickelt sich die Speicherkapazität digitaler Geräte immer weiter, was Ermittler förmlich mit Massen von irrelevanten Daten überschwemmt. Diese Daten müssen gefiltert und analysiert werden, um verwertbare Beweise zu identifizieren. Die Sicherung digitaler Beweise, auch Imaging genannt, erfolgt durch die Erzeugung einer exakten Kopie der Daten. Die erzeugte Beweis-Datei wird sektor-weises Image genannt und ist sehr groß, da sie sämtliche Daten, die auf dem Gerät gespeichert sind, enthält. Um die Datenmenge, mit der Ermittler umgehen müssen, zu reduzieren, ist es nötig die Sicherungsphase zu verbessern.

Selektives Imaging ist das Erzeugen partieller forensischer Images, bei der relevante Daten vor der Sicherung selektiert werden. Nur die selektierten Daten werden dann in das Image geschrieben. Auf Dateiebene wird diese Technik heute bereits teilweise eingesetzt, ist aber nicht immer anwendbar, da die Resultate komplexer Wiederherstellungsoperationen nicht immer Dateien sind. Wir entwickeln Verfahren für die selektive Sicherung mit beliebiger Granularität, um diese Art von Sicherung in allen Fällen anwenden zu können. Zur Sicherstellung der Verlässlichkeit werden Verfahren sowohl zur Dokumentation der Herkunft von partiellen Images als auch zu ihrer Verifikation entwickelt.

Die festgestellten Prinzipien und Verfahren fließen in die Entwicklung eines Prototypen ein, der in der Lage ist, partielle Images zu erzeugen, zu importieren und zu verifizieren. Mithilfe des Prototypen werden die Praktikabilität und Vorteile des selektiven Imagings evaluiert. Durch Interviews mit Ermittlern aus dem Bereich der Digitalen Forensik fließen auch Meinungen aus der Praxis mit ein.

Die Verfahren und Programme, die im Rahmen dieser Arbeit geschaffen wurden, erlauben Ermittlern partielle Images mit der gleichen Verlässlichkeit wie sektor-weise Images zu erzeugen.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Task . . . . .	2
1.3	Outline . . . . .	3
1.4	Results . . . . .	3
1.5	Related Work . . . . .	4
<b>2</b>	<b>Prerequisites</b>	<b>6</b>
2.1	Digital Forensics . . . . .	6
2.1.1	Properties of Digital Evidence . . . . .	6
2.1.2	The Investigative Process . . . . .	7
2.1.3	Forensic Disk Images . . . . .	10
2.2	Tools of the Trade . . . . .	11
2.2.1	Forensic Formats . . . . .	11
2.2.2	Forensic Frameworks . . . . .	15
2.3	Summary . . . . .	18
<b>3</b>	<b>Selective Imaging</b>	<b>19</b>
3.1	Selective Acquisition Procedures . . . . .	20
3.1.1	Acquisition Process Model . . . . .	21
3.1.2	Granularity . . . . .	23
3.1.3	Benefits and Applicability . . . . .	25
3.2	Partial Images . . . . .	30
3.2.1	Definition . . . . .	30
3.2.2	Provenance Assurance . . . . .	32
3.2.3	Legal Considerations . . . . .	34
3.3	Summary . . . . .	37
<b>4</b>	<b>Implementation</b>	<b>39</b>
4.1	Selection of Technical Foundation . . . . .	39
4.1.1	Analysis Framework . . . . .	40
4.1.2	Storage Format . . . . .	41

4.2	Architecture . . . . .	42
4.2.1	Framework . . . . .	43
4.2.2	Format . . . . .	44
4.2.3	Components . . . . .	45
4.3	Implementation Details . . . . .	48
4.3.1	Data-Copying . . . . .	48
4.3.2	Provenance and Meta-Data . . . . .	49
4.3.3	Image Creation . . . . .	51
4.3.4	Image Parsing . . . . .	54
4.3.5	Provenance Verification . . . . .	56
4.4	Development Facts . . . . .	57
4.5	Tool Usage . . . . .	58
4.5.1	Image Creation and Import . . . . .	58
4.5.2	Verification . . . . .	60
4.6	Summary . . . . .	61
<b>5</b>	<b>Evaluation</b>	<b>63</b>
5.1	Quantification of Benefits . . . . .	63
5.1.1	Test Data . . . . .	63
5.1.2	Disk Space Requirements . . . . .	64
5.1.3	Speed of Investigation . . . . .	64
5.2	Technical Details . . . . .	66
5.2.1	I/O Speed . . . . .	66
5.2.2	Reliability . . . . .	68
5.2.3	Disk Wearing . . . . .	69
5.3	Practical Acceptance . . . . .	70
5.3.1	Interviews with Forensic Examiners . . . . .	71
5.3.2	Quantification of Examiner-Opinions . . . . .	76
5.4	Summary . . . . .	79
<b>6</b>	<b>Conclusion</b>	<b>81</b>
6.1	Summary . . . . .	81
6.2	Future Work . . . . .	82
<b>A</b>	<b>Questionnaire</b>	<b>83</b>
<b>B</b>	<b>Source Code</b>	<b>88</b>
<b>C</b>	<b>Live DVD</b>	<b>89</b>
<b>D</b>	<b>Installation and Usage Manual</b>	<b>90</b>
	<b>Bibliography</b>	<b>91</b>

# List of Figures

2.1	The Investigative Process . . . . .	8
2.2	The EWF-E01 Format . . . . .	12
2.3	The AFF Format . . . . .	13
2.4	An exemplary RDF Graph . . . . .	14
3.1	The Selective Imaging Process . . . . .	22
3.2	Levels of Granularity . . . . .	23
3.3	Time-Narrowness Comparison . . . . .	27
3.4	Partial Image . . . . .	31
4.1	DFF Architecture . . . . .	44
4.2	AFF4 Architecture . . . . .	45
4.3	Acquisition Procedure of the Selective Imager . . . . .	46
4.4	Import Procedure of the AFF4 Connector . . . . .	46
4.5	Verification Procedure for Partial Images . . . . .	47
4.6	Selective Imager Architecture . . . . .	47
4.7	Selection of Evidence in DFF . . . . .	58
4.8	Acquisition of Evidence in DFF . . . . .	59
4.9	Import of partial image in DFF . . . . .	59
4.10	Simple Image Verification . . . . .	60
4.11	Automated Verification of Partial Image . . . . .	61
5.1	Imaging of a 20GB disk, speed comparison . . . . .	65
5.2	Imaging of a 4GB flash-drive, speed comparison . . . . .	65
5.3	Imaging of a 20GB hard-disk without carving . . . . .	66

# List of Tables

3.1	Exhibits per Crime . . . . .	29
3.2	Exhibits per Group . . . . .	29
3.3	Characteristics of Provenance Metrics . . . . .	33
4.1	Features of open-source digital forensic frameworks . . . . .	40
4.2	Comparison of image formats . . . . .	42
5.1	Imaging Speed by Tool and Features . . . . .	67
5.2	Device Wear by Investigative Procedure (Kingston USB Device) . . . . .	69

# List of Listings

4.1	DFF open()	49
4.2	DFF read()	49
4.3	RDF Sample Meta-Data	50
4.4	Function getFileNodes() ( <code>acquire.py</code> )	51
4.5	Image creation ( <code>acquire.py</code> )	52
4.6	Extraction of metadata ( <code>acquire.py</code> )	52
4.7	Extraction of byteruns ( <code>acquire.py</code> )	53
4.8	Selective Imager Acquisition Code ( <code>acquire.py</code> )	54
4.9	AFF4 Image Stream Wrapper ( <code>aff4.py</code> )	55
4.10	AFF4 Node Class ( <code>aff4.py</code> )	55
4.11	AFF4 Connector import code ( <code>aff4.py</code> )	56
4.12	Partial Image Verification Code ( <code>aff4verify.py</code> )	57

# 1 Introduction

Since the commercial launch of the first personal computers in the seventies, computer use has become an integral part of the everyday life of many people and is constantly increasing. According to a survey of the Pew Research Center in 2007, 80 percent of the U.S. citizens and 76 percent of the German citizens use computers at work, home or anywhere else, at least occasionally. These figures have grown by 7 percent in the U.S. and 13 percent in Germany between 2002 and 2007 [51].

Many of these computers are networked and access the Internet on a regular basis. The number of Internet users has increased by 444.8% during the past 10 years to a total of 1,966,514,816 (which equals 28.7% of the worlds population) [44]. More and more important activities, like banking or voting, are carried out with the help of computers. Because many computers are networked, new possibilities for scam, fraud and other criminal activities emerge, where criminals do not have to be physically present.

Also, with the proliferation of smartphones and tablet pcs, computers are often carried around by their owners. Crimes that are committed in the physical world thus often leave traces on digital systems, for example the movement profile of a cell phone owned by a suspect can indicate his physical location at a given time.

These developments cause computers and the data which is stored on them, often to contain evidence, valuable for criminal investigations, even if the actual crime was not carried out by use of a computer.

Digital forensics is a scientific discipline that deals with the identification, collection and analysis of evidence on digital systems. For the collection of digital evidence, a technique called *imaging* is regarded as the standard procedure. In this process a block-wise copy of the data on a digital storage device is created. The copy is then written into a file, called the *image* of the digital device. Analysis is then conducted on the image, not the original device. This mitigates the risk to damage the device and prevents its accidental alteration. However, the process takes a certain amount of time, during which analysis is not possible.

## 1.1 Motivation

Storage capacity of computers has been rapidly increasing over the years and will even more in the future [33]. However, the bandwidth to transfer this data is increasing significantly slower. Patterson discovered in 2004, that there is a gap between bandwidth and capacity that is constantly growing [49]. Hard-disk bandwidth is doubling every 2.7 years but the capacity is increasing by 240% during this period. This gap is becoming a big problem for digital forensics as most steps are I/O-bound [55]. Especially the

traditional imaging process is linearly dependent on I/O-bandwidth, which will cause the time in the overall investigation process required for imaging to steadily increase over the years. This is already a problem today. The time necessary to image a 2 TB hard-disk, using modern imaging equipment with a bandwidth of approximately 70MB/s, easily exceeds 8 hours, which is the duration of a standard working day in Germany. This means analysis will often be delayed at least for one day after digital evidence is discovered. In time-critical cases like child abductions or terrorist threats this delay is disastrous as the acquired devices might contain information which can save lives.

Another trend, which is problematic for digital forensics, is the increasing public use of cloud computing. In cloud computing users store their data on multiple remote systems, which are controlled by a service provider. Data from services like Facebook or Google Documents can not be acquired by traditional imaging, as taking the systems down that run the service disrupts the business of a company which is not involved in the case. Also, the acquisition of a sector-wise image seizes data which belongs to innocent people, whose privacy would be violated by such an action.

These issues result in an increasing amount of cases, where the traditional acquisition process cannot be applied. Forensic examiners in such cases are forced to deviate from standard procedures and improvise the acquisition. Usually they tend to selectively copy data with standard copying tools like the Microsoft Windows Explorer or Robocopy. Nevertheless, this process is not at all standardized and violates established forensic principles. As Richard III and Roussev predicted for forensic procedures [56] as well as digital forensic tools [55], it is inevitable to revise current acquisition tools and methods to reduce the amount of data that has to be analyzed by forensic examiners.

## 1.2 Task

To reduce the amount of acquired data, examiners can adopt a selective approach. The principle »Selection before Acquisition« [11] stipulates, that examiners analyze the data on digital devices quickly before acquisition. They will then acquire data which seems of relevance to the investigation into a so called *partial image*. This thesis aims at developing methods and software, which enable examiners to employ this approach in a forensically sound manner. Also, the possible abstraction levels of partial image are examined and a general type is defined.

Not every single investigation allows the use of a partial image. Access to the original device might only be available once in some cases. If evidence is overlooked during acquisition, it is lost forever and the failure to acquire it can compromise the entire legal action. This thesis analyzes different classes of investigations and identifies the constraints that a selective acquisition approach implies.

For the findings to be legally utilized, it is essential that examiners can prove that the image is an exact copy of the data on the original device. With traditional images this is very simple, because they are identical to the data on the device. Checksums like MD-5 or SHA-1 can easily prove the correctness of this claim. A partial image can not be compared to the original in this way, because it is an incomplete copy. In the course

of this thesis, the extension of an existing storage format for use with selective imaging is investigated, which must provide a mechanism to prove the provenance of its content.

To perform a reasonable selection, examiners need to have a basic understanding of the contents of a digital device. This thesis develops a usage concept, which enables forensic examiners to gain an overview of the device and define the necessary constraints for the selection.

Finally, a prototype is implemented, which enables existing forensic tools to perform a selective acquisition. The prototype is used to evaluate the concept with forensic practitioners and to quantify the time- memory- and cost-savings, resulting from selective imaging.

## 1.3 Outline

In Chapter 2 we give an overview on digital forensics. The investigative process is introduced and the imaging step is explained. Furthermore, an overview on the existing formats and tools for digital forensics is given to illustrate the design decisions made in the implementation.

In Chapter 3 we explain the concept of selective imaging in detail. The investigative process is adapted to account for the specifics of selective acquisition. The granularity of selection is investigated and the storage concept for selectively acquired images, called *partial images*, as well as a method for provenance documentation are introduced. We examine the suitability of selective imaging for different types of criminal investigations and discuss the legal acceptance of partial images.

The implementation of the prototype is introduced in Chapter 4. We discuss the design decisions taken and introduce the architecture of the tools which were developed in course of this thesis. We also provide an insight into the actual implementation of the most important routines of the tools and explain methods of verifying the produced partial images.

The Evaluation of the developed tools is presented in Chapter 5. We measure the performance and reliability of the tools and quantify the benefits of selective imaging in exemplary investigations. Furthermore, we present the tools to forensic practitioners and evaluate the practicality of the concept and software.

Finally, the thesis concludes with a short summary and an overview on opportunities for future work in Chapter 6.

## 1.4 Results

We have developed a concept for selective imaging, where examiners are not bound to any level of granularity. The images we propose can contain any data object, that examiners believe to be relevant. Furthermore, we designed a concept for provenance documentation and verification, that grants the same level of reliability to partial images, as exists for common sector-wise images.

These concepts were implemented as an extension module for an open source digital forensic framework, which enables examiners to create partial images and store them in an existing format for forensic images. The images can be re-imported into the framework by using a connector module, which was created in the course of this thesis. In addition, a verification program for partial images was created. The program allows forensic examiners to verify the provenance of partial images at any time. The created software was integrated into a Live CD, enabling examiners to test it with minimal effort.

We evaluated the software with two test cases, which were chosen to represent good as well as bad conditions for selective imaging. We measured savings between 23.7 and 40 percent in time for the investigation, as well as 94.3 to 99.6 percent in space required to store the acquired images. The raw transfer speed of the selective imager was determined to reach about 42 percent of the fastest sector-wise imager. However, due to the drastically reduced data amount, the overall imaging duration still was determined to be significantly shorter.

Forensic practitioners support these results, as the majority of them believe the acquisition phase can be significantly shortened with a selective imager. While most examiners see a need for selective imaging, they expressed doubts in the legal acceptance and reliability of partial images. Nevertheless, most points that were made only apply to file level selective acquisition and are mitigated by the concept we developed.

## 1.5 Related Work

The overall goal of selective imaging is to enable forensic examiners to handle large volumes of data. During the last five years some research on selective imaging has been conducted. Kenneally and Brown already laid the legal groundwork for selective acquisition methods [39]. Turner proposed the methods »Selective Imaging« and »Intelligent Imaging« to implement these techniques [63]. The developed principles allow to select files from a filesystem and create partial images with accurate provenance documentation. However, this concept can not entirely replace sector-wise images. In cases, where a lot of difficult recovery has to be conducted, important evidence is left out of the selection, because relics of deleted files, for example, might exist outside of the structure of the current filesystem.

Another approach to reduce the amount of data which has to be acquired during imaging is »hash based disk imaging« [20]. This technique reduces the size of images by considering the entire corpus of data, which examiners have acquired from other cases, during the acquisition. Every image that is acquired is segmented into runs of blocks. These runs are identified by their unique hash. When a new acquisition is performed, the device is read in runs, for which a cryptographic hash is calculated. If the hash is found in the corpus, the data run does not have to be stored. Instead, a reference to the run, that already exists in the corpus, is stored. While this technique does in fact reduce the size of images significantly in the presence of a large corpus, it does have some drawbacks. First of all, examiners have to carry the corpus around for each acquisition. When acquiring an image on site, this can be problematic, if the corpus is too large to

fit in the internal storage of the examiners computer. Also, the technique can not reduce the time required for acquisition, as all data has to be read in any case to calculate the hashes.

## Acknowledgments

First, I would like to thank Prof. Dr. Felix Freiling for the opportunity to research such an interesting matter. I also thank my advisor Andreas Dewald, for always giving me great advice and feedback on both, the technical details and the textual elaboration. Furthermore, I would like to thank the Arxsys team, for the helpful support with the plug-in interface of their software and for many interesting discussions on the subject. Thanks also go to Michael Cohen for helping me with the API of libAFF4. I also am very grateful to Heiner Stüttgen, Nina Stadler and Christina Crombach, for proof reading this thesis. Finally, I want to express my acknowledgments to the people with Polizeipräsidium Hessen, German Bundespolizei and PricewaterhouseCoopers, for providing me with valuable discussions and feedback, that enabled the evaluation of my work.

## 2 Prerequisites

In this chapter the principles and methods of digital forensics that this thesis is build upon are presented. Section 2.1 gives a brief overview on the discipline. The special properties of digital evidence are described and a standard investigation model for digital forensics is introduced. The relevant steps in this process are highlighted and the concept of forensic images is introduced. In Section 2.2, the corresponding software is presented. We give an overview on the formats used to store forensic images and show some of the frameworks forensic examiners use to acquire and analyze digital evidence.

### 2.1 Digital Forensics

Digital Forensics is described by practitioners as »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« [48].

Aside from being focused on digital sources, digital forensics is very similar to other forensic sciences such as forensic ballistics, where firearms and ammunition is analyzed to prove or disprove the commission of crimes. Unfortunately, digital evidence is very different from physical evidence. Due to its specific properties it allows and requires distinct handling, forcing investigators to operate a very different process.

#### 2.1.1 Properties of Digital Evidence

Casey describes digital evidence as a very »messy and slippery« form of evidence [17]. There is no hard evidence, but only data on a digital device. This data is very volatile and can easily be modified. While this is also the case with most physical evidence, tampering with digital evidence is much simpler in most cases. For example to authentically forge entries in a paper log, a lot of proficiency is necessary. Nevertheless, Questioned Document Examiners are often able to distinguish between forgeries and valid entries by the analysis of handwriting [45]. Digital logs on the other hand are simple text files, which can be changed without leaving any trace. While there are mechanisms like cryptographic signatures that can offer protection against unauthorized changes, these methods base their authentication on the possession of a secret, the cryptographic key. If a third person manages to steal the key this person is able to produce signatures that are completely valid, rendering the mechanism useless.

Furthermore, digital data is always subject to interpretation. For example the order in which the bytes of a number are stored in memory is not the same on every computer. This order is referred to as *Endianness*. It distinguishes between little-endian where the most significant byte comes first and big-endian where the least significant byte is in the first place. Which system is used depends on the architecture, for example Intel-X86 systems use little-endian and IBM-PowerPC use big-endian [16]. When analyzing data in a criminal case it is important to interpret it the right way, as it can lead to serious consequences if for example the defendant stole 0x00000010 dollars from a bank account but one interprets the log as 0x10000000 because the system stored the amount in little-endian.

Another characteristic of digital data is that it is often stored in filesystems, which may fragment data. This makes the arrangement of blocks also a matter of interpretation. When forensic examiners try to recover deleted files, they could be partially overwritten by that time so evidence from this sources might be incomplete.

Also, digital evidence can not be directly examined: When a murderer leaves his shoe at a crime scene, one can look at the shoe, measure it's size, have a dog smell it and analyze it in almost any way necessary, without taking the risk of damaging it. If the criminal instead leaves his cellphone, the analysis can only be conducted by using the device to extract the interesting data. Usage of course wears the device, so there is a risk of damaging it in the process, destroying the digital evidence inside.

In contrary to physical evidence, digital evidence can be easily manipulated. A log file for example, is ultimately just a simple accumulation of text stored on a digital medium. Framing someone with a crime is just as easy as changing some bytes on a hard-disk.

Luckily for the investigators, digital evidence has another interesting property: It can be seamlessly copied. Forensic investigators can use this property to mitigate the problems resulting from the specific characteristics of digital evidence. This is achieved by constructing »images« of digital storage. Images are essentially files, which can be protected from manipulation or damage by software. Files can be duplicated easily, allowing for backups or distribution of copies to multiple examiners, who can then analyze them simultaneously. The details on image construction and storage can be found in Section 2.1.3.

### 2.1.2 The Investigative Process

Due to the inherent characteristics of digital evidence, investigators have to be very careful when handling and using digital evidence in their investigations. Many process models for digital forensics exist, one of the most commonly known ones is the *Investigative Process Model* [17]. It illustrates the investigation as a flight of stairs that have to be climbed, each step representing a specific phase in the handling of a digital forensic investigation. In Figure 2.1, this process is depicted. It is a very detailed model that covers the entire investigation and not only the digital forensics part. The Steps of interest to forensic examiners start with *preservation* and usually end with *reporting*. Sometimes, forensic experts also have to be present at the crime scene to identify objects to seize. Also it can become necessary for them to testify in court.

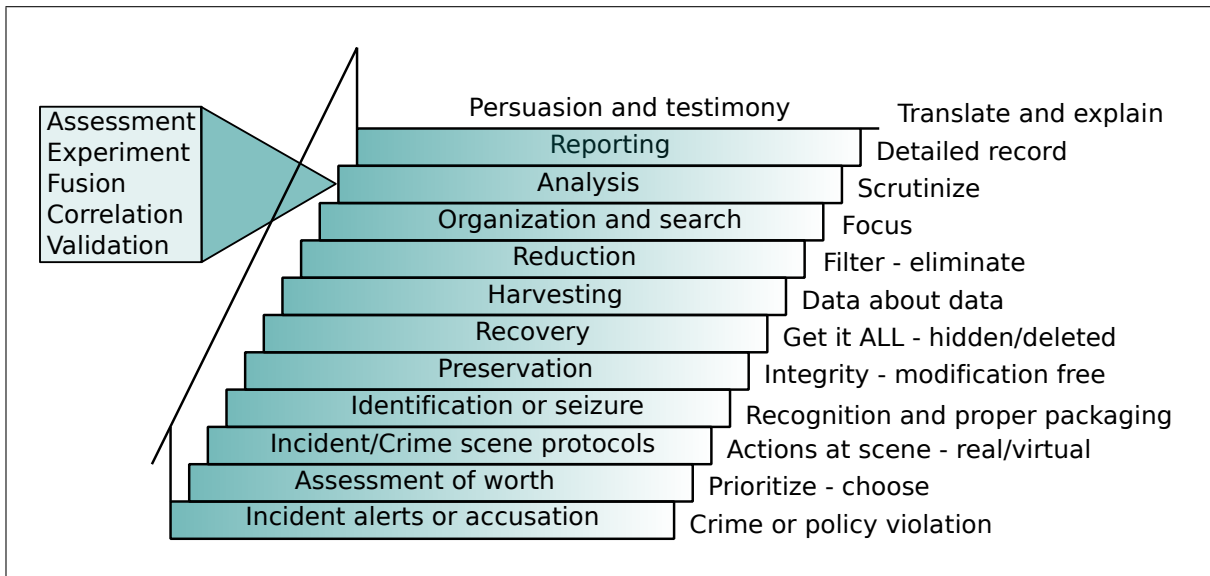


Figure 2.1: The Investigative Process [17]

While a complete explanation of this model goes beyond the scope of this thesis, the central steps are briefly presented to illustrate the placement of the imaging process in the model.

### Preservation

This step of the process focuses on the integrity of digital evidence. Due to the volatile nature of digital evidence this is a very important objective. At any point in the investigation, investigators have to be sure that the data they're looking at is exactly the data that was contained in the digital devices seized in the previous step. Measures to assure that the investigators can not accidentally change data during their analysis have to be taken. Failure to assure this might cause the results of the following steps being rejected in court.

### Recovery

Due to the way file-systems work, a lot of data that has previously been deleted by the user can be restored. Also the user might have tried to hide incriminating data somewhere on the device that can be recovered. Moreover, operating systems can decide to copy data from main memory to a persistent storage. This is called swapping and enables the operating system to run programs that require more memory than the system has available. Swapping uses a swap-partition on the hard-disk or simply a swap-file where the data from main memory is temporarily stored. Common operating systems do not delete this data so it can be used to recover the in-memory image of programs. The recovery step generally deals with the recovery of data from any of these sources, to provide investigators with the most exhaustive amount of data possible.

### Harvesting

This step is mainly geared at categorizing the massive amount of data that is available after the recovery step. Meta-data like the file extension, name and its location in the directory tree is gathered. This allows investigators to systematically analyze groups of data that are of specific interest regarding the case. If the investigation has a special interest in communication, the search could focus on chat-logs, web-mail-history or mail-client database files. If the focus is on the intrusion of a server, log-files or specific executables can be of interest. The result of this step ideally is an organized set of data that is relevant to the case.

### Reduction

After the existing data has been structured, it has to be reduced to an amount that investigators can examine more closely. The actual content of objects is still not likely to be considered. Instead, investigators make use of the meta-data harvested in the previous step, to eliminate as much irrelevant data as possible. The output of this step is the highest concentration of potential evidence that can be achieved without analyzing the content of individual data objects.

### Organization and Search

This step is the preparation for the actual analysis phase. Data is physically grouped to allow for a structured review in later stages of the investigation. It might also be indexed to allow fast and efficient searches in the contents. The aim of this step is to make the following analysis as structured and efficient as possible. This is important, as it influences the traceability of later findings, which directly affects their acceptance in court.

### Analysis

In this phase the examiners finally study the internals of individual data objects. This involves the reading of text, the watching of pictures and videos and the listening to audio contents. Findings are correlated and combined to validate or refute different hypotheses. This is the step in an investigation where the actual evidence that is used in court later on is found. The result are pieces of digital evidence that prove a certain course of events.

### Reporting

Since the final evidence resulting from the analysis phase was obtained by a large amount of processing and digital data is always subject to interpretation, it would be worthless without detailed documentation. The documentation must describe which steps exactly led to this specific result and that the methods and tools used were accepted standards. The resulting reports not only support the results of the analysis phase, they more

importantly make them traceable. The goal is to enable an expert witness under oath to come to exactly the same conclusions as the investigators.

The steps described in this model are very fine grained. A lot of other investigation models exist [52]. Recent models tend to be divided into fewer, more general phases [50]. However, most of them include the described processes in some degree. Especially the step of preservation is essential in any proposed model. Be it the *S-A-P* model (Secure-Analyse-Present) [32] or the *common process model for incident response and computer forensics* [25], all contain a step where evidence is preserved in some way. This is usually achieved by creating what is called a disk-image, which is elaborated in the next section.

### 2.1.3 Forensic Disk Images

Since it is possible to create identical copies of digital evidence, forensic practitioners have adopted a common practice of acquisition. This process consists of three steps:

- i) The device containing the digital evidence (e.g. a hard disk or a cell phone) is connected to a forensic workstation by a write-blocker. The forensic workstation is a trusted computer that is equipped with digital forensic software. The write-blocker is a tool, that allows the examiner to read data from any connected device, while preventing any modification of the data that is stored on it.
- ii) A forensic workstation is used to read from the device and construct a perfect copy of the contained data at block level. Block level refers to the level of abstraction used to read the data. It is the lowest possible level and does not imply the parsing of any information on the device. The different levels of abstraction are detailed in Section 3.1.2. The copy is stored in one of the many available formats described in Section 2.2.1.
- iii) A cryptographic hash of the data on the device is acquired. A cryptographic hash is a function that maps an input of arbitrary length to a fixed length output referred to as hash-value [30]. This hash-value has a very high probability to be unique for any given input and can thus later be used to verify that the copy has not been altered.

The copy of the digital evidence is further referred to as *image*. Since it is identical to the data on the original device, all analysis is conducted on the image to minimize the risk of damaging or modifying the original.

The write-blocker is a very important tool, because it prevents accidental (or purposeful) alteration of data on the device. Many operating systems automatically mount hard-disks upon connection. During this process data might be written to the disk, for example timestamps of files might be altered or journal entries in the file-system changed. Also there are no guarantees on the behavior of software running on the computer used to create the image. Some installed programs might also access newly connected disks in an alternative way. Write-Blockers are components, be it software or hardware, that are

designed to prevent any alteration to a connected disk while allowing read-access. This is usually realized by sniffing the command-traffic on the control channel and filtering any commands that might be used to modify data.

The acquisition of a cryptographic hash such as **SHA-1** is also very important, as it is the only way to verify the acquired image later. If the image has been altered during analysis, a cryptographic hash obtained from this image will no longer match the one acquired during the imaging process. This can be used to prove the integrity of the image and thus the validity of evidence that was obtained from the disk.

Validity and forensic soundness of the evidence aside, there are also purely practical reasons to work on copies of evidence instead of the original disk. Images can be copied and shared among examiners to enable concurrent analysis and accelerate the process. The handling of multiple devices is simplified, a write blocker is no longer necessary if the file is reliably write-protected by the operating system (for example with the immutable flag on unix based systems). Also it is possible to create backups that can be used in case the image gets damaged.

In conclusion, imaging today is *the* vital step in the preservation phase. Every forensic tool is designed to work with images and it is currently regarded as mandatory in any digital forensic investigation. Errors or malpractice during this step can compromise the result of any later stage in the investigation, which might result in having to redo all steps beginning from the preservation phase. In the worst case, they could even force examiners to terminate the investigation.

## 2.2 Tools of the Trade

As Garfinkel recently elaborated, digital forensic tools and formats today are heterogeneous and unstandardized [27]. Almost any forensic software uses its own image storage format and commercial vendors often keep the specifications of their proprietary formats secret. Open-Source solutions on the other hand constantly try to reinvent the wheel and thus also have limited compatibility. However, the tool developed in the scope of this thesis is supposed to integrate into existing forensic frameworks and use established formats. This section thus gives a short overview on existing tools and formats to explain the development decisions made in Chapter 4. The overview is presented without judgment, as the comparison and review of the different tools is presented separately in Section 4.1.

### 2.2.1 Forensic Formats

Carrier describes three different types of forensic image formats [14]:

- i) A raw image. This is basically a binary file, containing every sector on the disk in their original order.
- ii) An embedded image. This is a raw image, in which meta-data such as hashes is interleaved with the raw data.

## 2.2 Tools of the Trade

- iii) A raw image that has its meta-data in an accompanying file.

For academic work, the use of an open format is preferred. Most formats presented in this sections are open standards. However one proprietary format is included in this selection because its specifications have been reverse engineered and there is an open implementation available.

## The dd format

An exact sector-wise copy of a device is sometimes called dd-image. This is due to the unix program `dd`, which was normally used to obtain such a copy. This format is essentially a raw image. It is a flat file, containing an exact copy of all sectors on the device in their original order. Today, there are many alternatives to `dd`, for example `dcflddd` [36] or `dd_rescue` [29]. They still create raw images, but also allow for on-the-fly hashing or better handling of damaged sectors.

## The EWF-E01 Format

The Expert Witness Format (EWF) was originally developed by the company ASRData for their tool SMART [5]. The Specifications were published in 2002, but are not available on their corporate website anymore. However, they are still available in the web archive [6]. This format is the foundation of the Encase E01 format, which was created by Guidance Software [35] and is the most commonly used format in commercial software today. Guidance Software did not publish the specifications, so most of the information on the inner workings of the format are based on reverse engineering. Metz developed an open source library for creating and parsing EWF-01-Files [41]. From the libraries documentation, most of the technical details on the EWF-E01 format are known.

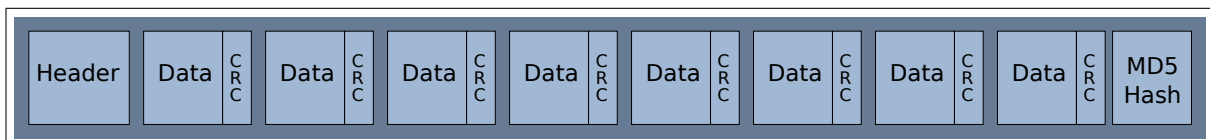


Figure 2.2: The EWF-E01 Format

An E01-Image basically consists of three parts: The header, the segments and the trailer. The header is used to store data about the image, like the name of the examiner who created it or a password. Data is stored in data-segments, which are 32 KB. in size, individually compressed and with a checksum attached to each segment. This way, read operations on small amounts of data do not require to decompress the entire image and corruption of a certain area does not affect all data. At the end of the image, a md5-hash is appended to verify the integrity. This concept is illustrated in Figure 2.2. Due to the integration of meta-data, the EWF-E01 format can be characterized as a format for embedded images.

### SGZIP

The sgzip format was developed by Cohen with the Australian Department of Defense for use with the PyFLAG framework [19]. It is basically a raw image that has been compressed with gzip [23]. The main difference is that it is seekable without decompressing the entire image. This is achieved by independently compressing blocks from the image, so only the blocks which are read have to be decompressed. Similar to the EWF-E01 format, the sgzip format compresses chunks of 64 blocks individually. Seeking to a specific position in the container only requires the decompression of 32 KB. of data. Despite it's advanced compression features, the sgzip format is essentially a format for raw images as it does not store any meta-data.

### AFF

The Advanced Forensic Format (AFF) is an open, extensible format developed by Garfinkel et al. [28]. The main reason for it's development was the lack of a suitable open format for forensic images. Raw images can not be compressed and at the same time be randomly accessible. Other formats like sgzip do feature seekable compression, but do not support the storage of metadata. Metadata is data about containers of data, in this case data about the image. AFF was developed to supports unlimited integration of meta-data, seekable compression, encryption and cryptographic signing of the image. It's implementation is freely available and open source. An open source library exists for several languages, to allow for integration into different forensic frameworks. Due to it's integration of various sources of meta-data, the AFF format can be categorized as an embedded format. To distinguish this format from its successor AFF4, the AFF format is also referred to as the AFF3 format.

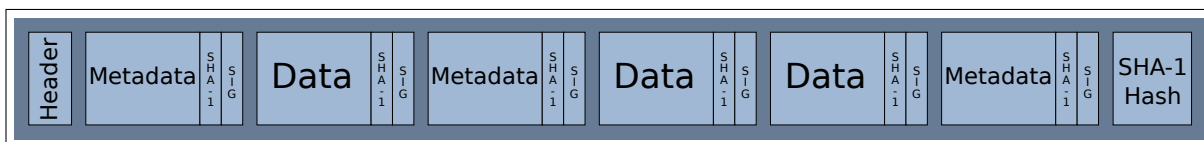


Figure 2.3: The AFF Format

The data organization is similar to EWF-E01, as can be seen in Figure 2.3. There is a header and trailer, but metadata is not stored in the header and the trailer stores a sha-1 hash instead of the md5 hash used in EWF-E01. In contrast to EWF-E01, there is a special segment type for metadata. These segments can store any type of user defined metadata, so the format is extensible. Data is stored in data segments, similarly to EWF-E01. The different segment types can be interleaved, which categorizes the format as an embedded format. To verify the integrity of individual segments, an individual sha-1 hash is stored. Also there is the possibility to store a cryptographic signature (SIG), which can identify the creator of a specific segment.

### AFF4

The AFF4 format is the successor to the AFF format [21]. The AFF format was completely re-designed, to overcome some of the inherent limitations. Investigations today often include several different systems with multiple hard disks. With traditional imaging, the evidence would be scattered into multiple files. The AFF4 format allows for multiple data sources in the same image, logically grouping related evidence into a single image.

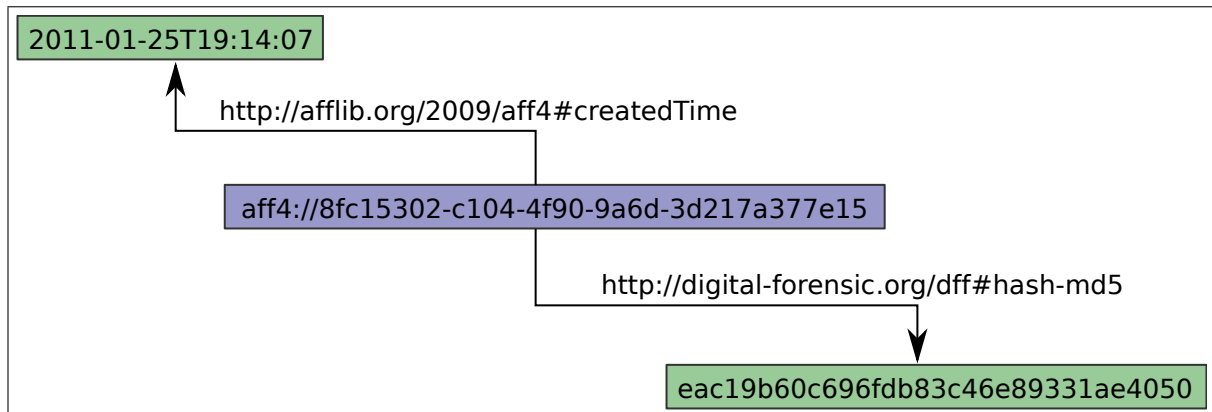


Figure 2.4: An exemplary RDF Graph

Each object in an AFF4 image is identified by a Uniform Resource Name (URN). The URN is unique for each object and is chosen randomly. Meta-data can be stored as RDF facts [68]. RDF facts are triplets, that define named *attributes* with a specific *value*, that are associated with a *subject*. Any user supplied facts are possible, enabling the examiner to store arbitrary meta-data like acquisition date, disk serial number or file timestamps with the image. To account for the provenance of the facts, they can be signed with x.509 certificates. RDF is visualized best as a directed graph, nodes being the objects and values, edges representing the attributes. An example of such a graph can be seen in Figure 2.2.1, blue nodes being objects and green nodes being values. RDF facts are serialized in turtle format [9] and stored within the image.

Objects can be any structured aggregation of data, the current AFF4 standard defines two important ones: volumes and streams. Volumes are basically containers for streams, at the moment they can be directories or zip-files. Streams are the interface to stored data. The most important types are segments, image-streams and map-streams. Segments are simple chunks of data. They can be individually compressed, thus seeking requires the whole segment to be decompressed. Therefore they are best suited for small files. Image-Streams store data in blocks that are individually compressed. The block-size is kept relatively small (usually 32 KB) to enable fast seeking in the stream. The blocks are grouped into so called bebies, their offsets are then stored in an index segment. Map-Streams are transformations of several arbitrary streams. They are realized by specifying a list of source offsets, target offsets and the source URN. This allows for zero-copy-carving, raid-reconstruction and any other logical rearrangement of data without having to actually duplicate anything.

Several other streams exist, implementing encryption or x.509 signatures. It is also possible to embed volumes into other streams, so images can be encrypted or signed retroactively. The technical details have been published at the Digital Forensics Research Workshop 2009 [21]. AFF4 is clearly an embedded format, as raw data is stored together with its meta-data in a single file.

### 2.2.2 Forensic Frameworks

A lot of different software exists in the field of computer forensics. There are small tools specialized in specific tasks, or bigger frameworks that target the entire investigative process. In this section, a number of all-in-one frameworks are introduced, that can be used for a wide range of computer forensic purposes. For academic work mainly open-source frameworks are relevant. We also introduce one commercial solution for comparison. There are currently two commercial frameworks that can be regarded as industrial standard (Encase and FTK), since they have a similar feature-set [7] there is no need to introduce both.

#### Encase Forensics

Encase is one of the most popular proprietary computer forensics frameworks [7]. It is used both in the private sector and by law enforcement agencies. It is a proprietary software sold by Guidance Software, Inc. Since the software, documentation and training are proprietary and cost money, it is rarely used in an academic context. However it is the de facto standard for both law enforcement and corporate investigators. In this section we classify the different features according to the steps in the investigative process introduced in Section 2.1.2. This information was obtained from Guidance Software.

For the *Preservation* phase, Encase supports the creation of several kinds of images. The main focus here lies on Guidance Software's EWF-E01 format. It does allow for automatic hashing of the image and secures the data with cyclical redundancy checksums. Aside the EWF-E01 format, Encase also allows the acquisition or import of raw images to be compatible with other tools.

The *Recovery* step is supported by numerous features. Deleted files are recovered automatically if possible and flagged as such. RAM dumps or unallocated blocks can be carved for the remains of files. File-carving is a technique to locate files in unorganized parts of a digital storage device, by searching for characteristic headers and trailers of known file formats. Parsers for different types of archives like `zip` or `rar` exist, that allow to extract data from them.

During *Harvesting*, forensic examiners can use several hashing algorithms to identify known files. File signature analysis can help to categorize data into different file types or formats. There are mechanisms to find browser histories, chatlogs or windows registry files.

*Reduction* can be achieved by creating bookmarks on files that seem promising. Also the creation of logical evidence containers is possible. They allow to create containers that hold only the files that seem worthy of further analysis.

For *Organization and Search*, Encase includes a search engine that can group files together that contain certain search terms. Parsers for logfiles, browser-histories and chatlogs enable those resources to be categorized, too. Data can be indexed to enable accelerated search later during analysis.

To help *Analysis*, viewers for about four-hundred different file formats are included. The (index) search can be used to find correlations between pieces of data. Pictures can be viewed as a gallery and windows registry files, system logs or email databases can be analyzed.

*Reporting* is supported by automatic report generation. Data about the acquisition and meta-data on the original devices can be saved. Also logs or browsing histories can be printed out as a report.

The entire framework is scriptable, every function that is accessible through the GUI can also be controlled this way. Scripts are created using a special language called “EnScript”. This language is the creation of Guidance Software and is used only with Encase.

### Autopsy and Sleuthkit

The Autopsy Forensic Browser [13] is an open source, web-based forensic framework. It’s functions are based on the Sleuth kit [15], which is a collection of tools for filesystem forensics. Many of the functions of commercial solutions like Encase are available. Especially when it comes to filesystem-parsing and the recovery of deleted files, the results are similar. Manson et al. compared the commercial solutions Encase Forensics and FTK [1] with the open source solutions Sleuthkit and Autopsy in 2007 and found that »the three tools provided the same results with different degrees of difficulty« [40].

There are some differences to traditional tools though. Since it is a web application, it can be simultaneously used by multiple examiners. If a central analysis server is set up, many examiners can connect to it with their web-browser and work on a case together (even from relatively slow terminals). Most steps in the investigation do not require the transfer of data contents, so there wont be too much traffic. Examiners can thus even work remotely, given proper encryption of their network connection.

Autopsy is not designed to be used for the *preservation* step of digital forensic investigations. Even though it supports a wide variety of image-formats, it does not have the capabilities to create them. There is also no file-carving feature, so it has to rely on external tools for this task such as foremost [2] or scalpel [54].

### PyFLAG

PyFLAG is short for Python Forensics Log Analysis GUI [19]. It was developed as a common tool for disk-forensic, memory-forensic and network-forensic tasks. Disk-Forensic in this context describes forensic processes related to the analysis of data which is stored in a persistent way, like on a hard-disk or flash-drive. Memory-Forensics does the same for data that is stored non-persistently, for example in random-access memory. Network-Forensic processes perform forensic investigations on network traffic.

PyFLAG's classification into the investigative process is similar to Autopsy, it is not designed for *preservation*. It is designed to be deployed starting from the *recovery* phase. The disk-forensic components are based on Sleuthkit, thus it's capabilities are very similar to Autopsy. However it does provide file carving capabilities and index-based searching.

Since PyFLAG is also a web-based application, the deployment on a central server is also possible. Multiple examiners can work on the same case simultaneously. If the investigative step they are performing does not require the transfer of big amounts of file contents, this can even be done remotely over the Internet.

What makes PyFLAG stand out from other forensic frameworks however, is the integration of memory- and network-forensics. PyFLAG fully integrates the Volatility Framework [64], enabling examiners to reconstruct address spaces of processes from memory. Furthermore, it is able to parse `pcap` files. `Libpcap` is a common library to capture packets from a network. While some tools exist to examine network dumps and dissect protocols (e.g. Wireshark [22]), until 2007 none existed that allowed to search and analyze huge amounts of them on a high level. Wireshark was not designed for forensic purposes and has trouble processing the very large amounts of data that are typical for forensic network analysis. Also it does not offer index searches or the carving of files from network traffic. PyFLAG allows examiners to use all the sophisticated tools intended for disk-analysis on network-dumps. It also offers reconstruction of webmail-sessions, file extraction and social network analysis.

### Digital Forensics Framework

The Digital Forensics Framework(DFF) is being developed by Baguelin et al. as an open source alternative to the big commercial products from Guidance Software or Accessdata LLC. It was designed to be modular and easily extendable. The framework itself provides only a GUI and the means to load and interact with objects of data. All real functionality is realized through modules, that can be applied to data objects in the framework. Since the framework is very young, it has not quite reached the extend of functionality available in big commercial solutions like Encase. Development is nevertheless advancing in a rapid pace and most core functions already exist, most importantly:

- Parsing of different file-systems (fat, ntfs, ext)
- Deleted file recovery
- File-Carving
- Analysis of memory-dumps
- Keyword searches
- Viewers for text, pictures and binary data
- Mobile phone forensics

- Image gallery

Any desired functionality can be implemented as a plugin in Python or C++, the framework can also be used in command-line mode and therefore be used in scripts.

Aside from raw- and EWF-01-images, DFF can also operate directly on devices attached to the system. The framework at the moment is not intended for use in the *preservation* phase due to the lack of an acquisition module. Since there is also no *reporting* functionality at the moment, DFF's classification in the investigative process currently spans from *Recovery* to *Analysis*. However, the team is working hard on changing this in future releases.

## 2.3 Summary

In this chapter the unique properties of digital evidence were illustrated. Because it is almost impossible to distinguish forgeries from real evidence and digital evidence always requires interpretation, it must be handled very carefully.

We presented a process model for digital forensic investigations, which illustrates the investigation as a flight of stairs that have to be climbed to complete it. The steps in this process that are of special interest to forensic examiners were elaborated and the similarities in the preservation phase of different forensic investigation models were explained. The industry standard used in this phase of the investigation called *imaging* was introduced and its significance for the rest of the investigation was shown. The second part of this chapter gave an overview on the different tools available to forensic examiners. First a set of formats for the storage of images (like EWF-E01, AFF and AFF4) were presented, followed by an outline of different all-in-one frameworks used by forensic examiners (for example Encase, PyFLAG and DFF). This set of tools will be compared in Chapter 4, when the foundation for the selective imager is chosen.

In the next chapter the idea of selective imaging is introduced. The investigative process is modified, to allow for the partial imaging of digital storage devices.

### 3 Selective Imaging

Current digital evidence acquisition procedures were developed in times, where cases usually involved a single computer with limited storage capabilities. The complete imaging approach, also referred to as sector-wise imaging, offered the benefit of absolute coverage, assuring that not a single piece of evidence could be missed. Circumstances have changed today and cases usually involve multiple computers with huge storage capabilities. The radical approach of acquiring everything that could potentially contain evidence is no longer practical. In the physical world, forensic examiners have come to accept that they can not achieve 100% coverage. Instead, their procedures include an on-site assessment of the surroundings of the incident, resulting in the acquisition only of objects that have a very high potential to be relevant to the investigation. To compare digital and physical forensic acquisition procedures, we will discuss the investigation of a common crime from both the physical and the digital forensics point of view. Digital forensics procedures of course can not be directly applied to physical evidence in this way. However, we will present these procedures strictly as if they were applied on digital evidence, for the sake of comparison.

Imagine a murder has been committed in a public building. Investigators arrive at the crime scene and use their standard procedures to try to solve the case.

The standard procedure for physical evidence proposes this process: The police tries to isolate the crime scene, preventing people from changing anything. Then the forensics team searches the scene for evidence, quickly assessing the value of all present objects and people for the case. Any potential evidence is taken to the lab for analysis. Potential witnesses are held for questioning and are set free afterwards. After the evidence is secured, the place is usually open for the public again. The actual analysis then takes place in the laboratory.

Using digital forensic procedures, the evidence acquisition will proceed a little differently: The police will take the whole building, including every person and object inside, with them to their laboratory. The forensics team will then go through the time-consuming task to produce a perfect copy of the building, the people and all objects inside. It is after this long process, that the investigators will finally commence investigating the actual crime by searching the copies for evidence. Depending on the circumstances, the building, objects and persons will not be released prior to the closing of the case.

While digital forensic procedures appear very excessive in this example, the analogy does not account for the specific attributes of digital evidence. Due to the reasons elaborated in Section 2.1.1, it does actually make a lot of sense to create copies of digital evidence. However, no actual reason exists to acquire copies of data that does not have anything to do with the actual case. Selective imaging is an acquisition method that

adapts the procedures developed over the years for physical evidence, but still accounts for the specifics of digital evidence.

#### Outline of the Chapter

In this chapter, the methods and formats that are necessary to perform selective imaging are discussed. In Section 3.1, the general procedure is introduced. Based on the *investigative process* by Casey, a process model for selective imaging is defined. The granularity of the approach is examined and its applicability to different categories of investigations is analyzed. In Section 3.2, a storage paradigm for selective imaging is presented. The term *partial image* is defined, methods for the provenance documentation of partial images are introduced and the legal implications are discussed.

## 3.1 Selective Acquisition Procedures

To successfully implement a selective imaging procedure for digital evidence, some key steps from the acquisition process for physical evidence have to be adapted. These steps include the search for relevant evidence directly at the crime scene and the preservation of only parts of a digital storage device.

The search for evidence directly on site in the physical world means that investigators look around the crime scene and quickly assess the value of objects in terms of evidential value. This enables them to perform a profound selection on which objects to take with them for further examination. The selected objects are then preserved in some way and taken back to the lab for an in-depth analysis. This procedure is already applied on a per-device basis in digital forensics today. Forensic examiners for example often discard media which are read only and obviously contain commercial content such as music CDs or operating system installation DVDs, if these contents are not relevant to the investigation. These decisions are made without actual analysis of the medium and only apply to rare cases where the content of a medium can be established without actually analyzing it. To be able to select on a more fine-grained level, examiners have to start assessing the content of digital devices by using digital forensic methods. Hard-disks for example can be analyzed directly, without the need to image them first by connecting them to a forensic workstation with a write-blocker. Forensic software can be used to determine the contents of the device. The examiner can select specific parts of the device that have a high probability to contain evidence, similarly to examiners selecting objects of physical evidence from a room or building.

To selectively acquire and preserve these data objects, two problems need to be solved. The first problem is packaging. The process for physical evidence stipulates that the object is packed into a plastic bag to protect it from being accidentally changed. This bag is then sealed, to protect the evidence from deliberate modifications. Finally, a tag is attached which describes the circumstances under which the evidence was acquired. Many of the formats currently used to store digital evidence do not qualify for similar measures. Some have limited metadata attached to the evidence, which allows a digital

form of tagging. Nevertheless, most of them can only store images of one complete device. Metaphorically speaking, this would equal a plastic bag that can only hold entire buildings but is not suited to carry multiple smaller items. To selectively store subsets of data on a digital device, the equivalent of an evidence bag is necessary, that can carry pieces of digital evidence. This concept was first introduced by Turner as *Digital Evidence Bags* (DEBs) [62]. Some implementations of formats that implement this concept exist, the specifics are evaluated in Section 3.2.

The second problem is the forensically sound acquisition of fractions of a digital device. Among the many reasons why the current procedure to acquire complete sector-wise images of digital devices is regarded as best practice, the easy verification of provenance is one of the most substantial ones. Since the image is identical to the contents of the device from a digital perspective, its integrity can be verified by simply comparing cryptographic hashes. Data at a specific offset in the image is found at the same position on the device, thus making the verification of findings from the image a straightforward comparison of specific addresses. When storing only subsets of data on a device, this matter becomes more complicated. The process of selectively acquiring digital evidence must accurately store provenance information for each data object in the image container. The specifics are detailed in Section 3.2.2.

#### 3.1.1 Acquisition Process Model

Searching for evidence prior to the preservation phase is not intended in conventional digital forensic investigation models. The sector-wise images allow forensic examiners to conduct the search right from the environment of their own laboratories. However, when performing a selection prior to acquisition, examiners need to have a basic understanding on what kinds of data reside on a device, to make qualified decisions on what to acquire and what not to. To put this in the context of the investigative process presented in Section 2.1.2, the *recovery*, *harvesting* and *reduction* steps have to be carried out before *preservation*. This does not necessarily mean they can not be repeated more thorough in the laboratory, but it is necessary to perform them roughly to gain an insight on what kinds of data reside on a device.

Figure 3.1 visualizes these changes in the investigative process. After *identification and seizure*, the devices that are suspected to contain digital evidence are connected to a forensic workstation through a write blocker. This does not necessarily have to be a hardware device, but can also be the read-only mounting of a network-share. If necessary, forensic examiners then perform the *recovery* and for example carve for data or reconstruct deleted files. After this step, *harvesting* is performed to gain knowledge on the type of data that is contained on the device. This knowledge is then used in the *reduction* step, where data that is irrelevant to the case is filtered out. These three steps are all performed in regard to the unique constraints of the case, ultimately discarding any data that examiners are certain will not be needed further in the investigation.

The resulting dataset from the *reduction* phase then gets acquired as a partial image. The details on partial images are elaborated in Section 3.2. Basically, a partial image is an image of only a fraction of data on a device, serving the same purpose as a sector-

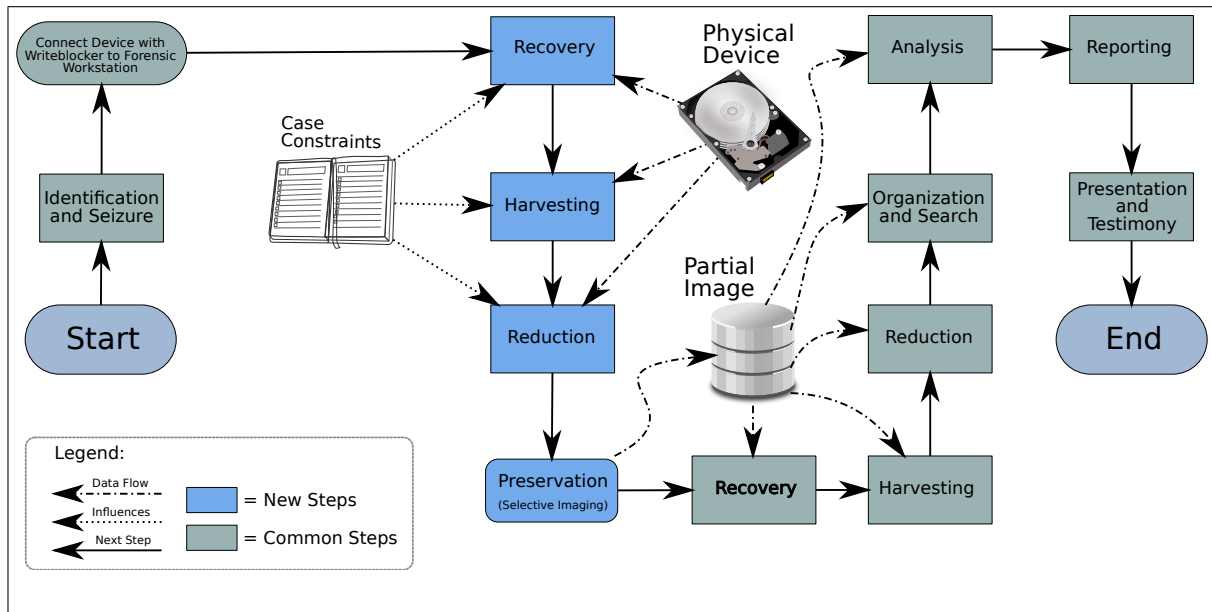


Figure 3.1: The Selective Imaging Process

wise image. It preserves potential digital evidence and tags it with information on the provenance of the data.

When the partial image is complete, the device can be disconnected and all further analysis can be conducted on the image. Since the initial assessment is done on site, it will often be coarse and not very precise. This is why in most cases the *recovery*, *harvesting* and *reduction* steps have to be repeated. However, they will complete much faster now because the data has already been filtered. The extent of this pre-filtering depends on the case. If the suspect is expected to be hiding information, only a small portion of data will be discarded. On the other hand, in a case where there is a demand for a specific type of information (e.g. e-mail), often none of the previous steps will have to be repeated because the required data is not hard to find.

#### Hybrid Model

Some cases will not allow for a strictly selective imaging process or force the selection to be very broad, because evidence might have been hidden or concealed. The preliminary analysis of the device is not extensive enough to discover every bit of evidence and examiners in some cases can not take the risk of missing anything because of the severity of the crime. In these cases, a hybrid approach between selective imaging and sector-wise imaging is possible. Initially, acquisition is performed by selective imaging. As soon as the preservation step is completed, examiners will initiate the creation of a sector-wise image. Since this process requires no interaction while running, examiners can then focus their attention on the image created with the selective imager. When analysis of this image is finished, examiners can continue working on the complete sector-wise image. This model will preserve the absolute coverage of the sector-wise imaging approach,

## 3.1 Selective Acquisition Procedures

while exploiting the time advantage of the selective imaging approach. It also mitigates the inherent problem of selective imaging, not to be able to come back to the original device for more data, if it becomes apparent that important evidence was omitted during selection.

### 3.1.2 Granularity

Digital devices store data encapsulated in several layers of abstraction. From the physical perspective, it looks like digital storage is simply a sequence of bits. In reality, computers utilize multiple logical concepts to manage their storage. Storage devices are divided into partitions, those in turn are organized in filesystems which contain files that again have their own format and internal management structure. For acquisition purposes, there are four relevant levels of abstraction:

- File Level
- Filesystem Level
- Partition Level
- Device Level

Each of these levels contain metadata related to the logical layout and management of the contained objects. Higher levels are encapsulated in lower levels, as can be seen in Figure 3.1.2. From the acquisition perspective, the data on a digital device is separated into different objects on different levels of abstraction, which can be selected for acquisition.

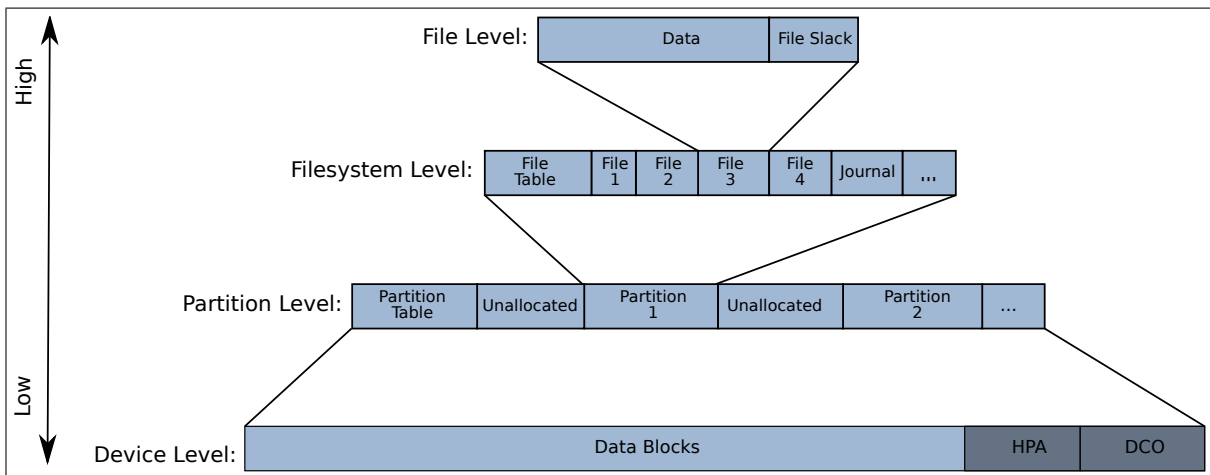


Figure 3.2: Levels of Granularity

Conventional forensic images are acquired at device level. Devices are regarded as indivisible objects that can either be imaged completely or not at all. Data on a device is simply a continuous stream of bytes that can be read sector-wise. Metadata on this level is the manufacturer of the device, its type and serial number. Some areas on the

device might not be directly accessible and will have to be unlocked. The ATA protocol, a common protocol to access digital storage devices, allows for an area called Host Protected Area (HPA). This area is hidden from the user until explicitly exposed by specific commands. Also recent specifications of the ATA protocol allow for something called Device Configuration Overlay (DCO), that can also lead to parts of the device being invisible (and inaccessible) to the user. The existence and position of such areas can also be considered metadata as they have to be made accessible before the device can be acquired completely.

On the next level, data is subdivided into multiple partitions. Size and location of the partitions is documented in a special data-structure, the partition table. Partitions do not have to cover the entire device. It is possible that some areas do not belong to any partition. They are called unallocated sectors. Unallocated sectors are not necessarily empty. The partitioning of a device can be deliberately changed, thus they could have belonged to a partition that existed earlier but has been deleted. When a partition is deleted, the actual data remains untouched and only the entry in the partition table is modified. Since normal system behavior does not affect unallocated sectors, they often contain old data that is believed to be deleted. This is why unallocated sectors are often valuable from a forensic perspective.

The filesystem level regards the internals of partitions. Almost all partitions are structured by means of a filesystem, except for some special cases. For example, swap partitions do not store files and thus do not need one. Filesystems are normally organized by a directory structure (e.g. the Master File Table in NTFS). This structure stores a list of all files, which sectors they occupy on the device and some additional metadata. The specific metadata that is stored depends on the type of the filesystem, common information includes filename, size, path, ownership, access rights and mactimes (mactime refers to the time a file was last modified, accessed and created). Some filesystems also maintain a journal on write operations. This information is used to prevent damage to the filesystem if the device gets disconnected or powered down during a write operation. This journal is a possible source of evidence as it contains a lot of information on filesystem activity. It is especially useful as it can be used to obtain a history of mactimes for files.

The file level is the highest layer of abstraction relevant for forensic acquisition. It consists of a number of files and some artifacts. Because most digital storage devices store data blockwise, files are spread out over the device in groups of blocks. The filesystem maps these blocks transparently to the user, so files appear to be a continuous stream of data. However, the size of files is rarely an exact multiple of the filesystems block size. Most filesystems also can not allocate single blocks for performance reasons, but allocate groups of blocks called clusters. That is why at the end of a file often a remainder called file-slack exists. The file-slack is the part of the last cluster of a file that does not contain any data related to the file. It is up to the implementation of the filesystem what is stored there. Most filesystems do not overwrite the file-slack, so forensic examiners might find pieces of files there that have already been deleted and overwritten.

Selection is generally possible on any level, but higher levels allow for more accurate segmentation. Users of digital systems explicitly interact with them on file level, which

would suggest this level as the most elemental form of selection. However, actions of users have implicit consequences to lower levels that can not be captured if these levels are not incorporated in the selection. If a user for example deletes a file, it is lost from the index of files and thus virtually does not exist on file level. Since most filesystems do not overwrite deleted files right away, some deleted files still exist on filesystem level and can be recovered. Also, metadata that records user actions is spread over different levels. Microsoft Office documents for example store information on the identity of the user on file level. Chronological information on the creation, modification or the last access of a file then again is stored on filesystem level. When choosing a level higher than the device level for selection, it is therefore important to additionally collect the metadata from lower levels and associate it with the according entities.

Some cases require an even finer grained approach. For example the extraction of fragments of data from the slack-space of a file, from unallocated space or even from space that is marked free in the filesystem can be important in cases where important evidence is expected to be concealed, hidden or deleted. A selective imaging approach thus can not be limited to objects on a specific level. It must always be possible to define new objects from an arbitrary amount of bytes on a device and then select them for acquisition. A selective imager has to be able to locate the relevant metadata from lower levels as described above, and image these objects the same way as it can image a file or partition. This possibility is essential to achieve the same potential that sector-wise images have.

#### 3.1.3 Benefits and Applicability

The most obvious benefit that can be gained from selective imaging is the ability to cope with very large cases. As Richard III and Roussev observed in a breakout session with forensic experts, data volume is increasing dramatically in digital forensic cases and is becoming increasingly difficult to handle [56]. Selective imaging can reduce this volume at the source. The most remarkable effect of this reduction is speed. Since not all data on a device needs to be imaged, the preservation phase is shortened. This can be a huge benefit in cases where results are needed very fast. On the other hand it even helps in cases where there is no time pressure, as the cost of the investigation scale with the amount of man hours spent on it.

Another notable effect is the smaller need for processing equipment and storage capacity in later stages of the investigation, as these factors are directly dependent on the amount of acquired data.

In conclusion, the overall gain of these effects are significant cost savings. These savings of course depend on the effectiveness of the selection. Selective imaging is a very general procedure. The granularity of the selection can be chosen freely, depending on the individual constraints of the investigation. The selection on device level can also be classified as an instance of selective imaging and the choice to image all devices is also a valid decision. Therefore, selective imaging can achieve equal coverage as the common procedure of sector-wise imaging. Actually, sector-wise imaging is an instance of selective imaging, where the selection is limited to objects on the device level. Depending on the

### 3.1 Selective Acquisition Procedures

---

type of the investigation, constraints on what evidence to gather vary from a very narrow subset of data to almost everything.

A recent analysis of police e-crime data in Australia identified several major areas of crimes that are of relevance to digital forensics [60]. The most significant categories are:

- Drug Trade
- Illegal Pornography
- Fraud
- Sex-related Crime
- Harassment
- Homicide
- Illegal Access to Computer Systems
- Terrorism

We will now analyze these categories to determine the applicability and gain, selective imaging can provide. For the assessment of the potential benefits, two criteria are of interest:

The first criterion is *time pressure*. Cases, where results are needed quickly, benefit most directly from selective imaging. Analysis can begin earlier and deliver results faster than with traditional sector-wise imaging. This can lead to success in cases, where otherwise the duration of the investigation would have lead to catastrophe or the escape of a criminal.

The second criteria is the overall *narrowness of the objective*. The more accurate the demands for specific types of evidence are, the easier is the reduction of the data before acquisition and the smaller the resulting image. This directly influences the extent of the time and equipment savings.

When analyzing the different categories of crime in respect to these criteria, they can be classified in three distinct groups:

Group A consists of crimes that often need to be solved under high time pressure but also can not be constrained much in regard to what kind of evidence is looked for. In a case regarding terrorism for example, investigators often need results very fast because there are many lives at stake. In homicide cases time also plays an important role, because dangerous individuals can cause a lot of harm if not apprehended quickly. In both cases, it is not easy to narrow down the type of data that is needed because any small detail might be important. Illegal access to computer systems is even worse in this matter, because individuals capable of such can be expected to have an above average knowledge of computers and thus might employ more elaborate methods to hide potential evidence. Time pressure however, is often a big issue in access crimes because the tracing of persons over a network gets more and more difficult the older the trace

### 3.1 Selective Acquisition Procedures

is. For instance, logs at telecommunications providers in Germany are not stored for a long time. Since they are subject to data protection laws, telecommunication providers like Deutsche Telekom usually delete them after a period of a few days [24]. Any traces older than this period can thus no longer be resolved. In conclusion, this group benefits the most from acceleration of the acquisition process, but its characteristics result in the smallest reduction of investigative duration compared to other cases. Given the severeness of the crime it can often be possible to seize devices permanently or at least for a great amount of time. A hybrid approach is then possible, where initially a selective image of the most promising data is obtained. While analysis is then conducted on the selection, a second acquisition is performed, covering all data on the device. This approach shortens the initial period to obtain preliminary results, without sacrificing completeness.

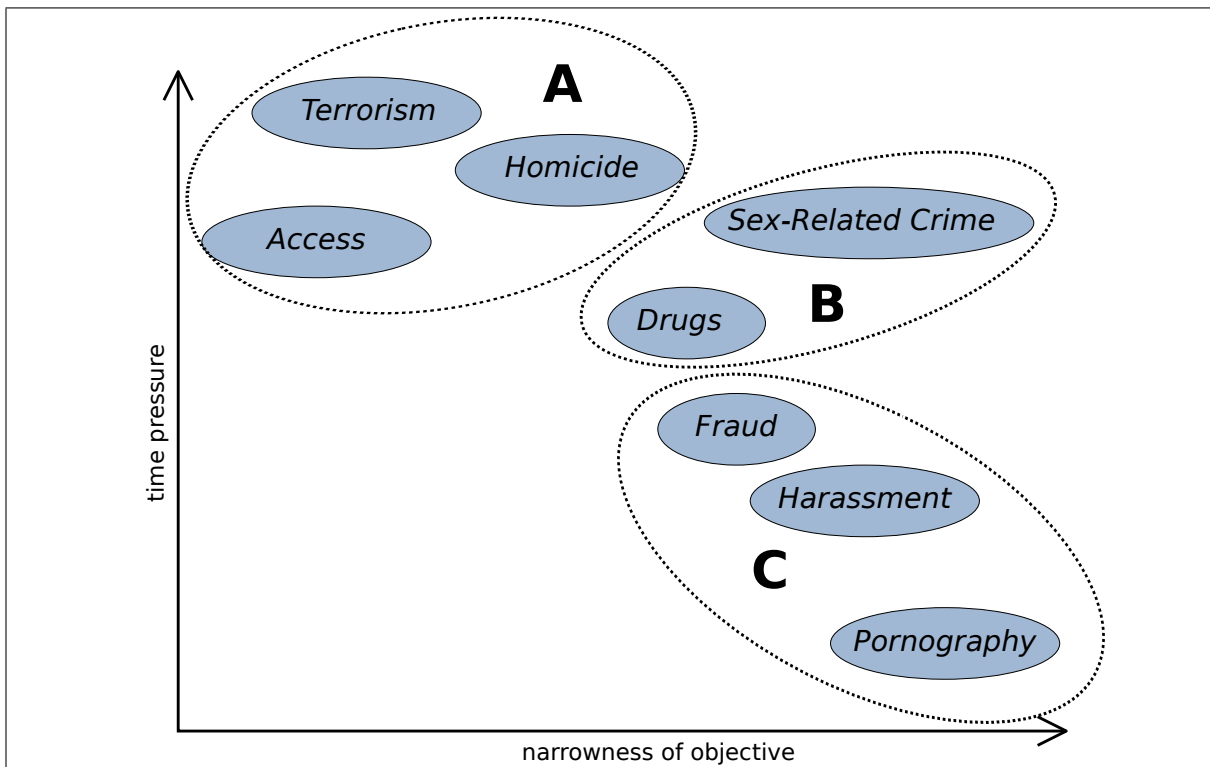


Figure 3.3: Time pressure versus narrowness of objective

Group B is the group that combines relatively high time pressure with a narrow objective. Sex-related crimes (e.g. rape or child abuse) have to be cleared up fast because the victims can be in danger until the suspect is arrested. The focus here is mainly on communication and photo or video material, so it is very narrow. In drug crime, communications are the main interest of investigations. The time pressure is lower than with sex-related crime, but still high enough to benefit from a faster investigation. However, it is important to keep in mind that the most common objects that are analyzed in drug cases are mobile phones [60]. Digital forensics is used in this case to discover contacts and communication, identifying the location and plans of the people involved.

Since the memory on mobile devices is relatively limited, the time-savings that can be achieved are not very high. Mobile phones put aside, this is the group that can benefit greatly from selective imaging, as it has the best combination of potential and demand for time improvements in the investigation.

Group C are crimes where time pressure is not a big concern but investigators have a good understanding of the data that contains potential evidence. For instance, fraud investigations mostly target storage devices like desktop computers and flash-drives [60]. The main targets of the investigation are communication (e.g. e-mail) and office documents. This type of data can be easily identified and selected for acquisition. Harassment also mostly reveals itself in traces of communication so the objective is very narrow. Since in most fraud or harassment cases no greater danger for the public exists, the time pressure is not very high compared to crimes like terrorism. However, the shortened investigation process will decrease the costs significantly. Pornography mainly involves audio and video material. If not only the possession but also the trade of illegal pornography is investigated, also communication will be of interest. However, it is a relatively narrow set of data and investigators know very early in the process what to look for. For these cases, selection will be relatively easy and the reduction of data will be significant. Many cases merely center on the question if a certain person was in possession of illegal pornography. The time pressure in these cases is very low as nobody will get harmed if the investigation takes a long time. These characteristics indicate that although there is no pressure to accelerate the acquisition procedure in this case, the benefits in terms of cost will be substantial.

In Figure 3.3, the different categories of crime are arranged in the two compared dimensions. While the positions in the diagram do not represent exact values, which also heavily depend on the individual case, they still indicate the tendencies of the average case of the categories.

To estimate the overall fraction of investigations where selective imaging can be applied, we put the assessment of these groups of crime into context with the amount of exhibits that are analyzed each year. Turnbull et al. analyzed the work of the South Australian Police Digital Forensics Division and published statistics on the number of exhibits in each category of crime [60]. The data represented in Table 3.1 accounts for about 75 percent of all exhibits analyzed in the fiscal year 2007/2008. The remaining 25 percent are smaller or unlisted categories, that we left out for simplicity. To compare the categories used in this thesis, the percentages are recalculated to represent values relative to the overall number of exhibits in the table.

Table 3.2 aggregates these values to the groups defined in Figure 3.3. Group B represents the cases that are best suited for selective imaging as they both allow for large reductions of data and have a need for faster results. Of 309 Exhibits analyzed, 159 belong to this group. This means that 51.46 percent of the exhibits are very well suited for selective imaging techniques. Group C does not have a strong necessity for time improvements. Nevertheless, they possess a significant potential for time and cost reductions. Those crimes amount for 42.39 percent of the cases. In conclusion this means that 93.85 percent of all exhibits are well suited for selective imaging and only 6.15 percent are problematic and require hybrid or adapted techniques to profit from selective

### 3.1 Selective Acquisition Procedures

---

Table 3.1: Exhibits per Crime in Australia 2007/2008 [60]

Crime	Exhibits	Percentage Absolute	Percentage Relative
Terrorism	1	0.24%	0.32%
Homicide	13	3.18%	4.21%
Access	5	1.22%	1.62%
Sex-related	46	11.25%	14.89%
Drugs	113	27.63%	36.57%
Fraud	31	7.58%	10.03%
Harassment	29	7.09%	9.39%
Pornography	71	17.36%	22.98%
<b>Total</b>	<b>309</b>	<b>75.55%</b>	<b>100.00%</b>

imaging.

Table 3.2: Exhibits per Group

Group	Exhibits	Absolute Percentage	Relative Percentage
A	19	4.65%	6.15%
B	159	38.88%	51.46%
C	131	32.03%	42.39%

These numbers only account for 75 percent of all cases, as an exhaustive analysis of all possible crime scenarios is out of the scope of this thesis. However, even if we assume that the other 25 percent are not suited for selective imaging, at least 70.90 percent of all exhibits are suited for selective imaging and will gain significant benefits with this technique.

Another possible benefit concerns data protection. In some cases it can be impossible to create a sector-wise image of a device, because it contains data that is not covered by the search warrant. For example large multiuser environments like corporate email servers store data of many different users. If some of the users have nothing to do with the case, it can be illegal to copy their data. The details on legal constraints in these cases are elaborated in Section 3.2.3. Selective imaging in these cases is a suitable alternative to circumvent legal problems by only acquiring data that is inside the investigations boundaries.

When dealing with damaged or old devices, selective imaging can also provide an advantage over common sector-wise images. These devices can often handle only a limited amount of read operations before they break. When creating a sector-wise image, data is copied linearly, starting at the beginning of the device. If the device fails during

this procedure, only a part of the disk can be acquired. Examiners have no control over this process, if important evidence is located on a part of the drive that is located behind the point of failure it is lost. Selective imaging allows examiners to prioritize the acquisition order. The parsing of metadata such as the filesystem does not require large data transfers, after which examiners are able to specifically copy those data objects first that have a high probability to be of value to them. If the device fails in the middle of the acquisition process, the amount of copied data will be the same but the value evidence-wise will be much higher.

Despite of these benefits, examiners have to keep in mind that selection is sometimes an irreversible process. If the acquisition is performed on systems that are afterwards returned to their respective owners or made accessible to their users, any data that is overlooked in the selection process can often not be acquired at a later point in time. In such situations the selection should be as broad as possible to mitigate the risk of losing evidence. Cases where the searched type of data is unclear should be treated like Group A, even when there is no time pressure.

## 3.2 Partial Images

Conventional storage methods are suited for sector-wise images of exactly one device. The identical copy of the device allows for easy verification and the sector-wise approach assures completeness. The nature of selective imaging makes it very difficult to use established storage methods. The acquisition process does not produce one continuous stream of data, but many small data objects. Also, these objects carry a lot of metadata from lower levels of abstraction that has to be stored and associated with them. Finally, the provenance information for these objects needs to be extracted and stored with them. These characteristics require a new form of storage container, which will further be referred to as *partial image*.

### 3.2.1 Definition

One reason why forensic examiners usually acquire hard disks at device level is metadata. Each level of abstraction has its own set of metadata and if the acquisition happens on a higher level, this metadata is lost. For example, acquiring only specific files on a disk will discard any information that exists in filesystem data structures. A partial image must therefore contain any metadata from levels that are below the level of selection.

The pre-acquisition steps discussed in Section 3.1.1 also produce information on the selected objects. For instance, examiners might sort files by their characteristic header and footer. This technique involves reading the first and last few bytes of a file, that often are characteristic to its type. Examiners might also calculate cryptographic hashes of files, to identify those that store known content. Several databases with hashes of known files exist, police investigators for example often compare hashes of pictures to a database with known values for illegal pornography. The results of these harvesting steps are lost if not stored together with the data. Partial images must therefore store

### 3.2 Partial Images

any results of the preliminary steps performed before acquisition.

The verification of provenance and integrity is relatively straightforward with sector-wise images. A simple comparison of hash values between the original device and the image is sufficient. When applied to partial images, this approach does not work because there are additional factors to consider. For hash based comparisons the exact location of the data object on the disk must be known. Additional information such as the path of a file is also necessary to prove the location on filesystem level. The requirements for provenance verification are discussed in length in Section 3.2.2. A partial image needs to store all information necessary to be verifiable against the original at all times. Also, the partial image shall not be limited to contain objects from one specific level of abstraction. For instance, the concept of acquiring only files is very inflexible. During pre-analysis, parts of a file might be reconstructed from slack. Furthermore, metadata regarding this file will often be found in some data structure of the file-system. The combination of both file fragment and metadata is not clearly separable into layers. However, it might be exactly what is needed as evidence. A partial image can thus not be fixed onto a layer, but must be flexible enough to contain any aggregation of data from the disk, which is of relevance to the investigation.

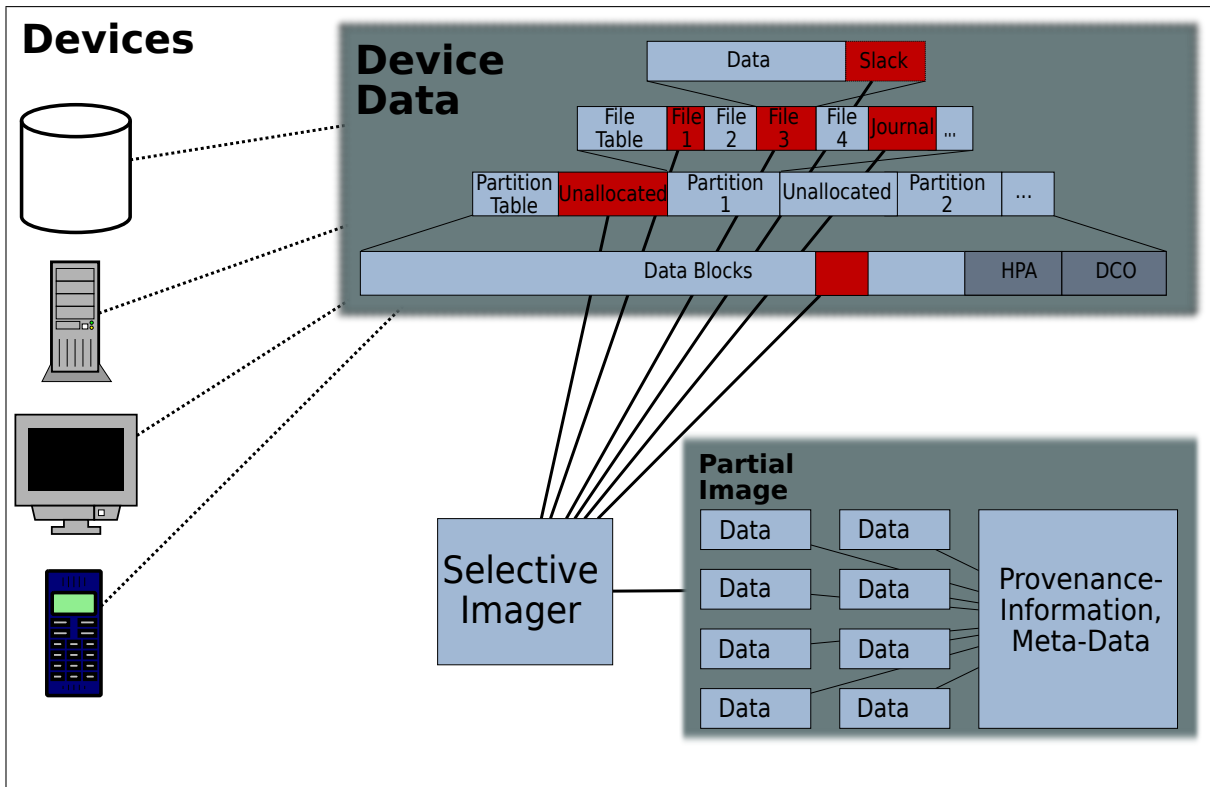


Figure 3.4: Partial Image

Taking these requirements into account, a partial image can be defined as *an aggregation of data objects from a digital device, together with all relevant metadata, that can be verified against the original at all times*. Figure 3.4 illustrates this container and its

data sources on the base of the granularity model, introduced in Section 3.1.2. Any kind of device can be the data source, also multiple devices can be aggregated. The examiner selects an arbitrary amount of data objects from the device, regardless of the abstraction layer. The selective imager then extracts these objects, their metadata, the results of preliminary analysis procedures and the complete provenance information. This data is then stored in a container that is referred to as *partial image*.

### 3.2.2 Provenance Assurance

The provenance of a sector-wise image can be easily proven by hashing it and comparing the hash with a hash taken from the original device. Additional metadata, like the devices serial number, can be collected during acquisition and be compared during verification of the image. This information can be easily stored as written and signed note by the examiner. The provenance of a partial image on the other hand is not as easy to verify, as it contains only parts of the data on the original device and thus the hash of the partial image will not match the hash of the device. Partial images thus need a dedicated mechanism to assure the provenance of the data objects they store.

Turner describes five attributes of digital provenance documentation, which are required for it to be reliable [61]:

- Uniqueness
- Unambiguity
- Conciseness
- Repeatability
- Comprehensibility

Provenance information must be unique, as it should identify a single instance of a data object. For files, this implies that a cryptographic hash is not precise enough, as it is identical for any copy of this file as well.

Also, provenance information must be unambiguous in a sense that it can not be interpreted in any other way. A block address for example implies that the interpreter knows the block-size of the device. If this information is omitted, the block address can be interpreted in many ways and thus is ambiguous.

Conciseness of provenance information is also important, as it might have to be understood by third parties, like an expert witness under oath.

Furthermore, the acquisition of provenance information must be repeatable, to allow for verification. If a third party is commissioned to verify the provenance of digital evidence, it must be able to easily replicate this information. This must be possible independent of the forensic tools used, as not every examiner uses the same tools and it is a common procedure in digital forensics to verify results of one tool by the use of another.

Finally, provenance information must be comprehensible, as it might have to be explained to a non-technical audience in court. These people are often unfamiliar with technical terms like block-address or cryptographic hash and will thus prefer more comprehensible explanations such as filenames or a path in the filesystem.

For provenance documentation of arbitrary data objects on a block-device, multiple metrics [61] exist that have to be considered:

- The block-address on the device
- The cluster-location in the filesystem (if the object resides in a filesystem)
- The path in the filesystem (if the object is a file)

None of these metrics comply with all of the required attributes. As illustrated in Table 3.3, each metric has its specific strengths, but fails to measure up to all of them. The block-address and cluster-address lack comprehensibility, cluster-addresses in addition requires additional information on the filesystem and thus is not very concise. The path on the other hand is comprehensible even for non-technical people, but lacks conciseness as it also requires supporting data from the filesystem. To achieve reliable documentation of provenance with a path, not only the path must be documented but also the partition and filesystems structure.

Table 3.3: Characteristics of Provenance Metrics

Attribute	Block-Address	Cluster-Address	Path
Unique	•	•	•
Unambiguous	•	•	•
Concise	•		
Repeatable	•	•	•
Comprehensible			•

As proposed by Turner, reliable provenance documentation requires the combination of multiple proveniential metrics also referred to as proveniential keys [63]. For simplicity, the block-address can be used when communicating digital evidence provenance to technical people. For communication with non-technical people, the path is sufficient as comprehensibility trumps conciseness in this context. When a partial image is acquired, at least two of these metrics should be documented, to be able to provide reliable provenance information to any audience.

To verify the partial image, in addition to the proveniential keys, a cryptographic hash is necessary. A hash does not qualify as a metric of provenance because it can not distinguish an original data object from a copy. However, it can uniquely identify the content of the object. The hash therefore serves as verification metric for the content of data objects.

### 3.2.3 Legal Considerations

Evidence acquisition procedures stand and fall with their acceptance in court. In this section, we will evaluate the way digital evidence is handled in court and assess the most common arguments against selective evidence acquisition approaches.

Legal procedures and laws differ from country to country and it is beyond the scope of this thesis to evaluate the legal implications of selective imaging in the entire world. The conclusions drawn in this section are based on German law and the code of criminal procedure in Germany unless otherwise stated. However, the basic principles are similar in most countries and thus most arguments apply to other countries as well.

#### Legal Acceptance of Digital Evidence

From the judicial perspective, the digital era is very young and thus has not yet been incorporated seamlessly into the legal system. For example, there is no well defined way to bring data into a legal action as evidence. The only possible ways in Germany at the moment are either as a document or as something called »Augenscheinsbeweis« (evidence of appearance). The tribunal is free to adjudge the value of such evidence, which is mostly brought forward in form of printouts or expert testimony [31].

When acquiring evidence in the scope of a search, §94 of the German code of criminal procedure (StPO) allows for the seizure of objects, that can be used as evidence [42]. The understanding of *object* in the context of digital evidence thus defines what can be seized by investigators. Bär defines *object* as any information system or device that can store data. Data can also be transferred as an immaterial object to another storage device, which would make the device it is stored on an *object* that can be seized [12]. This implies that an image, be it sector-wise or partial, can be regarded as object and thus is seizable, as long as it's stored on a physical device.

#### Judicial Appraisal of Partial Images

§94 subsection 1 StPO stipulates, that evidence has to be taken into custody or be *secured in another way*. This explicitly allows for arbitrary methods of acquisition, if the device, the digital evidence resides on, can not be physically acquired. Bäcker et al. state, that if the goal of the investigation is only to verify the existence of specific files, a selective acquisition approach on file level is sufficient if the provenance of the files is properly documented [11]. Even more, §110 subsection 3 StPO explicitly allows examiners to selectively secure files from remote systems, if they are relevant for the investigation but the systems are not directly accessible to perform a sector-wise image. This suggests that the selective acquisition of specific files does not violate any judicial requirements. However, this approach at the moment is limited to cases where standard acquisition procedures are impossible.

These facts show, that currently no legal framework exists for the selective acquisition of data in cases where sector-wise images are possible. Nevertheless, the legal framework for digital evidence is very sparse in general and the way digital evidence is brought into

legal action is more reliant on the person, that presents it than on the method it was acquired [11].

### Reliability of Partial Images

Digital evidence is often challenged in court, even when acquired with standard sector-wise imaging techniques. The most common arguments are:

- The incriminating data was planted by the examiners, it was not on the device before the image was taken but deliberately put there after acquisition.
- The data was altered during analysis due to improper handling.

To refute such allegations it is important to have a proper documentation of the acquisition and analysis procedure, that proves the data was always handled according to forensic principles. Opposers of selective imaging approaches often argue, that a partial image can not provide adequate provenance documentation and thus is vulnerable to challenge in court. However, as long as partial images employ the mechanisms for provenance documentation described in Section 3.2.2, they are equally reliable as conventional sector-wise images.

Even advocates of selective approaches believe in strong boundaries for file level selection. For example, Bäckér et al. state that sector-wise approaches are necessary in cases where data is expected to be concealed or deleted [11]. This is because the current perception of selective imaging is limited to a specific level of abstraction. However, this is not strictly necessary. Section 3.1.2 demonstrates that selection does not have to be limited to a fixed level of abstraction. It is easily possible to selectively acquire the slack of a specific filesystem, a swap-partition or even the entire unallocated space between partitions. As long as the narrowness of the selection is chosen correctly according to the characteristics of the case, there is no problem. Nevertheless, selective imaging places great responsibility in the examiner that performs the acquisition. Sector-wise images can be acquired forensically sound with minimal training. Partial images require a good understanding of forensic techniques and inherit high risks if the acquisition is not performed correctly.

Another common argument against partial images is, that they can not offer the same coverage as sector-wise image. The principle of selection inherits the omission of some data on the device and opposers argue that there are no guarantees the omitted data might become relevant later in the investigation. While this is undeniably true, all other fields of forensics have abandoned this claim a long time ago. When acquiring physical evidence, investigators will not ban the public from the crime scene until the end of the investigation, just to be able to come back any time and search for additional evidence. This is referred to as principle of reasonableness and is equally valid with digital evidence [39]. Of course, perfect coverage is a nice feature, but in reality costs are a factor that has to be taken into account.

A serious argument is, that images that have been acquired by selective approaches might lack exculpatory evidence. §160 subsection 3 StPO stipulates that investigators

not only have to acquire incriminating evidence, but also evidence that can exculpate the defendant. Opposers of the selective approach argue, that exculpatory evidence could be missed in the selection and thus the partial image violates the defendants due process rights. However, Bäcker et al. state that examiners are not obliged to dimension their search as broadly as possible in any case [11]. The regulation only applies if there is an indication that such data might exist. Of course selective imaging also allows to acquire this kind of evidence. While the argument does have an element of truth, it actually does not attack partial images but simply highlights the responsibilities of the examiners.

### Legal Benefits from Partial Images

Apart from the controversial issues, Kenneally and Brown argue that partial images provide solutions for some problems that can not be solved with sector-wise approaches [38].

An issue that becomes increasingly important is data protection. In the United States this is addressed by the 4th Amendment. Search warrants can be challenged if their scope overreaches the standards for narrowness, particularity and reasonableness [38]. In Germany this is covered by the principle of commensurability. This principle stipulates that evidence has to be acquired in a reasonable way and that the basic rights of the suspect must be respected. If possible, data should be acquired partially to prevent unnecessary strain on the concerned party [12]. The only common method to achieve this at the moment is the duplication of files with copying tools, which violates the forensic principles that stipulate provenance documentation and integrity assurance. For this reason this technique is only used as a measure of last resort. Selective imaging enables investigators to address data protection issues in many cases where the only other option would be a sector-wise image and thus can prevent 4th Amendment challenges.

Some cases involve evidence on systems that contain data from multiple parties. In Germany, §97 StPO prohibits the confiscation of written correspondence between the defendant and persons who have the privilege to refuse to give evidence. This regulation also covers digital correspondence and is especially relevant in a business context where tax consultants or attorneys might be involved. The German Federal Constitutional Court explicitly emphasized the principle of commensurability in these cases [12, mn. 425]. While it is acknowledged that in many cases a sector-wise image is the only possible solution, selective imaging has the potential to ease these situations considerably for both parties.

Finally, Kenneally and Brown argue that time and cost factors can lead to problems when sector-wise images are the only means of digital evidence acquisition [39]. When time and cost constraints limit the amount of devices that can be imaged, investigators risk the omission of important evidence from devices that are ignored. In cases involving multiple digital devices, sector-wise imaging is basically a form of selective imaging, carried out on device level. This level however is much too coarsely grained to allow for an efficient allocation of limited resources. When time or cost are a limiting factor, it is much better to risk overlooking single files and cover all devices than it is to cover some

devices 100%, but miss entire devices.

In conclusion, selective imaging is a technique that is well within the code of criminal procedure in Germany and most other countries. If applied correctly, it is equally reliable as sector-wise imaging but has the potential to overcome some of the problems, such as data protection issues or limited resources, that have recently emerged.

## 3.3 Summary

This chapter provided the principles and storage methods that are necessary to perform selective imaging. Based on the *investigative process* by Casey, we introduced a process model for selective imaging. Because examiners require information on the contents of digital devices ahead of the actual acquisition process, we postulated the need for recovery, harvesting and reduction performed on the device itself through a write-blocker prior to the preservation step. The acquisition will then be carried out on the reduced data amount and the investigative process can be resumed on the image. If the pre-acquisition steps were not narrow enough due to the complexity of the case, the investigation can repeat those steps in the lab in greater detail.

We then discussed the matter of granularity, regarding the elemental data unit of acquisition. Data on digital storage is organized within several nested layers of abstraction. The most basic unit of data is a file. Files are organized in filesystems, which again are embedded in partitions. Partitions reside on top of the device level, where data is just an unorganized stream of bits. If restricted to a specific level, selection becomes inflexible because fragments of evidence can reside outside the organizational units of these structures. We concluded, that selection should be possible on any level, ignoring logical structure, to be able to select any arbitrary fragment of evidence. When doing so, it is important to explicitly acquire any metadata that is stored in the organizational data structures of the different levels, as it would otherwise be lost.

Aside from the technical details, we analyzed the applicability of the concept to different categories of crime, and estimated the benefits for each. While we determined that selective imaging can be employed in virtually any type of investigation, a small group of crimes exist where it is advisable to also perform a complete sector-wise image. This is to account for cases where data has been concealed and investigators can not take the risk to overlook anything. A hybrid model for selective imaging was defined, that enables examiners to benefit from selective imaging even in these cases. The model stipulates the creation of a sector-wise images in parallel to the examination of the image that was acquired selectively. When the potential of the selectively acquired image has been fully utilized, examiners can migrate to the complete image.

We also determined that the benefit from selective imaging is greatly dependent on the narrowness of the objective. We estimate the number of cases that benefit highly from selective imaging to about 51 percent, while the technique is expected to be applicable to a fraction of 70 to 93 percent of all crimes.

As a storage container for selective imaging, we defined the term partial image as *an aggregation of data objects from a digital device, together with all relevant metadata*,

*that can be verified against the original at all times.* We evaluated the requirements to reliably document the provenance of partial images and concluded that a combination of multiple metrics, such as block-address or path of a data object, is adequate.

The legal acceptance of selective imaging was determined to be equal to the common approach of sector-wise imaging, when implemented according to the guidelines for provenance and verifiability we postulated. While opposers of selective imaging claim the opposite, we discussed and invalidated the most common arguments. There are even legal benefits when employing a selective approach, as data protection issues or cost factors can sometimes invalidate evidence that is acquired with the complete, sector-wise method.

The methods and principles introduced in this chapter serve as the foundation for software that can create partial images according to the modified investigative process. The guidelines developed for provenance and verifiability assure that the images the software creates are legally reliable. The next chapter documents the implementation of a prototype that follows these principles.

## 4 Implementation

This chapter focuses on the implementation of a prototypical selective imager. Section 4.1 explains the reasons for the decision to base the selective imager on a specific framework and storage format. The different frameworks and formats introduced in Chapter 2 are compared and the tools best suited for selective imaging are chosen. In Section 4.2, the architecture of both the framework and the storage format is illustrated. Based on this structure, the design choices for the selective imager are explained and its components are described. Section 4.3 illustrates the inner working of the most important parts of the acquisition module, the AFF4 connector and the partial image verifier. Finally, Section 4.5 explains the usage concept of the selective imager. The application of the selective imager in the Digital Forensic Framework is shown and methods for the verification of the created images are discussed.

### 4.1 Selection of Technical Foundation

One of the most important design goals for the implementation is the integration of the selective imaging process into existing forensic tools. The procedures for recovery, harvesting and reduction in the pre-acquisition phase are similar to the ones usually employed after acquisition. To simplify the transition between the device and the image, that takes place after the image has been acquired, it should be possible to use the same tools for pre-acquisition analysis as for post-acquisition analysis. Also, it would be rather pointless to reinvent the wheel for such well researched and often implemented procedures.

Another important goal regards the storage format. If possible, the selective imager should use existing forensic storage formats for the created images. Even if small modifications are necessary, this enables examiners to use their favorite choice of tools for the post-acquisition phase. Also, this allows examiners to verify their findings with multiple tools, which is a common procedure in digital forensics to assert the correct function of the analysis software. If evidence is challenged in court, it will also simplify the work of an expert witness under oath, who might be unfamiliar with the tool that was used in a specific case.

This section analyzes the suitability of the different forensic frameworks and formats for the purpose of selective imaging and determine the best foundation to base the implementation on.

### 4.1.1 Analysis Framework

In Section 2.2.2, we presented an overview on current digital forensic frameworks. Because the reference implementation should be accessible by everyone without restrictions, only non-proprietary solutions will be considered. This limits the selection to Autopsy, PyFLAG and the Digital Forensic Framework. There are a number of essential features, a framework must support to be suitable for selective imaging. Since a design goal of the selective imager is to integrate into the framework, it has to have some kind of plug-in system. It also needs to be able to operate on live storage devices, meaning it needs a mechanism to provide read only access to connected devices and apply its full functionality on them.

To perform the pre-acquisition analysis, the framework has to support the most commonly used techniques for the steps recovery, harvesting and reduction. Frequently used techniques for the recovery step include the recovery of deleted files as well as carving techniques to recover data that no longer is inside a valid filesystem. The harvesting step usually includes the parsing of metadata from the filesystem and the analysis of file-magic, which is the detection of filetypes by identifying know signatures of the contents. On windows systems, the registry is a very potent source of metadata and thus needs to be parsed during harvesting. The reduction step needs some method of categorization, normally this is done with searches and timeline analysis. The categorized results need to be marked in some way, this is usually accomplished with some sort of bookmarking functionality. Also an indexing engine can be very useful, as it accelerates searches significantly.

These frameworks are all available for the Linux operating system. Some have been ported to run on MacOS or Windows, but the initial development started on Linux. To avoid compatibility issues, we decided not to evaluate the choice of operating system and compare the Linux version of the tools, as the selective imager will also be developed on a Linux system.

Table 4.1: Features of open-source digital forensic frameworks

Feature	Autopsy	PyFLAG	DFE
Plug-in System		•	•
Access Live Devices	•	•	•
File-Carving		•	•
Deleted-File Recovery	•	•	•
Filesystem Metadata Parsing	•	•	•
Analysis of File-Magic		•	•
Registry Parsing		•	•
Time-Line Analysis	•	•	•
Search	•	•	•
Bookmarking			•
Indexing		•	

In Table 4.1, the three frameworks are compared in regard to their implementation of these features. As the table illustrates, PyFLAG and the Digital Forensic Framework are on par in regard to the relevant features. Both Frameworks miss one feature for the reduction phase. However, neither the missing bookmarking functionality in PyFLAG nor the non-existent indexing engine in DFF disqualify either one of the frameworks for selective imaging. Autopsy on the other hand, due to the lack of a plug-in system and several other gaps, is not well suited as a platform for a selective imager.

To decide between PyFLAG and DFF, the maturity of the projects also needs to be considered. PyFLAG was publicly released in 2008 [19], while DFF is a rather young project, which is available since the end of 2009. While this indicates the PyFLAG project is more stable, the current maintenance and development efforts also determine the usefulness of the framework. The most recent version of PyFLAG is Version 0.87pre1 (released 3rd Sep. 2008). DFF on the other hand is actively developed by the french forensics company Arxsys [3] and new versions are released every two or three months.

Taking all these considerations into account, we chose the Digital Forensic Framework as the platform for our selective imager, mainly due to its active development and extensive feature set.

### 4.1.2 Storage Format

Over time, a number of features for image storage formats have been developed that can be useful to forensic examiners. The most important one is compression. Because image files require random access during analysis, it is very inefficient to use conventional compression software to reduce their storage size. Before a single byte can be read, the entire Image has to be decompressed. Modern image formats compress images block-wise, so only small blocks have to be decompressed when reading arbitrary data blocks in the image. This is called seekable compression, because it does not require noticable effort to seek to an arbitrary position in the image. Because compression reduces the size of images on disk, storage costs are significantly reduced.

Encryption is also a very practical feature, because it prevents unauthorized access to the image. Computers often contain large amounts of personal or confidential data. The leakage of sensitive information can have severe consequences. If storage devices containing forensic images are lost or stolen, the owner of the data they contained might sue the examiners for the leakage of proprietary or private data. If the images are encrypted properly, the possession of the image alone does not grant access to the contained data and can prevent this problem to a certain degree.

Furthermore, several formats have incorporated the storage of metadata. Some only allow to store a set of predefined data like a serial number or the name of the examiner. Others allow for any arbitrary value to be stored within the image, which can be used to extend the format with new functionality. This feature is very important for partial images, as it is needed to store the provenance documentation discussed in Section 3.2.2. Cryptographic signing can strengthen this even further, as it can be used to guarantee for the authenticity of the stored information.

Finally, some formats allow to incorporate multiple data objects within a single image.

Table 4.2: Comparison of image formats

Feature	RAW/DD	SGZIP	E01	AFF3	AFF4
compression		•	•	•	•
encryption		•		•	•
basic metadata			•	•	•
arbitrary metadata				•	•
cryptographic signing				•	•
multiple objects					•

This greatly reduces storage and analysis complexity of big cases with multiple exhibits, as there is only one file containing the entire digital evidence of the case.

Table 4.2 compares the image formats introduced in Section 2.2.1 in respect to these features. To achieve a maximum of interoperability, the storage format for partial images has to be an open standard. Due to this restriction, the EWF-01 format will not be considered. That leaves exactly four of the formats introduced in Section 2.2.1:

- RAW/DD
- SGZIP
- AFF
- AFF4

Because the RAW and SGZIP formats do not support metadata integration, the provenance documentation would have to be stored in a separate file. This would severely limit interoperability, because each tool supporting raw images would still have to create a parser for this metadata file to be compatible. The AFF3 format does allow for metadata, but supports only one data object per file. The selective imager therefore would have to create one file for each data object. While this is not impossible, it greatly complicates the organization of partial images, as they consist of a large number of different data objects. AFF4 does not have this limitation, as it allows for an unlimited amount of arbitrary data objects per image. It also supports unlimited arbitrary metadata storage, which allows the selective imager to store all information directly within the image. Considering these criteria, AFF4 seems to be the best format to store partial images, as no modifications to the format are necessary. The selective imager can simply store each data object inside the container and attach the provenance documentation and other information using the existing metadata storage mechanism.

## 4.2 Architecture

In this section, the architecture of the selective imager is presented on a component level. We also give a short overview on the Digital Forensic Framework and the AFF4 Library, to illustrate the implementation choices made in the selective imager.

### 4.2.1 Framework

The Digital Forensic Framework is structured in a modular way. It consists of a graphical user interface (GUI), a virtual filesystem (VFS) and several modules, that provide the actual forensic functionality.

The core component is the Virtual File System, which is a generalized abstraction of a filesystem, designed to hold many kinds of data streams. It has a tree-like structure of nodes, that can be explored in the GUI very similar to a common filesystem like NTFS or EXT4. Nodes can be directories, files, links to other nodes or any other object that contains a stream of data. For example, the entire file-slack of a filesystem can be defined as a mapping of the unused bytes at the end of the last cluster of every file. This logical construct can also be a node, which illustrates the difference between common filesystems and the VFS. Another major advantage is that, given the proper parser, the VFS can contain multiple different filesystems simultaneously and make them available to other programs in a standardized way.

The actual functionality in DFF is achieved by plug-ins called modules. Modules are pluggable objects that interact with the nodes in the VFS to perform forensic operations. They can be used to extract information, re-arrange data or even create new nodes. For example, the VFS Translation Driver that provides read access to forensic images is also a module. If instructed to load an image from the local hard-disk, the driver will open the image-file, connect a parser for the specific format and finally create a new node in the VFS representing this image. Other modules can then easily access the image through the standard interfaces of the node. Almost all functions of DFF are realized as a module. For example even the picture viewer is a module and reads pictures through the node interface. Modules can also produce results that can not be stored in form of a new node in the VFS. For example, a filesystem parser will not only create nodes for the files in the filesystem, but also retrieve a lot of information like mac-times or file-size. As this information is always bound to a specific node, the module can associate it with the node through an attribute, which basically are key-value pairs that hold information on a node.

To illustrate the process flow in DFF, Figure 4.1 depicts a simplified sample configuration during the analysis of a hard-disk. First, the VFS Translation Driver connects to a storage device or image and creates a node that provides access to its raw data. The partition parser module is then invoked, reading the partition table and creating nodes that represent the partitions that exist on the device. The partition contains a FAT filesystem, thus a FAT parsing module is invoked on the partition. The module will analyze the filesystem and create nodes representing the directory structure. If implemented in the module, also more abstract constructs like a node representing file-slack can be created. Files again can contain data-structures that need to be parsed to analyze them. In this example, a zip-archive containing a compressed file is stored in the filesystem. A decompression module is thus applied to the archive, creating a node that represents the contained file. Accessing this file will then invoke functions that decompress the file from the archive.

This is a very limited example to demonstrate the architecture of the framework.

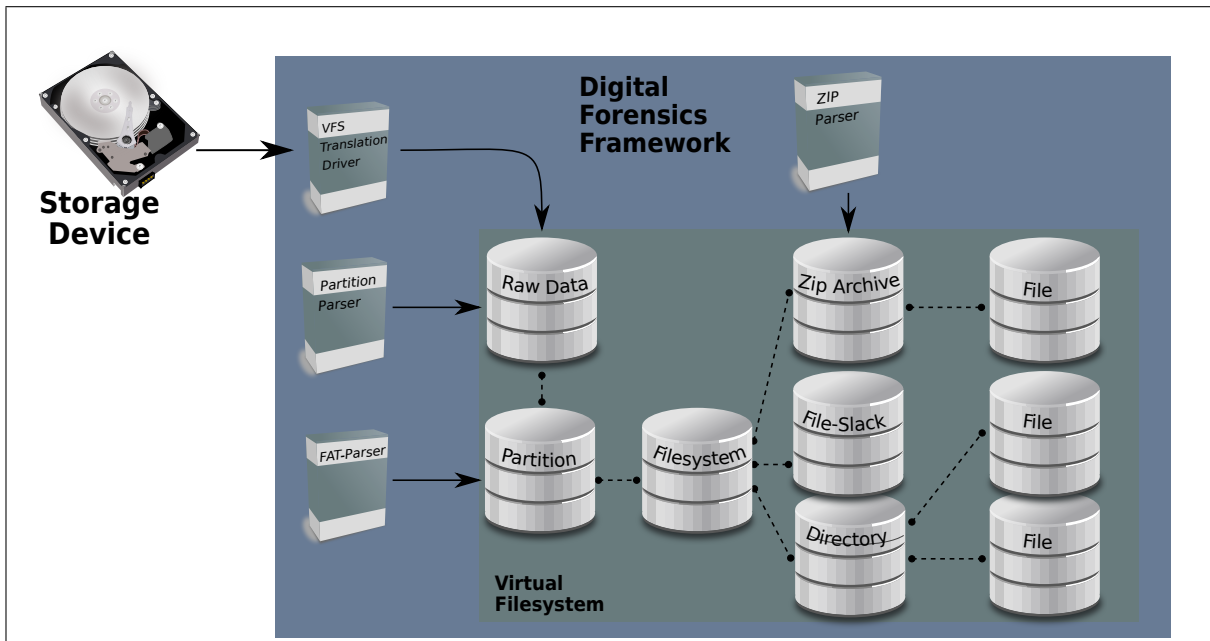


Figure 4.1: Architecture of the Digital Forensics Framework

Building on these principles, more advanced functionality can be implemented. This includes modules for the reconstruction of RAID arrays, which are arrays of independent disks where data is spread in a pattern over multiple devices. Any arbitrary functionality that operates on node data can be realized, for example a module that parses e-mail archives and creates text nodes for every e-mail is also possible.

### 4.2.2 Format

As introduced in Section 2.2.1, an AFF4 volume consists of one or multiple streams, as well as an aggregation of RDF-Facts [68], describing metadata of these streams. Objects in an AFF4 volume are identified by a Uniform Resource Name (URN) [37], that is uniquely generated for each object. RDF-Facts describe attributes of these URNs.

The creator of AFF4 provides a standard library to access and create AFF4 volumes. The central unit of the library is the resolver, an object that can resolve and access data objects in AFF4 volumes by their URN. Also, the resolver can read and write RDF-Facts and make streams accessible for external programs.

Figure 4.2 illustrates the way an application interacts with the resolver to access AFF4 volumes. In this example, the volume only contains image-streams. For the sake of clarity other streams are omitted. After the application has created a resolver for a specific volume, it can start querying for information. For example the application can instruct the resolver to retrieve an iterator, containing all objects in the volume. The application can then retrieve and store metadata on these objects, by calling the respective functions of the resolver. The resolver will serve these requests by accessing the information storage and parsing its contents. If the application needs to read from or write to a stream, it will query the resolver with the respective URN. The resolver will

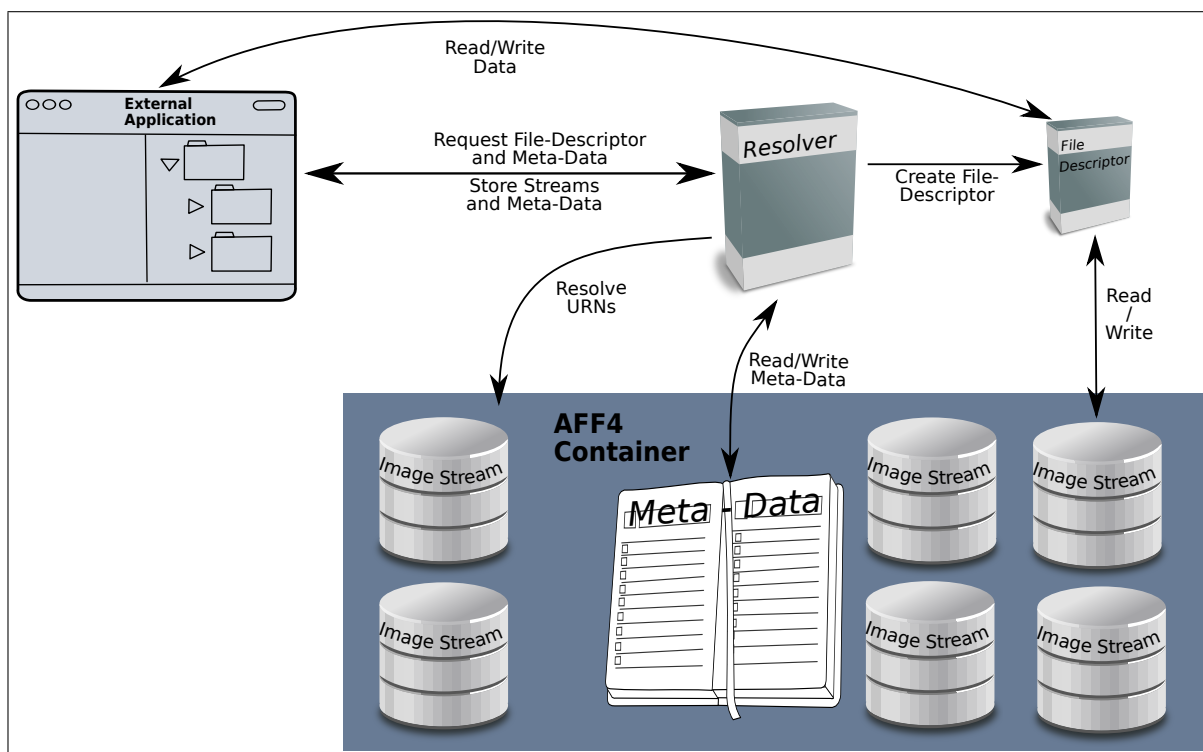


Figure 4.2: Architecture of the AFF4 Library and Format

then look up the location of the stream and create a file-descriptor object, that provides access to the stream through standardized `read()`, `seek()` and `write()` functions. The application can then access the stream as if it was a file.

The library contains several more advanced features, like map-streams, encryption or cryptographic signing. However, for the creation of a selective imager, these are the necessary functions. All other functionality of AFF4 can be utilized with the standard software that comes with the library.

### 4.2.3 Components

The selective imager is based on both, the DFF framework and the AFF4 library. The functionality is provided as a plug-able module for the DFF framework. The implementation itself consists of three components:

- The Selective Imager
- The AFF4 Connector
- The Verifier

The selective imager module is used by examiners to create a partial image and write the data objects with their respective metadata to it. This process is illustrated in Figure 4.3. When invoked on a group of nodes, it will first query the VFS for any

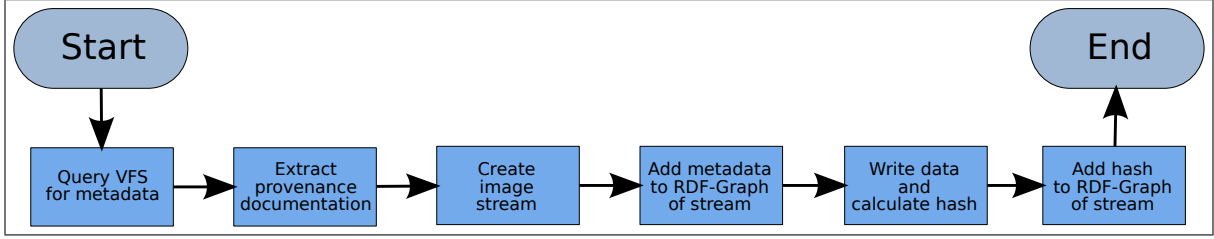


Figure 4.3: Acquisition Procedure of the Selective Imager

metadata regarding the selected nodes. It will extract provenance information and the results of any prior work on the nodes, that is stored in their attributes. When the metadata extraction is complete, the selective imager will create an image stream for each node and associate a RDF-Graph of all metadata with it. Finally, it will copy all data from the nodes to their respective image stream, calculating a cryptographic hash in the process. These hashes are then added to the RDF-Graph of the streams, as a verification metric.

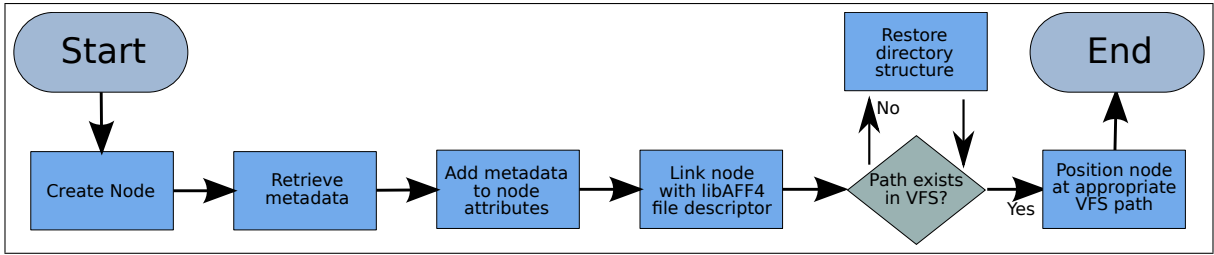


Figure 4.4: Import Procedure of the AFF4 Connector

The AFF4 Connector module is used to import partial images into DFFs VFS. When invoked on a partial image, the connector will load it into a resolver and create the necessary objects in the VFS. This process is illustrated in Figure 4.4. For each image stream in the image a new node will be created in the VFS. The metadata in the corresponding RDF-Graph will then be retrieved and associated with the nodes through their attribute list. The connector manages the read access to nodes by creating a file descriptor for the respective image stream and redirecting the `read()`, `seek()` and `tell()` functions of the node to this descriptor. The descriptor is created by the resolver and executes in the context of the AFF4 library. The nodes will be arranged in the same hierarchical order that existed before acquisition. This implies that any filesystem structure that was parsed before acquisition will also be restored.

The verifier is an independent application, performing verification of provenance of the data objects in a partial image. For this purpose, the original device from which the partial image was acquired from is connected to the analysis workstation and a comparison of data objects on the device and in the image is performed. This procedure is illustrated in Figure 4.5. The verifier extracts the provenance documentation that is stored in the partial image. It then reads data from exactly those sectors on the device, that are documented as the source of a data object in the image. This data is hashed with the same algorithm that was used to create the verification metric in the

## 4.2 Architecture

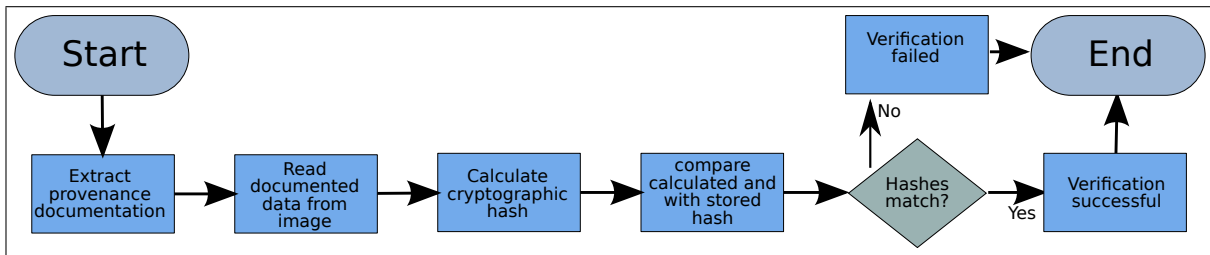


Figure 4.5: Verification Procedure for Partial Images

image and the result is compared with this stored hash. Due to the characteristics of the hashing algorithm, the stored hash will only match the calculated hash, if the data that was acquired in the image is exactly the same as the data that is read during the verification procedure. This enables examiners to prove the correctness of the provenance information.

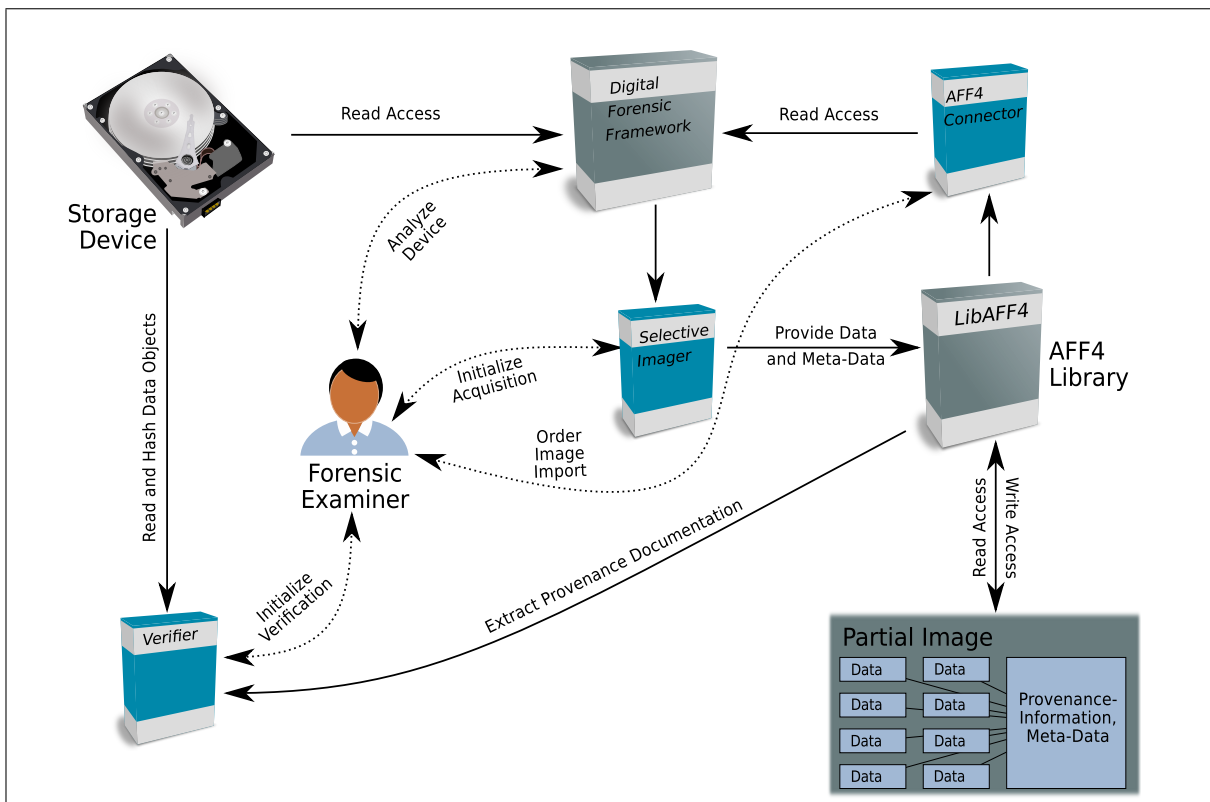


Figure 4.6: Architecture of the Selective Imager

The interaction of all components is illustrated in Figure 4.6. The selective acquisition procedure begins with connecting a storage device to the forensic workstation. While this connection should always be secured by a hardware write-blocker, the Digital Forensic Framework always opens devices in read only mode and thus not only relies on hardware measures to insure the integrity of the device. The forensic examiner then uses DFF to perform preliminary recovery, harvesting and reduction procedures. When a selection

has been made, the selective imager is invoked. The selective imager will retrieve the metadata of all data objects by parsing the attributes of the respective nodes. It then will pass this information to the resolver in the AFF4 library, who will in turn write the data to the image.

After the image is completed, the device can be disconnected from the forensic workstation and further analysis can be conducted on the image. Because the selective imager utilizes the official AFF4 library to create the image, any tool that supports AFF4 in general can be used to process it. To load these images into DFF, we created a connector to interface with the library. The AFF4 Connector is a module that analyzes AFF4 images and creates a node for every stream inside. It then retrieves the RDF Graph for this stream, and attaches any DFF-compatible fact to the node as an attribute. DFF-compatibility in this context means the RDF Fact is in the DFF namespace. This import procedure will result in a VFS tree similar to the one that existed before acquisition. The nodes will be arranged in exactly the same order that existed before acquisition and all attributes that existed before will also be restored. This is of course restricted to the selection, nodes that were not selected for acquisition will not be referenced or stored in the partial image and thus not appear in the VFS tree on import of the partial image. Since modules store their results either as a node or as an attribute to a node, all preliminary analysis results are stored in the partial image.

To determine the integrity and provenance of data objects in the partial image, the verification program is used. It is a simple console program, that will use the aff4 library to extract a list of all streams and their provenance documentation. It will then read and hash the byteruns for all streams in the image and verify if their hash matches the one stored in their provenance documentation. This allows examiners to quickly prove the correct implementation of forensic procedures.

## 4.3 Implementation Details

In this section, the operational details of important forensic sub-procedures of the selective imager are described. The implementation of the actual copying and the underlying functions of DFF, the import process for images, the provenance documentation and the extraction of metadata are discussed.

### 4.3.1 Data-Copying

The exact implementation of Data-Copying is very important from a forensic point of view, because accidental writing to evidence devices or images can result in the corruption of evidence and most likely in the rejection of evidence in court. The Digital Forensic Framework uses the POSIX low level file access library `fcntl.h` [58] to open devices or images for reading. As can be seen in Listing 4.2 in Line 8, when a node is opened a read only file-descriptor is created by passing the `O_RDONLY` flag to the library. This effectively prevents any write access to the file through this descriptor, even if some other code tries to write to it later.

```
1 #include <fcntl.h>
2 ...
3 int local::vopen(Node *node)
4 {
5     int n;
6     std::string file;
7     file = lpath[node->id()];
8     n = open(file.c_str(), O_RDONLY)
9     return (n);
10 }
```

Listing 4.1: How DFF Opens Devices/Images (`local.cpp`)

Read access to devices or files is also realized through `fcntl`, as can be seen in Listing 4.2. There is no write function and the employed library guarantees not to change anything in the target file. In theory, these mechanisms will ensure read only access to devices, so a hardware write-blocker is not strictly necessary. However, there are no guarantees, the operating system or other software will not try to access the device in other ways. It is advised to employ hardware-based write-blocking in any case, to assure the integrity of connected evidence.

The selective imager uses this interface to read data from objects during imaging. This ensures that the imager can not accidentally modify devices it images.

```
1 #include <fcntl.h>
2 ...
3 int local::vread(int fd, void *buff, unsigned int size)
4 {
5     int n;
6     n = read(fd, buff, size);
7     return n;
8 }
```

Listing 4.2: How DFF Reads From Devices/Images (`local.cpp`)

### 4.3.2 Provenance and Meta-Data

The storage of provenance documentation and metadata is realized by attaching RDF-Facts to each data object. Each fact stores a named value that is associated with an entity in the volume. The resulting RDF-Graph is serialized using the TURTLE [9] notation. Listing 4.3 shows an excerpt from the RDF-Graph of a JPEG file in a partial image, created by the selective imager.

As stipulated in Section 3.2.2, the selective imager needs to store multiple provenance metrics. We implemented a generalization of the block-address scheme, where a triple

```
1 @base <aff4://ca3f8ff0-46bf-410b-b97a-c4553dae69a6> .
2 @prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
3 @prefix dff: <http://digital-forensic.org/dff#> .
4 </01540_driftwood_1920x1200.jpg>
5     dff:MFT entry number "45";
6     dff:MFT physical offset "62464";
7     dff:byteruns "fileoffset/imgoffset/len:0/31383552/1455304";
8     dff:accessed "2010-10-25T13:50:04"^^xsd:dateTime;
9     dff:created "1970-01-01T00:59:59"^^xsd:dateTime;
10    dff:modified "2010-02-21T08:44:41"^^xsd:dateTime;
11    dff:deleted "False";
12    dff:hash-md5 "22fe330688a8459a2728c1ff0aea8378";
13    dff:mime-type "image/jpeg; charset=binary";
14    dff:name "01540_driftwood_1920x1200.jpg";
15    dff:path "/part1.dd/NTFS/pics/";
16    dff:size "1455304";
17    dff:type "JPEG image data, JFIF standard 1.02";
```

Listing 4.3: RDF Sample Meta-Data for a JPEG file (simplified)

defines the exact location of all bytes both on the device and in the image. If the data-object is fragmented, this metric will consist of a list of multiple triples, each identifying a continuous block of data. Each element consists of the address on the device where the block came from, the address where it is positioned in the data object and its length. An array of triples, describing the exact provenance of every byte inside a data object is referred to as its *byteruns*. This attribute can be seen in line 7 in Listing 4.3. The selective imager combines this primary provenencial key with a secondary key. The secondary key is the *path* of the data object in the VFS. This is more general than the filesystem path, as data objects are not necessarily files and thus might not have a filesystem path. Also, the VFS path unambiguously documents the partition a data object was obtained from and the type of the filesystem it was stored in, if it is a file. In the provided example, the path is stored in line 15 and the data object is a jpeg file that resides on the first partition, inside a NTFS filesystem in the folder */pics/*.

Additionally to the provenance documentation, the selective imager stores any attribute a data object has in the VFS. In this case, the example file was analyzed with modules to extract metadata from the filesystem. The file-size and mactimes where stored as an attribute and thus serialized into the RDF-Graph of the node. Also, a reference to the relevant data structure of the filesystem called the MFT entry is stored. This allows to re-examine and verify this information, if the filesystems data structures are also acquired. Also a file-magic cataloging module was applied to the file, to determine its contents independently to its file-extension. The result of this module is stored in the *type* attribute and identifies the file as a JPEG compressed picture. Information other than text is serialized using the XML Schema Document notation (XSD) [69]. When reimporting the image, these data-types are restored and thus are still manipu-

lable and searchable in their natural way. For example, it is still possible to filter data objects by date and display all objects that were accessed between the first and the third of May 2010.

### 4.3.3 Image Creation

Selection in DFF is realized with a bookmark system. Examiners can add nodes they wish to acquire to a bookmark category. When the selection is complete, the selective imager is invoked on the root of the relevant bookmark category. This root node is passed to the selective imager as a parameter. The node and all its children will be acquired into the partial image. Not all nodes in the VFS contain data, some exist only for organizational purposes. For example, directories in the VFS are also nodes, but do not contain any data.

```
1 def getFileNodes(self, root):
2     nodes = []
3     processNodes = []
4     processNodes.append(root)
5     while(len(processNodes) != 0):
6         currNode = processNodes.pop()
7         name = currNode.name()
8         children = currNode.children()
9         processNodes.extend(children)
10        if(not currNode.size() <= 0):
11            nodes.append(currNode)
12    return nodes
```

Listing 4.4: Function `getFileNodes()` (`acquire.py`)

To obtain a list of all nodes that contain data in the selected subtree, the function `getFileNodes()` recursively traverses the nodes below the selected root node. The code for this function is presented in Listing 4.4. The function manages two queues, `nodes` and `processNodes`. For every node starting with the supplied root node, all children are added to the `processNodes` queue. If the size of the node is larger than 0, it is added to the `nodes` queue, because it contains data. The function is named `getFileNodes()`, because a node that contains data is classified as a *Filelike Object* in AFF4.

When the list of selected nodes is completed, the imager checks output parameter, provided by the user. This is illustrated in Listing 4.5. It checks if the output is a file and creates an AFF4 zipfile volume if positive, which can be seen in Line 5 of the listing. Otherwise the output is simply opened. This is realized in Line 9 of the listing and has been implemented for compatibility with other features of AFF4, like volumes that are accessed through a network.

After the image volume has been created, the selective imager will extract the meta-data for each node in the selection. This is performed by the function `getMetaData()`, which is shown in Listing 4.6. Because a node in the VFS can also be a link to another

```
1 out_fd = self.resolver.open(output_URI, 'w')
2 # If output URI is file
3 if isinstance(out_fd, pyaff4.FileLikeObject):
4     # create volume
5     volume_fd = self.resolver.create(pyaff4.AFF4_ZIP_VOLUME
6                                     , 'w')
7     self.resolver.set_value(volume_fd.urn, pyaff4.AFF4_STORED
8                             , output_URI)
9     volume_fd = volume_fd.finish()
10 else:
11     volume_fd = out_fd
```

Listing 4.5: Image creation (acquire.py)

```
1 def getMetaData(self, node):
2     while(node.isVLink()):
3         node = node.vlink().linkNode()
4     meta = {}
5     meta["name"] = node.name()
6     meta["size"] = node.size()
7     meta["deleted"] = node.isDeleted()
8     meta["parent"] = node.parent().absolute()
9     meta["path"] = node.path()
10    self.fillStaticAttributes(meta, node)
11    self.fillExtendedAttributes(meta, node)
12    self.fillTimes(meta, node)
13    self.fillByteRuns(meta, node)
14    return meta
```

Listing 4.6: Extraction of metadata (acquire.py)

### 4.3 Implementation Details

---

node, the function first needs to resolve these links. This is performed in Line 2 and 3. The bookmarking feature of DFF for example uses links to aggregate a user defined selection of nodes. To extract accurate path information on these nodes, the imager needs to resolve the links first. The function then extracts some attributes of the node directly by calling the respective functions, as shown in Line 5 to 9. Finally, some functions are called to extract attributes that can not be obtained directly.

```
1 def fillByteRuns(self, meta, node):
2     # get the file mapping from the node
3     fm = FileMapping()
4     node.fileMapping(fm)
5     parent = node.parent()
6     grandparent = parent.parent()
7     # Translate filemappings until
8     # they are independent from others
9     while (grandparent.name() != '/'):
10         fm = self.translateBR(fm, parent, node)
11         parent = grandparent
12         grandparent = grandparent.parent()
13     byteRuns = "fileoffset/imgoffset/len:"
14     # extract a list of chunks for the file
15     chunks = fm.chunks()
16     for chunk in chunks:
17         byteRuns += "%d/%d/%d " % (chunk.offset, chunk.
18                                     originoffset, chunk.size)
19     meta["byteruns"] = byteRuns
```

Listing 4.7: Extraction of byteruns (acquire.py)

An example for such an extraction routine is the function `fillByteRuns()` in Line 13 of Listing 4.6. Its code is shown in Listing 4.7. The function first obtains a copy of the nodes `filemapping`. The `filemapping` is a data structure in DFF that maps the continuous stream of bytes in a node to chunks in the parent node. Not all nodes have a `filemapping`, as some actually represent a file or device on the analysis system. However, any node that is derived from another, for example a file that is derived from a filesystem, has a `filemapping`. If the parent of a node is in the root directory, it is always backed by a file or device and does not have a `filemapping`. In this case, the `filemapping` of the node is relative to the file or device and accurately documents its provenance. If the parent of the node is not in the root directory, it can also have a `filemapping`. In this case, the nodes own mapping is relative to the parents. To document the provenance of a node relative to the device it resides on, the mapping has to be recursively translated for each parent that has a `filemapping`. This is performed by Line 9 to 12 in Listing 4.7. After the translation is completed, the function serializes the addresses of the nodes chunks and stores them in the `byteruns` attribute, as shown in Line 15 to 18.

After these steps, all the required metadata is available. The selective imager then

## 4.3 Implementation Details

uses the `add_value()` function in `libAFF4`, to store each value in the image and associate it with the image stream.

Finally, the data is written to the image. This happens in the function `image()` and is shown in Figure 4.8. To provide a verification metric, the selective imager simultaneously uses `hashlib` to calculate a hash of the copied data. `Hashlib` is an implementation of cryptographic hashes in the Python Standard Library [53]. When the selective imager finishes writing a data object to the image, it appends the resulting `md5`-hash to the `RDF-Graph` of the object. This procedure is illustrated (in a very simplified fashion) in Listing 4.8. The stored hash can be seen in Line 12 of Listing 4.3. The interleaved hashing and writing of data ensures that the hash represents the exact data that is written to the image. If the imager were to write the image first and then read the data again to calculate the hash, possible read-errors for damaged sectors on the device could lead to a non-matching hash. Additionally, this approach is faster and ensures minimal load to the device, as the data has to be read only once. Despite the `md5`-hash being insecure [65], the selective imager uses this algorithm by default to stay compatible with known-file hash libraries such as the NIST NSRL [67]. The only alternative that is also widely used in known-file libraries is `sha1`, which has also been broken [66]. Examiners can use the hash documented during acquisition to verify the integrity of every data object in the partial image at any given time. This is achieved by calculating the hash of this object again from the data in the image. If the hash acquired in this way matches the acquisition hash, the integrity of the data object is verified.

```
1 def image(self, node):
2     image_fd = resolver.create(pyaff4.AFF4_IMAGE, 'w')
3     fd = node.open()
4     while 1:
5         data = fd.read(BLOCKSIZE)
6         if not data: break
7         hasher.update(data)
8         image_fd.write(data)
9     hash = str(hasher.hexdigest())
10    resolver.add_value(image_fd.urn, "hash-md5", hash)
```

Listing 4.8: Selective Imager Acquisition Code (`acquire.py`)

### 4.3.4 Image Parsing

The `aff4` connector is a module that parses the partial images created by the selective imager and imports them into `DFF`. It consists of three classes:

- `Aff4ImgStream`
- `Aff4Node`
- `AFF4`

### 4.3 Implementation Details

---

```
1 class Aff4ImgStream(fso):
2     """represents an aff4 image stream in the node tree"""
3     def __init__(self, resolver, urn):
4         self.resolver = resolver
5         self.urn = urn
6         self.fd = None
7
8     def vopen(self, node):
9         self.fd = self.resolver.open(self.urn, 'r')
10        return 1
11
12    def vread(self, fd, buf, size):
13        buf = self.fd.read(size)
14        size = len(buf)
15        return (size, buf)
16    ...
```

Listing 4.9: AFF4 Image Stream Wrapper (aff4.py)

The class `Aff4ImgStream` is a wrapper for libAFF4 filedescriptors. An excerpt of the code is shown in Listing 4.9. The class implements a standard interface for data access in DFF and redirects requests to libAFF4. The function `vopen()` calls the resolver in libAFF4 to obtain a filedescriptor for the image stream of a node. The function `vread()` uses this filedescriptor to read data from the image stream. Similar to these exemplary functions other wrapper functions for `vseek()`, `vtell()` and `vclose()` exist, that also redirect to libAFF4.

```
1 class Aff4Node(Node):
2     """ a single aff4 image stream with metadata """
3     def __init__(self, name, size, parent, imgStream, metadata):
4         Node.__init__(self, name, size, None, imgStream)
5         self.metadata = metadata
6         self.__disown__()
7     ...
8     def extendedAttributes(self, attr):
9         for (key, val) in self.metadata.iteritems():
10            vval = Variant(val)
11            attr.push(key, vval)
```

Listing 4.10: AFF4 Node Class (aff4.py)

The class `Aff4Node` is derived from the standard node class in DFF, to account for the special characteristics of an AFF4 image stream object. Its main purpose is to store the filedescriptor and metadata of an image stream, to provide access to them through the standard node interface. The constructor in Line 3 to 6 stores references

### 4.3 Implementation Details

---

to the metadata and registers the filedescrptor with the node interface (Line 4). The function `extendedAttributes()` in Line 8 to 12 is called by DFF to obtain the nodes attributes. It is given a reference to the attribute storage (`attr`), which is accessed in Line 12 to provide the metadata to DFF. For compatibility reasons, the metadata has to be packaged into a generalized object (`Variant`), which is performed in Line 10.

```
1 for stream in streams:
2     metadata = self.getXSDMetaData(stream)
3     imgStream = Aff4ImgStream(self.resolver, stream)
4     parentPath = metadata["parent"]
5     name = metadata["name"]
6     size = int(metadata["size"])
7     # find the parent dir or create one
8     parent = self.getParentNode(parentPath)
9     strNode = Aff4Node(name, size, parent, imgStream, metadata)
10    # need to register this node with its parent
11    parent.addChild(strNode)
```

Listing 4.11: AFF4 Connector import code (`aff4.py`)

The class **AFF4** contains the actual functionality to parse and import AFF4 images. It is responsible for the creation of nodes and the restoration of the VFS directory structure. Each data object in the image is stored in an image stream. For each stream, a node needs to be created and its metadata has to be made accessible to DFF. The simplified code for this procedure is shown in Listing 4.11. First, the metadata is retrieved. This is managed by the function `getXSDMetaData()`, which extracts and de-serializes the metadata. The module then creates a wrapper for the AFF4 filedescrptor in Line 3. The function `getParentNode()` parses the stored VFS path of the stream. If the directory structure documented in the path attribute does not already exist in the VFS, it is created. Finally, an `AFF4Node` is created and linked into the directory structure by calling the `addChild()` function of the parent node.

#### 4.3.5 Provenance Verification

Examiners have to be able to prove the correctness of the techniques employed during acquisition. Especially the provenance documentation must be verifiable. To verify the provenance of a data object, examiners need at least one proveniential key, the cryptographic hash and access to the original device. The data documented with the proveniential key is extracted from the device and hashed. The resulting hash is then compared to the hash that was obtained during the acquisition of the data object. If the hashes match, the data object in the image is identical to the data object on the device, described by the proveniential key.

The partial image verifier is a console application that implements this technique to verify each stream in a partial image automatically. In Listing 4.12, an excerpt of the verification routine is illustrated. The verifier first extracts the stored hashes and

```
1 for imagestream in verification_streams:
2     byteruns = Byteruns(imagestream.metadata["byteruns"])
3     oldhash = imagestream.metadata["hash-md5"]
4     hasher = hashlib.md5()
5     for run in byteruns.runs:
6         # seek to the offset on the device
7         dev.seek(run[1])
8         # read a chunk of lenght=len of the byterun
9         data = dev.read(run[2])
10        hasher.update(data)
11    streamhash = hasher.hexdigest()
12    # Compare hash with stored record
13    if(streamhash == oldhash):
14        self.r.render("SUCCESS:")
15    else:
16        self.r.render("ERROR!")
```

Listing 4.12: Partial Image Verification Code (aff4verify.py)

byteruns from the image in Line 2 and 3. The byteruns document the exact location and order of each byte in the data object, stored in the image stream. The verifier then seeks to the position of each chunk documented in the byteruns and extracts it from the device. The data is passed to a hashing module, which calculates an md5 hash. When the extraction is finished, the hash is compared to the one stored in the image. If the hashes match, the provenance documentation is verified.

## 4.4 Development Facts

The selective imager was developed as a reference implementation of an imager that is able to create partial images with arbitrary granularity. The implementation consists of three programs:

- The Imaging Module (acquire.py)
- The AFF4 Connector (aff4.py)
- The Partial Image Verifier (aff4verify.py)

The imaging module and the AFF4 Connector are implemented as a Python plug-in for DFF. The verification program for partial images is a Python console application, that operates independently from DFF. To access and create AFF4 images, all developed programs use the publicly available version of libAFF4. The library was obtained in November 2010 from the official Google Code repository [18]. Aside from components of the DFF API and the Python Standard Library, the modules do not use any other libraries or imports.

DFF runs on multiple platforms, at the moment Windows and Linux builds are available. Because the developed modules are written in Python, they are platform independent in theory. Practically, they rely on the Python bindings of libAFF4, which have been developed for Linux systems and do not compile on other platforms at the moment. If the library would be ported to other platforms that are supported by DFF, the selective imager would be equally functional there.

The implementation is designed to work with Version 0.9 of DFF, which was the current version at the time of development. The source code of all software developed in course of this thesis is available in Appendix B. It is also publicly available in the branch `mod_aff4` of the DFF Git repository [4]. The code is licensed under version 2 of the GNU General Public License. A guide on compilation and application of the software can be found in Appendix D. It is also published in the DFF wiki [57]. For testing purposes a Live-DVD is available in Appendix C. It contains a working installation of all tools, which were developed in course of this thesis.

4.5 Tool Usage

This section documents the most common tasks that forensic examiners will execute in the context of selective imaging. The usage of the selective imager for creation and import of partial images is explained, as well as the verification procedure for partial images.

4.5.1 Image Creation and Import

The creation of partial images is the central procedure in selective imaging. After examiners perform recovery and harvesting, the reduction procedure will yield a selection of data objects, that are likely to be of use to the investigation. In DFF, this selection is realized through the bookmarking feature. During reduction, examiners can check the selection box next to the nodes that are regarded as relevant.

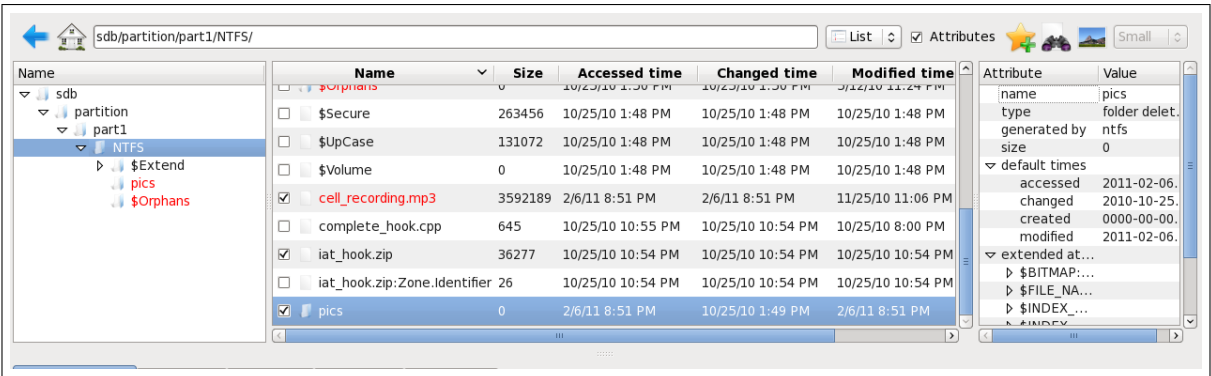


Figure 4.7: Selection of Evidence in DFF

In Figure 4.7, this procedure is illustrated with the selection of several files in an NTFS filesystem. Examiners have performed recovery operations and the file `cell_recording.mp3`

## 4.5 Tool Usage

and the folder `pics` have been recovered from the filesystem. They are marked red, because they have previously been deleted. Some information from the filesystems data-structures have been gathered during the harvesting phase, for example mactimes and information from the Master File Table, which is the central management data structure of the NTFS filesystem. This information is stored in the attributes of the nodes and can be seen in the right content pane of the application.

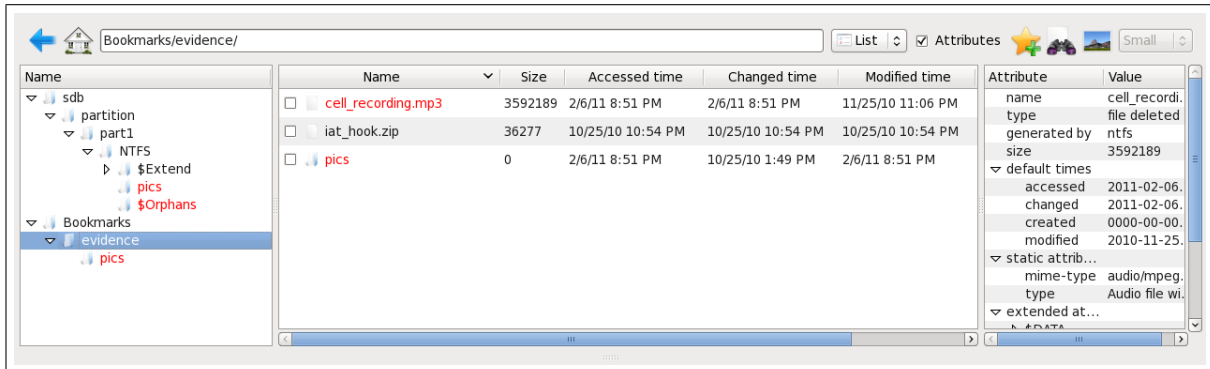


Figure 4.8: Acquisition of Evidence in DFF

Examiners have checked several files and folders and in the next step added them to a bookmark category called *evidence*. In Figure 4.8, the selection is depicted. Examiners can now apply the acquisition module called **acquire** to the root node of the bookmark tree. The selective imager will then write each bookmarked node into a partial image, appending any metadata that exists as attribute.

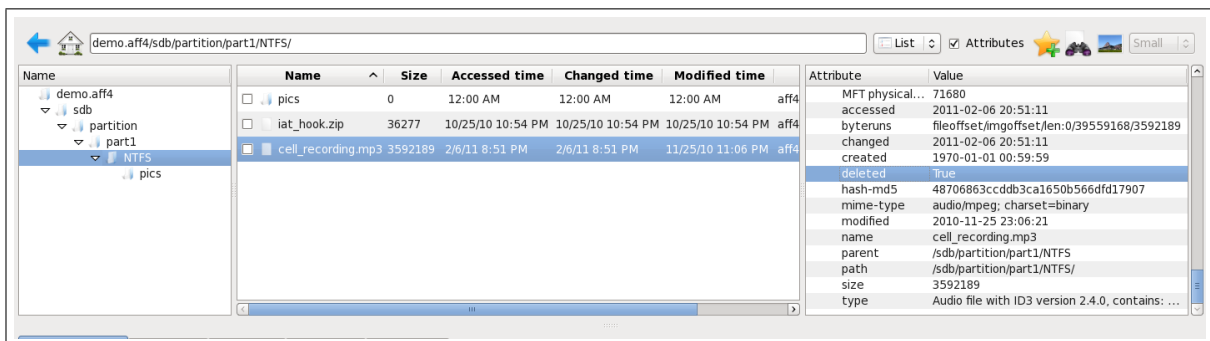
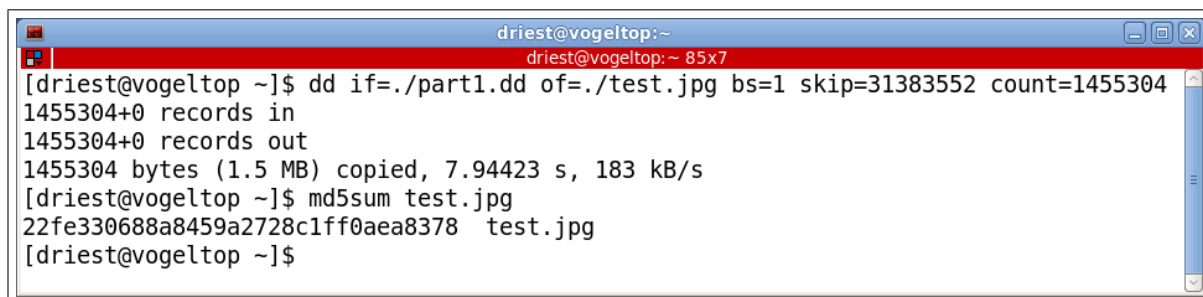


Figure 4.9: Import of partial image in DFF

The image created this way can be re-imported into the VFS at any time, by adding the image-file to the VFS and applying the aff4 connector module. The contained data objects will then be restored in their original hierarchy, determined by their path in the VFS immediately before the acquisition took place. The resulting VFS subtree will be an exact copy of the pre-acquisition tree but without the nodes that were excluded from the selection. Figure 4.9 shows the state of the VFS after the import of the image, created with the previous selection.

### 4.5.2 Verification

Examiners have to be able to prove that they have not altered the acquired digital evidence in any way, when challenged in court. This can be accomplished by verifying the stored hash and the provenance documentation of the data objects. The stored hash can be verified by calculating a new hash on the stored data and comparing it with the stored one. This can be achieved either by using the hashing module in DFF or by extracting the data of a node and using an external program to perform the calculation.

A screenshot of a terminal window titled 'driest@vogeltop:~'. The terminal shows a series of commands and their outputs. The first command is 'dd if=./part1.dd of=./test.jpg bs=1 skip=31383552 count=1455304', which outputs '1455304+0 records in', '1455304+0 records out', and '1455304 bytes (1.5 MB) copied, 7.94423 s, 183 kB/s'. The second command is 'md5sum test.jpg', which outputs '22fe330688a8459a2728c1ff0aea8378 test.jpg'. The prompt returns to the user's shell.

```
driest@vogeltop:~  
driest@vogeltop:~ 85x7  
[driest@vogeltop ~]$ dd if=./part1.dd of=./test.jpg bs=1 skip=31383552 count=1455304  
1455304+0 records in  
1455304+0 records out  
1455304 bytes (1.5 MB) copied, 7.94423 s, 183 kB/s  
[driest@vogeltop ~]$ md5sum test.jpg  
22fe330688a8459a2728c1ff0aea8378 test.jpg  
[driest@vogeltop ~]$
```

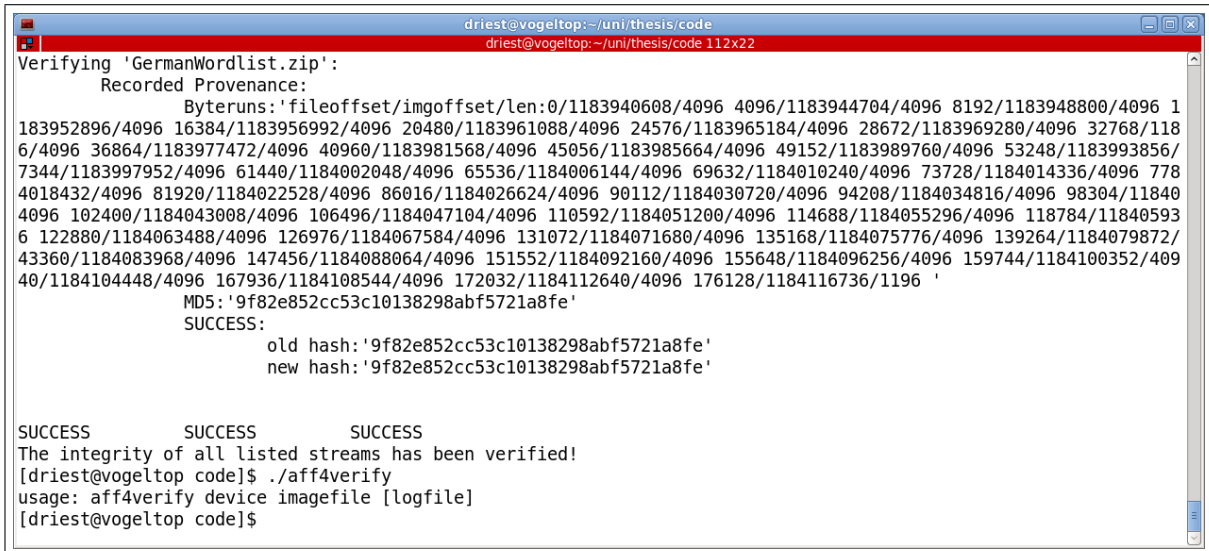
Figure 4.10: Simple Image Verification of Sample File from Listing 4.3

Additionally to the verification of the stored hash, the documentation of provenance has to be verified. This is accomplished by connecting the original device to the forensic workstation and extracting the data documented in the provenance documentation from the device. The extracted data is then hashed and the hash is compared with the one stored in the image. If the hashes match, the provenance is verified.

The easiest way to verify the provenance of a file that is not fragmented, is to use a simple copying tool such as `dd` to extract the bytes at the addresses documented in the *byteruns* attribute from the original device. A md5 hashing application such as `md5sum` can then calculate the hash of the extracted data. If the calculated hash matches the stored hash in the image, the provenance of the file in the image is verified. Figure 4.10 demonstrates this technique, by extracting the byteruns of the sample JPEG file from Listing 4.3.

The verifier we developed automates this process. Figure 4.11 shows a screenshot of a test run of the verifier. The verifier takes 3 parameters. The first is the device from which the data is extracted. The second is the image file that is verified. The third parameter is optional and specifies the path to a file where the verifier will log its output.

Examiners can also use the second provenential key, the VFS path, to verify the image. This provenential key is better suited for manual verification, as it is on a level very close to the one on which people normally interact with digital storage. The file can be extracted from the specified path on the device with any forensic tool suitable and the hashes can then be compared in a similar way as in the previous method. However, this method of verification does require parsing partition- and filesystem-structures, which makes it more complicated to automate.



```
driest@vogeltop: ~/uni/thesis/code
driest@vogeltop: ~/uni/thesis/code 112x22
Verifying 'GermanWordlist.zip':
Recorded Provenance:
  Byteruns: 'fileoffset/imgoffset/len:0/1183940608/4096 4096/1183944704/4096 8192/1183948800/4096 1
183952896/4096 16384/1183956992/4096 20480/1183961088/4096 24576/1183965184/4096 28672/1183969280/4096 32768/118
6/4096 36864/1183977472/4096 40960/1183981568/4096 45056/1183985664/4096 49152/1183989760/4096 53248/1183993856/
7344/1183997952/4096 61440/1184002048/4096 65536/1184006144/4096 69632/1184010240/4096 73728/1184014336/4096 778
4018432/4096 81920/1184022528/4096 86016/1184026624/4096 90112/1184030720/4096 94208/1184034816/4096 98304/11840
4096 102400/1184043008/4096 106496/1184047104/4096 110592/1184051200/4096 114688/1184055296/4096 118784/11840593
6 122880/1184063488/4096 126976/1184067584/4096 131072/1184071680/4096 135168/1184075776/4096 139264/1184079872/
43360/1184083968/4096 147456/1184088064/4096 151552/1184092160/4096 155648/1184096256/4096 159744/1184100352/409
40/1184104448/4096 167936/1184108544/4096 172032/1184112640/4096 176128/1184116736/1196 '
MD5: '9f82e852cc53c10138298abf5721a8fe'
SUCCESS:
  old hash: '9f82e852cc53c10138298abf5721a8fe'
  new hash: '9f82e852cc53c10138298abf5721a8fe'

SUCCESS      SUCCESS      SUCCESS
The integrity of all listed streams has been verified!
[driest@vogeltop code]$ ./aff4verify
usage: aff4verify device imagefile [logfile]
[driest@vogeltop code]$
```

Figure 4.11: Automated Verification of Partial Image

## 4.6 Summary

In this chapter, we described the design, implementation and operation of the selective imager and its auxiliary tools. To achieve a high acceptance among forensic examiners, we decided to extend an existing forensic framework by creating a plug-in. We analyzed the different available tools and chose to use the Digital Forensic Framework (DFF), mainly because of the active developer base, its extensive features and the ease of extensibility. To maximize compatibility and simplify crosschecking of results with other forensic tools, we decided to use an existing forensic format to store the partial images. After analyzing the different available formats, we decided to use the AFF4 format, because it is the only open standard that is capable of storing multiple data objects and arbitrary metadata inside a single container. The AFF4 format is managed through an open source library, providing functions for the creation of images and access to stored information. Metadata is stored in form of an RDF-Graph, that consists of triples describing attributes of specific objects.

We then explained the architecture of the Digital Forensic Framework. The framework itself only provides limited services and a graphical user interface, all forensic functionality is provided by pluggable modules. Data is managed inside a virtual filesystem (VFS) and is organized as a tree of nodes. Each node can represent any arbitrary data object, be it a directory, a file or even a logical mapping of data. Metadata on nodes is stored in an attribute system that associates name value pairs with them.

The selective imager integrates into these frameworks. The acquisition module is a plugin to the Digital Forensic Framework. It uses the standard node access functions to read data from objects in the VFS and obtains all metadata through the nodes attribute system. The imager then accesses the AFF4 library to create the image and write data to it. The metadata is stored in the image by serialization of an RDF-Graph of attributes, the AFF4 library associates with each data object.

We also developed a connector module, to load images back into the VFS. The module employs the AFF4 library to provide access to partial images from within DFF. For this purpose, it extracts the metadata from the image and re-creates the node hierarchy in the exact same way it existed before acquisition. Read access is wrapped around functions in the AFF4 library and provided through the standard node interface. This allows examiners to continue their work from before the acquisition phase with the same tools. The transition of examination, from the device to the image, does not require any change in software and can be picked up right where it was interrupted to acquire the image.

For verification purposes, we developed the partial image verifier. This program extracts the provenance documentation and verification metric of all streams from a partial image. The provenance documentation is used to extract exactly those bytes from a connected device, that are documented to be the source of the data object in the image. The extracted data is then hashed and the resulting hash is compared with the hash that was calculated during acquisition of the data object. If these hashes match, the data object originated from the documented regions on the device and is considered as verified.

The selective imager uses standard DFF node access functions to read data from attached devices. We showed that DFF uses low level file access functions from the POSIX library to prevent accidental writes and thus makes it impossible for the selective imager to modify connected devices during acquisition. The provenance documentation for data objects is realized in a redundant way, as the selective imager records both byteruns and the VFS path. The byteruns exactly document where data originated from on the device and where it is stored in the data object. The VFS path provides the same information on a higher level of abstraction, documenting the partition a data object is stored on, the filesystem and its path inside the filesystem.

Furthermore, we explained the *modus operandi* for the most common tasks that can be performed with the developed tools. The bookmarking and selection of evidence is shown, as well as the creation of partial images from DFF, the import and their verification.

Finally, we described some facts on the development of the selective imager. We listed the components of the developed software and gave an overview on the lines of code and the created documentation.

The implementation developed with this thesis makes selective imaging available for everyone, without the need for proprietary solutions. In the next section we evaluate the performance, correctness and implications for forensic practitioners using these tools.

# 5 Evaluation

In this chapter, the performance and practicability of the software developed in the course of this thesis, as well as the selective imaging approach in general is evaluated. In Section 5.1, the benefits gained from selective imaging are measured. Especially, the time and disk-space savings required to create partial images are assessed and compared to the common approach of sector-wise images. Section 5.2 analyzes the technical performance of the selective imager. The raw transfer speed is measured and compared to other solutions. Also, the reliability of the created images and the implications of selective imaging on disk wearing are analyzed. Section 5.3 assesses the practical acceptance of the technique. The selective imager is presented to forensic practitioners and their opinion and attitude towards the approach is evaluated. Furthermore, a questionnaire is handed out to forensic examiners and the results are analyzed.

## 5.1 Quantification of Benefits

In Section 3.1.3, we identified multiple groups of crime and assessed the benefits, investigators can gain when employing selective imaging in these cases. We concluded, that some cases require advanced recovery techniques and this will benefit less from selective imaging, because the selection has to be very broad. Other cases, where the objective is very narrow and evidence is not concealed, will benefit more significantly from selective imaging. In this section, the selective imager is employed in simulated investigations to quantify these assumptions.

### 5.1.1 Test Data

The evaluation is based on two test devices. The first exhibit is a Seagate ST320423A PATA hard-disk with a capacity of 20.4 GB. It was acquired in the scope of a student forensic analysis project on ebay. Because the previous owners deliberately sold the device, they have taken measures to permanently delete all data on it. The old filesystem was deleted and a new empty NTFS filesystem was created. This forces examiners to employ file-carving and other more complicated recovery techniques to extract any data from the device at all. The device contains a lot of office documents and compressed pictures (JPEG), that cover personal details of the life of the previous owners. Because this exhibit requires extensive pre-acquisition recovery, harvesting and reduction, the device simulates a rather complicated case.

The second exhibit is a Kingston Data Traveler G2 USB-Flash-Drive with a capacity of 4 GB. It was created deliberately for the testing of forensic procedures and contains mul-

multiple partitions with several different filesystems and files. Some files have been deleted, but no measures have been taken to prevent the recovery of data. The device stores pictures, audio files and programs, containing simulated evidence of illegal activity. Other than that it is filled with innocent data like Linux distribution installation media. Due to the evidence not being concealed, this device simulates a relatively straightforward case that will benefit a lot from selective imaging.

These two examples are realistic but arbitrary and not necessarily representative. They were chosen, because they simulate two opposite cases and the analysis results allow to judge a large number of similar cases.

### 5.1.2 Disk Space Requirements

To evaluate the disk space requirements to store both sector-wise and selective images, we performed a forensic analysis on the test exhibits. The objective in the first case is to find all data that appears to have personal information on the previous owner of the device. The device was analyzed very thoroughly and a lot of personal data was recovered, especially office documents and personal pictures. The final selection had a size of 1178 Megabytes, which amounts to roughly 5.7 percent of the device capacity. This results in 94.3 percent less required storage capacity for the image. However, since there are no guarantees that we covered 100 percent of the relevant data, we would be uncomfortable returning the exhibit to its owner before the case is closed. If that became necessary at some point, a complete image would have to be acquired.

The second case is of significantly lower complexity. The recovery was limited to files that are still listed in the filesystems data structures. No carving of empty space was performed. While some files had been deleted, they could be recovered by examining filesystem data structures. The objective is to retrieve any pictures, audio files, text and programs from the exhibit, that seem related to illegal activity. The selected data had a size of 18.7 Megabytes, which amounts to about 0.4 percent of the device capacity. The savings in image storage space therefore amount to 99.6 percent.

### 5.1.3 Speed of Investigation

The acquisition of a raw image with the tool `dd_rescue` took 25 minutes for the Seagate hard-disk. Acquisition of the selection with the selective imager took 77 seconds. This shortens the imaging step to about 5 percent of its original duration. Nevertheless, to judge the benefits from improvements in imaging speed in regard to the overall duration of the investigation, the pre-acquisition phase also needs to be considered. The entire pre-acquisition phase took 75 minutes. The first five minutes, we determined the structure of the partition table and filesystem. We then spend 40 minutes on file-carving and 30 minutes to assess and select relevant data. This sums up to a period of 76:17 minutes, until first results were available.

Because the raw image is identical to the physical device from a forensic perspective, it is safe to assume it takes an investigator an equal amount of time to deliver similar results when performing the same investigative steps on a raw image instead. When a

## 5.1 Quantification of Benefits

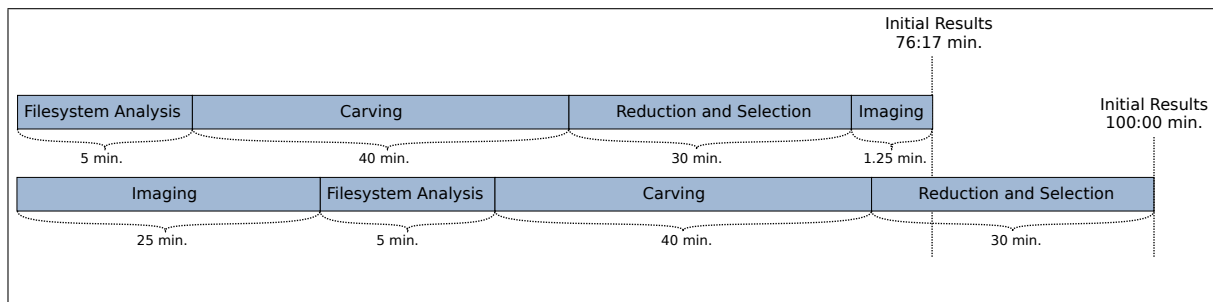


Figure 5.1: Imaging of a 20GB disk, speed comparison

raw image is used, the period until initial results become available is 100 minutes long, as the 25 minutes of acquisition time add to the analysis time of 75 minutes. The difference is illustrated in Figure 5.1. Preliminary results with a selective imaging approach can be delivered 23.75 percent faster than with a sector-wise approach. Since all employed steps scale almost linearly with the amount of data they operate on, it is safe to assume this figure as the average savings in this type of investigation.

The acquisition of the flash-drive into a sector-wise image took 4 minutes, due to its small size and relatively high speed compared to the hard-disk. The selective imager took 3 seconds to acquire the potential evidence. This is an even greater time improvement of 98.75 percent in the imaging step. The time spent on the pre-acquisition in this case amounts to 10 minutes. Initial analysis of the filesystem, including the recovery of deleted files, took 5 minutes. The selection of the relevant data took another 5 minutes. This results in an overall duration of 10:03 minutes for the selective imaging approach before initial results are available. The sector-wise approach took 14:00 minutes to reach similar results. The difference is illustrated in Figure 5.2. The winning margin for the selective imager in this case is 28.2 percent. Due to the same reasons valid for the hard-disk, this figure scales linearly with the amount of data.

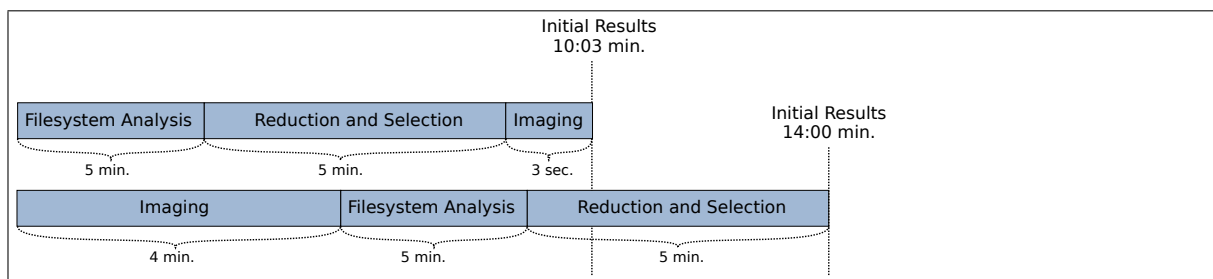


Figure 5.2: Imaging of a 4GB flash-drive, speed comparison

This margin even has significant potential for improvement, because the time spent on imaging compared to the time for analysis is very small. To compare this case for a larger data volume, the time data for the Seagate hard-disk is re-evaluated without the carving procedure. This results in an investigative duration of 36:17 minutes for selective imaging, compared to 60 minutes for sector-wise imaging, as illustrated in Figure 5.3. This implies, that investigators can deliver first results roughly 40 percent faster, when

employing selective imaging in a case with a large data volume and an uncomplicated recovery procedure.

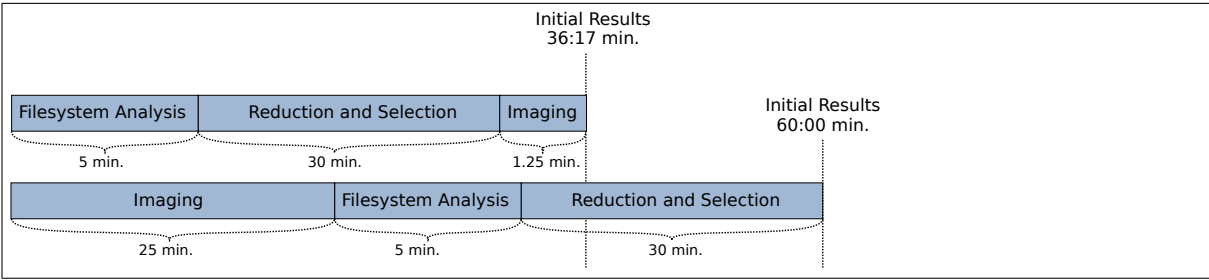


Figure 5.3: Imaging of a 20GB hard-disk without carving

5.2 Technical Details

Selective imaging is very different from sector-wise imaging in several technical details. These details have implications on the acquisition phase, that investigators need to be fully aware of, to benefit most from selective imaging. In this section we use the selective imager to evaluate the raw speed, correctness and disk wearing impact of this approach.

5.2.1 I/O Speed

The selective imager employs several techniques that hinder performance a bit, but are necessary to maximize disk space savings and verifiability of the images it creates. Some of these techniques are also used in other software, as they have benefits even for sector wise images. The two most notable ones are cryptographic hashing and compression of the image. Hashing is practically always performed, as it is the main metric of integrity verification for sector wise images. For partial images it is also mandatory, but performed on a per-file base not on the entire image. Compression is used to reduce the size of the image on disk. Modern formats for sector-wise images employ seekable compression to significantly reduce the size of images.

Table 5.1 compares the raw speed of several imaging tools. To minimize the performance impacts of the write-blocker and the usb-bus that connects the write-blocker to the forensic workstation, we created a raw disk image beforehand and used it as the image source for all tests. The blocksize was forced to 32768 in all tools, to eliminate speed-differences caused by different settings for some tools. The speed measurements of the different tools turned out very different, with the fastest tool achieving almost three times the speed of the slowest. However, this is not only due to differences in the implementation, but mostly because the tools also have a different feature set. When possible, we performed the test multiple times for each tool with different settings, to illustrate the performance impact.

Table 5.1: Imaging Speed by Tool and Features

Tool	Compression	MD5-Hashing	Speed (MB/s)
dd			39.00
dd_rescue			36.00
dcfldd			35.00
aimage			35.00
dff (raw)			32.28
dff (aff4)	•		27.03
dff (aff4)	•	•	26.62
dff (selective)	•	•	15.56
aimage	•	•	13.30

The highest copying speed is achieved by `dd`. This is mainly because of its simple feature set. Besides copying data, `dd` has no other features. The achieved speed of 39 MB/s can be regarded as the maximum I/O throughput, the test environment can provide.

The next three tools `dd_rescue`, `dcfldd` and `aimage` are closely grouped together, with about 90 percent of the speed of `dd`. The main reason for that are the extensive error checking features. These tools were developed specifically for the purpose of creating forensic images and emphasize error free copying and logging of irregularities, which naturally reduces performance a bit.

The raw copying implementation in DFF reaches roughly 83 percent of the maximum possible speed, as the imager is implemented in python and the data needs to be transferred between native objects that are implemented in c++ and the Python Runtime Environment.

The selective imager can also acquire a sector-wise image of the entire disk, when the selection is configured to do so. In Table ??, this is the DFF(AFF4) entry. It is slower than a raw acquisition with DFF, as the image is compressed and hashed during acquisition. The speed with compression and hashing enabled amounts to about 68 percent of the maximum speed. When run with the test image, the compression rate was about 65 percent. Hashing did not impact performance significantly, the slowdown compared to the run with compression enabled but hashing disabled is roughly 1.5 percent.

Operating the selective imager in its intended way has a significant impact on performance. This is most likely due to two reasons. One reason is the meta-data extraction. For each node that is acquired, the selective imager has to collect all meta-data and write it to the RDF-Graph of the node. More importantly, the order in which the data is read from the disk is not sequential. Nodes can be fragmented and the acquisition order not necessarily matches the order their data is stored on the device. Disk transfer speeds drop significantly when performing random access instead of sequential reads, because the disk head has to be repositioned frequently and can not read during this time [47]. The impact of this issue can be reduced by analyzing the location of the data

objects on the storage device and scheduling the acquisition in a way that minimizes the repositioning of the disk head. However, this requires a lot of effort and is beyond the scope of this thesis. It will be evaluated in future work.

The slowest speed was achieved by `aimage`, the reference imaging implementation of the aff3 format. While it uses the same feature set as the selective imager targeting an entire device, it only achieves half the I/O throughput. We have no explanation on why performance is that slow, as the raw copying speed of the tool was significantly faster than the DFF implementation.

In conclusion, the performance of the selective imager is sufficient, but not perfect. When comparing it to other tools, effects of the python implementation need to be eliminated, as they are not related to the general approach and can be mitigated with a native implementation. The programming language was chosen due to its quick development time and good readability, as the implementation is intended as reference to further development. Also, effects of compression and hashing need to be eliminated from the comparison, as these features are equally useful for sector-wise images and are not specific for selective imaging. The average speed impact on data copying of selective imaging can thus be assessed by comparing the selective imager acquiring an entire device with the same program acquiring each file on the device separately. The selective approach achieves 58 percent of the speed of the sector-wise approach. While this number seems relatively small, it implies that the overall copying time will be shorter than with the fastest sector-wise imager, as soon as the selected data volume is smaller than 58 percent of the capacity of a storage device.

### 5.2.2 Reliability

The most important property of any forensic acquisition tool is reliability. There can not be any doubt on either provenance authenticity or integrity of the created images. Because the provenance documentation for partial images is more complicated than for sector-wise images, special verification procedures need to be developed. Turner defines the *ultimate test* for any imager that does not generate a bit stream copy of a digital storage device as follows: »The method and storage container used must be able to store sufficient information about the provenance of the information capture such that when the information is restored it is identical to that which would have been acquired should a bit stream image have been taken« [63]. This means, the provenance documentation must enable examiners to extract data from objects in the image in the same order and to the exact same addresses, as they existed on the original device. Ultimately, this allows the construction of a sector-wise image from a partial image, if the partial image contains every data object on the device. While this is an interesting test, it is not very practical, as the main purpose of partial images is not to acquire every data object on a device. However, the demanded ability also implies examiners can extract data from the original device, at the addresses that are documented in the provenance key of a data object in the image. This data can then be compared to the stored data in the partial image. If both datasets match, the reliability of the acquisition method is verified. This is the exact procedure that is implemented in the verification program, introduced in

Section 4.5.2.

We have applied the verification program on every test image we acquired during the evaluation. There was not a single case in which the selective imager produced erroneous provenance information or data. In two cases we identified problems with filesystem parsers in DFF, which supply the selective imager with data. These problems were promptly fixed by the maintainers. After applying the fix, we repeated the acquisition of the problematic data, which was then verified correctly. This asserts the selective imager produced reliable data in all of the test cases. For the user, the verification of a partial image is just as easy as the verification of a sector-wise image. In both cases there is simple software available that accomplishes this task.

### 5.2.3 Disk Wearing

As mentioned in Section 3.1.3, we expect selective imaging to perform much better on damaged or old storage devices than sector-wise imaging. The idea is based on the assumption, that the device will only survive a limited amount of read operations or data transfer, as these actions wear out a hard-disk significantly faster than just being powered on. Selective imaging allows examiners to use the limited operations a damaged device has left in a directed way, to extract a maximum amount of evidence from the device before it dies. However, the pre-acquisition steps, providing examiners with the necessary knowledge to perform a useful selection, also use up some of those operations. In this section we evaluate the impact of typical procedures on the wearing of a storage device. Measurement of read operations is achieved through the block device statistics of the Linux kernel. For every block device `<dev>`, there exists a file located in `/sys/block/<dev>/stat`, which documents the amount of read operations that have been performed on the device, as well as the number of sectors read [59].

Table 5.2: Device Wear by Investigative Procedure (Kingston USB Device)

Procedure	Read Operations	Sectors Read	Relative Amount
Filesystem Analysis	202	4,528	0.06%
Selective Imaging	963	119,624	1.53%
Carving	61,048	7,827,392	100.00%
Sector-wise Imaging	61,122	7,827,392	100.00%

We performed the same forensic analysis of the test exhibits as in Section 5.1.2 and logged the number of read-requests to the disk as well as the number of sectors read for each step. The results for the USB-Flash-Drive are depicted in Table 5.2. The device had a size of 4GB and a total sector count of 7,827,392. The selection that was acquired with the selective imager was about 60 MB in size, which amounts to 1.5 percent of the total storage capacity. As the table shows, the parsing of filesystem data-structures required

202 read operations and transferred 0.06 percent of the device data. The evidence that was selected for acquisition based on this information required 963 operations and the transfer of 1.53 percent of the devices storage capacity. In comparison, the acquisition of a sector-wise image took more than 63 times the amount of read operations and the transfer of 100 percent of the data on the device. Because the filesystem on the Seagate hard-disk is empty, a similar comparison for this device does not make sense. Nevertheless, we also measured the amount of read sectors during carving for this device. The result is similar to the observations for the USB device, carving results in the transfer of all sectors on the device.

This data indicates, that selective imaging will significantly decrease the wear of the device during acquisition and thus allows for extensive evidence acquisition in cases where sector-wise images will most likely only capture a very small percentage of the relevant data. Nevertheless, there are some limitations that examiners need to be aware of to make use of this potential. Because advanced recovery techniques like file-carving rely on direct analysis of the contents of data objects, their application results in the transfer of large amounts of data. In the worst case, selective imaging will result in a significantly stronger wearing of the device than sector-wise imaging. When using file-carving techniques during recovery, the carving step alone resulted in the transfer of the same amount of read data as the creation of a complete sector-wise image. While some operations are cached by the operating system, most will have to be repeated during acquisition of the carved data. Examiners thus need to be aware on the impact their pre-acquisition techniques have on the device and choose each step wisely. If carving becomes necessary, the part of the device that the carver will be applied to should be selected completely for acquisition. Examiners can later extract it to a healthy device and perform the step here.

## 5.3 Practical Acceptance

In Section 3.1.3, we projected extensive benefits to examiners employing selective imaging procedures. To evaluate the practicability of these benefits, we performed an extensive evaluation of the approach with forensic partitioners. The main focus of this evaluation lies on the applicability and acceptance of selective imaging. The evaluation was performed in two ways:

- Live interviews with forensic examiners
- A questionnaire

To gain an extensive insight on the opinion of forensic practitioners, we performed several interviews with examiners. During these interviews we presented the concept and software that has been developed in the course of this thesis and asked for the experts opinion on its application in their everyday work. Two forensic examiners were also equipped with a Live CD containing an installation of our tool, to test it in their environment.

To obtain the opinion of a wider audience on some important questions, we developed a questionnaire with questions on acceptance and reliability of selective imaging. This questionnaire was then distributed to several forensic examiners.

### 5.3.1 Interviews with Forensic Examiners

We interviewed forensic practitioners from the private sector as well as government agencies. The law enforcements perspective was represented by two forensic examiners from Polizeipräsidium Südhessen in Germany. The private sectors perspective was represented by three forensic experts from PricewaterhouseCoopers, working with their Forensic Technology Solutions branch. The focus in the interviews was on the acceptance of selective imaging, the areas of application and the potential time- and cost-savings. We also tried to find out to what extend selection is already practiced, which tools are used and how the experts perceive the legal acceptance.

#### The Law Enforcement Perspective

The police examiners confirmed that there is a growing need for selection before acquisition. In fact, selective techniques are already employed in many cases. The problem with current procedures is, that they are not forensically sound. Often digital evidence from systems that can not be shutdown and dismantled is needed. Especially in cases involving white-collar crime, the data that examiners need to acquire often resides on systems that are vital to operations of the company. Taking down these systems would result in massive financial losses for the company. Therefore, traditional images can not be acquired.

Also, Network-Attached-Storage(NAS) devices have become very common. In many small companies, most of the data resides on consumer-grade NAS boxes attached to the companies network. These systems usually store data on some type of RAID. They often run custom embedded operating systems that are poorly documented and provide access to the contained data by means of web-services or windows-shares. Examiners have little to no chance to access the drives directly to create an image. Also, the reconstruction of RAID arrays is very complicated and expensive without the original hard- and software. Due to these problems, dismantling the device and directly imaging each drive is only rarely an option.

Due to these reasons police investigators sometimes have to rely on the companies network to be able to acquire any evidence at all. However, none of the existing forensic tools supports forensically sound acquisition of SMB- or CIFS-shares, which are the standard protocols to make data available on Windows based networks. This is why usually acquisition is done by mounting a network share and manually copying data using Windows Explorer or other file copying tools. These tools do not provide any details on the provenance of the acquired data, so it can only be documented by the testimony of people present at the acquisition.

Furthermore, access to the companies network with a forensic workstation is often not possible, because security mechanisms deny access to any unregistered devices. The

process to register the examiners workstation with the networks security mechanisms is usually cumbersome and takes a lot of time. Examiners thus have to use computers that are owned by the company for the acquisition process. This not only restricts the tools that can be used, but also introduces another untrusted layer between the actual evidence and the examiner.

Computers frequently store data from many different people and sources. Especially servers or networked storage systems often contain a lot of data, most of which is not relevant to the case. In many cases, investigators are not even legally authorized to acquire every bit of data on shared systems. Larger corporations usually have several attorneys that monitor searches very carefully. Especially in Germany there are very strict data protection laws. Acquisition of entire hard-disk images in these cases is impossible. Another group of cases involve searches at places, that are not owned by anyone involved in the case. These searches are covered by §103 StPO in Germany. Because the affected people and objects do not have anything to do with the case, the principle of commensurability has to be followed closely, to minimize the impact of the investigation on innocent people. Furthermore, some occupations in Germany like accountants, lawyers and journalists are protected by special laws against searches and seizure. The acquisition of data from a file server in an office shared by multiple lawyers is problematic at least.

There is a need for tools, that allow forensically sound acquisition of data even in these difficult conditions. Police examiners expressed a need for a selective imager with the ability to mount network shares and selectively acquire files from them, while also documenting and saving provenance information. They also expressed the need for a statically compiled binary that could accomplish the same from an existing (company owned) workstation.

In average investigations that only involve a small number of private computers, a necessity for selective imaging was not perceived. The current capacities still allow to archive compressed images and the acquisition process itself can run unsupervised over night. Many cases also require a very detailed analysis of computer systems. Selective imaging would be very difficult in these cases, because of the inherent risk to miss important pieces of evidence in the acquisition process. This evidence would most likely be lost forever because in many cases devices can not be reacquired since they have already been returned into the possession of the owner when the loss of specific data is detected. However, examiners felt that in the future the mere mass of data could change that. They expressed that investigators some day will probably have to dare the gap of missing some hidden evidence, to be able to get to any result at all.

Aside from the daily bulk of investigations, a small amount of cases exist where results are needed very fast. Police examiners feel that the duration of traditional imaging has a negative impact in these specific investigations. They reckoned that Selective Imaging could shorten the initial analysis process, delivering preliminary results faster and thus leading to a higher success rate.

Finally, some cases are opened because data that is perceived as proof of a crime is discovered by a third party. An example would be the ex-wife, that claims she saw pictures with possible child-pornographic content on her husbands computer. Usually

an initial assessment of this evidence by non-forensic personnel is conducted before a full scale investigation is launched. Even if the assessment yields that the data is harmless, it must be archived to disprove accusations if doubt arises. Many of these cases could be closed without having to involve the forensic experts, if software existed that allowed non-forensic personnel to selectively acquire the data in a forensically sound way. This would relieve forensic experts from insignificant work, allowing them to focus on work where an actual crime happened and their skills are required.

### The Private Sector Perspective

Commercial digital forensics deals with three different kinds of investigations:

- Computer Forensics
- eDiscovery
- Data Analysis

Computer Forensics projects usually involve the analysis of data contained on one or multiple computers. Usually the examiners image workstations and laptops at the client, then analyze the image in their own lab. Since the client has to pay for the time forensic examiners spend on site, anything that accelerates this process is greatly appreciated.

The examiners stated, that projects exist where the budget does not allow for the imaging of every possible hard-disk. It is therefore not unusual for them to perform selective imaging on disk level. Since the time that can be spent on the project is limited, one has to carefully select the hard-disks that can be imaged and analyzed. Those hard-disks still contain a lot of irrelevant data. It was estimated that a rough two percent of the acquired data is of use to the project. The remainder is personal data, system files and business data, that has nothing to do with the project. Since a selective imaging approach on file level allows for a more fine-grained selection, the forensic experts can utilize their capacities much better and thus reduce costs dramatically.

One criticism on the application of selective imaging in this context is, that in many computer forensics cases the acquisition phase is not repeatable. The forensic examiners often image the computer directly at the clients site, and have to return it to the user immediately after completion of the acquisition phase. If any relevant data is overlooked during selective acquisition, examiners can not return to the client and re-acquire these pieces of evidence. Even if this was possible, the user would have had plenty of time to get rid of any incriminating data. However, the examiners did not regard this as a severe problem. They stated that in most projects the concealment of evidence is not an issue. The data is usually in plain sight and even if it's encrypted, the it-department of the company usually has the keys. If there is the slightest indication that data might be hidden, examiners can still image very broadly. However, there are several kinds of data, like operating system files, known programs or anything in the National Software Reference Library [46], that never have to be imaged. This is because their content is known and a simple hash comparison can prove it was not modified by the user. Even

in these cases, the time savings will be significant. Examiners can simply image all user data, but discard operating system files and known software.

Similarly to the police investigators, private forensic examiners also often image selectively on file level using windows tools. Examiners often can not take down systems, that are vital to a companies business. If the investigation requires data from those computers, it has to be acquired by copying data over the network. Also, there are investigations where there are many systems involved and only a certain type of information is needed (for example office documents). In these cases files are selectively copied using standard copying tools, because the budget does not allow for complete imaging of all involved systems. The examiners reported they mostly use the tools `xcopy` and `robocopy` for this purpose. These tools are part of Microsoft's Windows Ressource Kit [43]. They maintain the timestamps of copied files, but do not record any provenance information, so the resulting copies are not forensically sound. Software that uses forensic methods to accomplish this kind of task would be gladly appreciated by private investigators, as it would finally allow them to account for provenance and integrity of evidence acquired this way.

E-Discovery projects mostly have a similar acquisition procedure. Usually a large amount of emails or office-documents have to be reviewed. This data is often acquired by using the operating system to copy it onto a USB-Drive or a CD-ROM. It is then processed (and usually indexed) in some way to enable reviewers to search and assess the data. Provenance information on the data is not needed by the reviewers, as they mostly analyze on application level. The exact position of the email-clients database on the disk is usually less interesting to them than the sent-date of the actual email inside. However, if doubt arises on the integrity or origin of the data, there is no way to prove it due to the non-existence of provenance information. The examiners expressed that a selective imager that does record provenance and integrity information would be a valuable addition to these kinds of projects, because it would strengthen the confidence in the data that is analyzed.

Data-Analysis projects are even more unclear when it comes to data provenance documentation. Often the data is supplied by the customer, so the methods by which the data is acquired remain undocumented. The people supplying the data rarely have any forensic experience, so the lion's share of digital evidence in these cases is acquired by someone at the customers company exporting data into an Excel-Sheet and burning it onto a CD-ROM. The data acquired this way is usually processed in some way to be loaded into a database, where the actual analysis takes place. Since the origin of the data can be described as dubious at best, results usually need to be backed by paper records. Data-Analysis projects therefore usually involve digital forensic analysts as well as business economists and accountants. The results of the forensic examiners are constantly checked and lined up with the existing paper records, to make sure they are verifiable. It is even possible that a project has to be discontinued, due to the lack of paper evidence. Even if incriminating evidence exists in digital form, results from analyzing it can not be used if no paper records exist that back the data.

The consultant affirmed, that a selective acquisition software that records provenance information and verifies data integrity would be of great use in any of these cases. Not

only could the headcount on the accountant side be significantly reduced by avoiding most of the paper-checks, the remaining business staff involved in the project could focus their competence on actual analysis which would result in a significant reduction in costs for personnel. However, the tool would need to be usable by laymen, so our reference implementation needs to be improved usability-wise to be applicable in these kind of projects.

The biggest concern in private industry regarding selective imaging was reported to be legal acceptance. Even though only a very small fraction of projects in the private sector end in court, corporations select the tools and methods they use very carefully. They want to make absolutely sure that if they do have to explain their methods in court, they are regarded as industry standard. The examiners stated, that the acceptance of a selective imaging tool in the private sector depends on the certification by official authorities (for example the US NIST), but is also influenced by the acceptance in court. If some reference cases emerge where a specific selective imaging tool was used for acquisition and courts accept this procedure, the probability of other companies adopting the new software will rise.

If a reliable selective imager existed, whose legal acceptance was certified or at least observable by legal practice, a broad usage through the whole private sector would most likely result. In an industry as cost driven as the consulting sector, savings in this magnitude are very unlikely to be ignored. When asked for an estimation on current efforts, the examiners believed the percentage of time that the imaging process takes in an average computer-forensics investigation to be about twenty percent. Since one does not know upfront where exactly on the drive this data is located, the memory savings resulting from selective imaging were estimated to about seventy percent. The amount of acquired data also influences the time the actual analysis takes, the overall time savings due to selective imaging in computer forensic projects therefore were estimated to be about thirty percent. Since personnel costs are the biggest factor in these types of investigations, the time savings are expected to have an almost linear impact on project cost.

The overall acceptance of selective imaging methods in court is estimated to be similar to the sector-wise approach. The experts believe that the acceptance of evidence in court is mainly dependent on the credibility of the examiner who performed the acquisition, as well as the documentation of the employed methods.

The examiners are very certain that selective methods will become increasingly important in the future. They believe, that due to the massive increase in storage capacity, future investigations will not be able to function with sector-wise images. Also, they observed that operating systems and technology change more and more in a way that hinders low-level forensic analysis. Solid State Disks for example do not allow direct access to their storage system, but provide a transparent abstraction of storage through their controller. This makes it very difficult for examiners to recover deleted data from unallocated sectors. In the future, forensic investigations are believed to focus on data that is readily accessible in the filesystem and recovery techniques will become less efficient. This development favors selective acquisition methods, as they are most efficient in cases with no or little recovery.

Nevertheless, the experts believe that the sector-wise approach is still the best acquisition technique in cases where a lot of difficult recovery has to be performed and data is expected to be hidden. They expect both methods to coexist in the future, selective acquisition being employed on conventional cases and sector-wise acquisition in difficult investigations.

### 5.3.2 Quantification of Examiner-Opinions

To gain an insight into the opinion of a larger audience of forensic examiners, we developed a questionnaire that covers similar issues as the ones discussed with examiners during the interviews. A blank copy of the questionnaire is available in Appendix A. We distributed the questionnaire to as many examiners as possible and received 17 filled out replies. These include six replies from examiners with German Bundespolizei, nine from examiners with Polizeipräsidium Südhessen and two from consultants with PricewaterhouseCoopers. The numbers presented in this section are based on an aggregation of individual results and thus represent averages, not individual opinions.

The questionnaire consists of 18 questions, that can be grouped into four different categories. The first group of questions is aimed at the need for selective imaging in typical forensic investigations. According to the received answers, the imaging process takes 19.41% of the time in an average investigation. 88.24% of the participating examiners stated, that the duration of the acquisition process is a problem in some cases. Figure 5.4(a) shows the different problematic cases, as well as how frequently they were mentioned. The biggest group of cases are those, where results are needed urgently. In cases of homicide or terrorism for example, time is often of the essence to prevent danger to the public. Another big group of cases where the imaging period is perceived negatively, are cases where the acquisition has to be performed on site. In Germany there are laws that limit the time employees are allowed to work per day, called the *Arbeitszeitgesetz* [10]. The duration of the acquisition is a problem, if it is longer than the time the performing examiner is allowed to work. Furthermore, cases exist where time or cost limitations affect the amount of storage devices that can be acquired.

The second group of questions evaluates the potential benefits that can be achieved with selective imaging. Examiners estimate the amount of data on storage devices that is relevant to the case to 13.13% in average investigations. Based on this assumption, they believe it is possible to save 41.25% of the overall time required for the investigation, when adopting a selective acquisition procedure. The savings in disk space, required to store the acquired evidence, were estimated to 55.63% with a selective acquisition procedure. Examiners believe it is possible to use selective imaging in 47.06% of all cases. They named many examples, the most frequently mentioned one is a case, where precise criteria on the kind of relevant data are known prior to the acquisition step. The next two big categories were cases involving the acquisition of data on corporate servers or cases where data is located on servers that are not physically accessible and have to be acquired over a network. Another group of cases that are suited for selective imaging are those, where the principle of commensurability has to be regarded very closely. This is the case when protected professions like attorneys or journalists are affected by the

### 5.3 Practical Acceptance

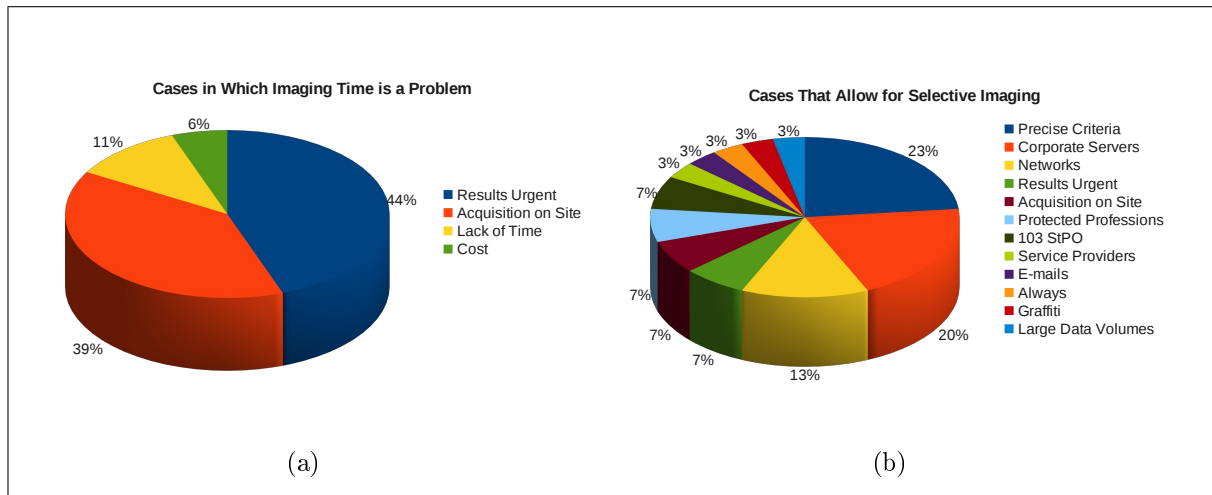


Figure 5.4

search. Also, cases involving §103 StPO are very sensitive in this matter. §103 StPO regulates searches at places, that are not owned by a suspect but need to be searched to apprehend him or to collect evidence on his whereabouts. Because the investigation affects innocent people in this case, investigators have to choose the data they acquire carefully, to minimize the impact on them. Data acquisition from service providers is also a sensitive matter, because these systems usually contain data from a lot of people that have nothing to do with the case. The principle of commensurability applies similar to cases involving §103 StPO. Many more cases were named, the detailed list and the frequency of mentioning is shown in Figure 5.4(b).

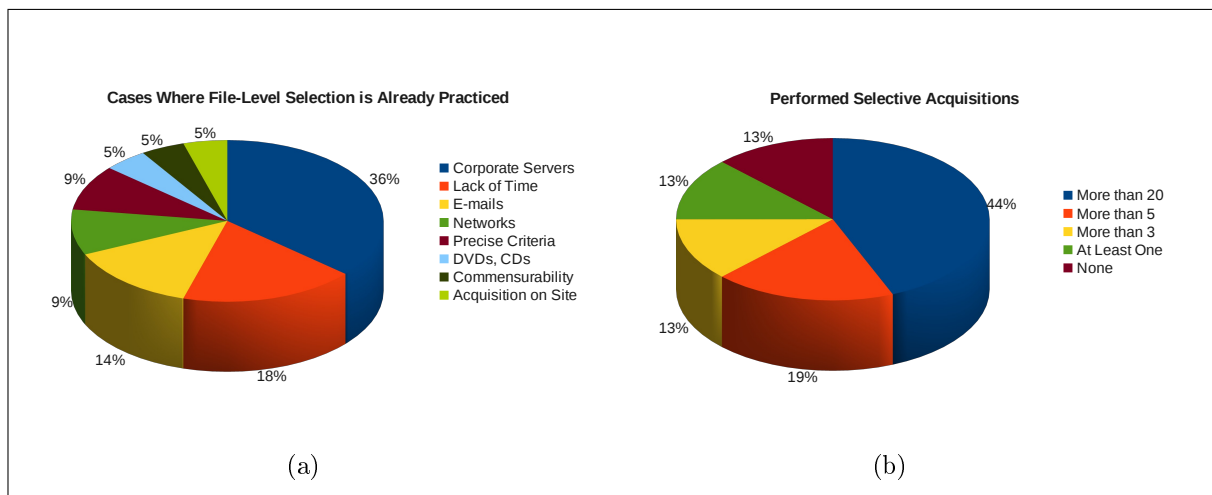


Figure 5.5

The third group of questions is focused on the methods that are currently used to handle situations where a sector-wise acquisition approach is not possible. The interviews and questionnaires showed, selective imaging is already performed by forensic examiners in some cases on file level. Forensic frameworks like Encase allow the extraction of so

### 5.3 Practical Acceptance

called *logical images*, which basically are partial images that limit the granularity of selection to file level. Often, even standard copying tools like robocopy [43] are used to selectively acquire files. The questionnaire showed that 82.35% of the questioned forensic examiners have witnessed a *logical* acquisition. An even higher fraction of 88.24% have at least once used copying tools to selectively acquire files. 58.82% of the examiners have witnessed that courts have accepted evidence acquired this way. 50% of the examiners that have performed a selective acquisition on file level have done this more than 20 times, only 14% have performed less than 3 selective acquisitions. The detailed distribution is shown in Figure 5.5(a). Various tools have been reported to be used for file level selective acquisition, a complete list is shown in Figure 5.6(a), only 53% of them have been developed for forensic purposes. Figure 5.5(b) shows the used tools with the frequency of their occurrence in the results. EnCase, FTK-Imager and X-Ways Forensic are all tools developed for the purpose of forensic acquisition. However, all other tools were developed for file copying and do not follow forensic principles like provenance documentation and protection of data integrity. Windows Explorer for example modifies the timestamps of copied files and thus can destroy important pieces of evidence.

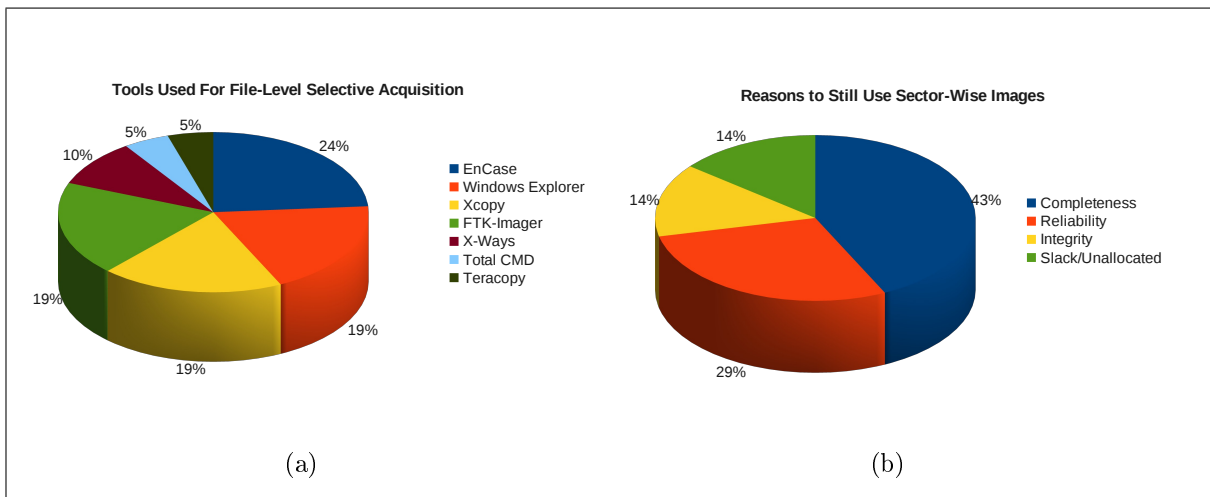


Figure 5.6

The fourth group of questions is aimed at finding out the confidence, examiners have in selective imaging. Even though the majority of forensic examiners practice selective acquisition at least in some cases, not all are confident the legal acceptance of partial images is sufficient. While everyone who filled out the questionnaire would be willing to employ selective imaging if it were legally recognized as a valid acquisition method, the overall confidence in the technique is rather low. On a scale from 0 to 10, where 0 means selectively acquired evidence would definitely be rejected in court and 10 means it would definitely be accepted, the examiners estimate an average acceptance of 5.47. However, the standard deviation in this case is 3.262, which indicates that opinions differ strongly from person to person. 47.06% of all examiners predict that sector-wise imaging will still be used in the future and is the only means of acquisition in some cases. The reasons given for this estimation are illustrated in Figure 5.6(b). The biggest

concern with partial images is completeness. Examiners are concerned to give up the all encompassing coverage of evidence that sector-wise images deliver. Also, reliability and integrity of the images is a big concern with partial images. Finally, examiners state that often evidence is found in file-slack or unallocated space. File level selective imaging can not acquire this data, which is why examiners feel a need for sector-wise images in cases where recovery has to be performed.

All things considered, the evaluation comes out in favor of selective imaging. Examiners agree there is a need to shorten the acquisition phase and believe selective imaging can lead to significant improvements of this factor. While many examiners doubt selective imaging can fully replace the complete, sector-wise approach, most of their arguments only apply to file level selection and are not an issue with a selective imager with arbitrary granularity as developed in course of this thesis. Our implementation allows to employ file-carving techniques directly on the device and selectively acquire the results, as well as the acquisition of entire regions of the device such as file-slack or unallocated space if necessary. Reliability and integrity of the acquired partial images are easily verifiable, it is just the underlying process that is a little bit more involved than the verification of a sector-wise image.

## 5.4 Summary

This chapter evaluated both the investigative approach to selective imaging as well as the software that was developed for this purpose. We conducted simulated investigations on two different exhibits with opposing characteristics in regard to the complexity of the necessary investigative procedures. The first exhibit was a used hard-disk acquired on ebay for a student forensics project. The previous owners have taken measures to erase any data contained on the device, so recovery procedures were extensive and complex. The pre-acquisition analysis of the device resulted in a dataset of 1178 Megabytes, which amount to 5.7 percent of its total capacity. The entire process, from the start of the pre-acquisition procedure until initial results became available, took 76:17 minutes. Acquisition of a sector-wise image took 25 minutes, which increased the time to obtain similar results with the sector-wise imaging approach to 10:00 minutes. The comparison of these figures results in a saving of 23.7 percent in time and 94.3 percent in disk space. Being very close to the worst conditions that are possible for selective imaging, this result can be seen as the minimal amount of savings that can be achieved by selective imaging. Savings in cases with better conditions will be significantly larger, we determined savings between 28 and 40 percent in time reductions and up to 99.6 percent in storage space savings.

When analyzing the absolute transfer speed, we determined a performance drop of 42 percent. This is mainly the result of the selective imager not acquiring disk blocks in a sequential way, which greatly impacts the transfer speed of block devices. Another source of the slowdown is the meta-data extraction, performed for each individual data object before transfer. While this is clearly a disadvantage, the impact is minimal in most situations because the amount of data that needs to be transferred by the selective

imager is smaller than with a sector-wise imager.

The practical acceptance of selective imaging is good. Forensic examiners already perform selective acquisition on file level and believe in the necessity of selective imaging in future investigations. File level selective acquisition is often performed with standard copying tools, which do not follow forensic principles, due to the lack of software that can perform this kind of procedure. The major commercial forensic frameworks have reacted to this problem and offer *logical imaging* functionality, which basically is selective imaging on file level. Because this technique can not acquire carved files or manually recovered data fragments, examiners still feel a need for sector-wise images. They believe, both techniques will coexist in the future, where selective imaging will be used on simple cases with large data volumes and sector-wise imaging will be used in cases where complex recovery or absolute data coverage is required.

## 6 Conclusion

In this chapter, we give a summary on the major points, presented in this thesis. Furthermore, we present some ideas on future work in this field, as well as improvements that are possible to the developed concept and software.

### 6.1 Summary

This thesis introduced an exemplary investigation model for digital forensics. The process of creating copies of digital evidence, called imaging, was explained and an overview on the commonly used tools and formats for this purpose was given.

Based on the investigative process, we developed a process model for the selective acquisition of digital evidence. The possible granularity of selection was examined and we concluded that it is necessary to be able to select data objects of any arbitrary granularity, for selective imaging to be able to completely replace sector-wise imaging. Because metadata also exists outside of logical data units such as files or partitions, it is important to explicitly acquire any metadata that is not stored inside the chosen level of granularity, as it would otherwise be lost. We analyzed the applicability of selective imaging in several different categories of investigations and determined that the approach can be employed in almost any type of investigation, but requires a supporting sector-wise image if the case requires investigators to return the original device to its owner, before the case is closed.

The partial images that are the result of a selective acquisition were defined as an aggregation of data objects from a digital device, together with all relevant metadata, that can be verified against the original at all times. To achieve the same level of legal reliability as sector-wise images, partial images require a combination of multiple provenance metrics, such as the block address of the contained data objects, as well as a verification metric such as a cryptographic hash. When these criteria are fulfilled, partial images are on par with sector-wise images from a legal perspective.

We developed multiple tools for selective imaging. A selective acquisition module and an import connector for partial images were developed as plug-ins for the forensic framework DFF. The imager is able to acquire any data object from DFF and extract multiple provenance documentation keys as well as all metadata that is stored in the framework on this object. The import connector recreates the logical structure of the evidence, as it existed before the acquisition procedure, and makes the data objects accessible in DFF together with all the metadata that was stored during acquisition. Finally, the partial image verifier was created, which is a program that can verify the provenance of a partial image, if the original device is connected to the computer it runs

on.

Finally, we evaluated the created software with two exemplary test cases. We determined improvements in the overall duration of the investigation between 23.7 and 40 percent, as well as storage space savings between 94.3 and 99.6 percent, even though the raw transfer speed of the selective imager only reached 42 percent of the fastest sector-wise implementation. The attitude of forensic practitioners towards selective imaging is mainly positive. Most of them agree that the acquisition phase in digital forensic investigation needs to be revised and selective imaging is an approach that can lead to significant improvements in this phase. However, many examiners do not believe that sector-wise imaging can ever be completely replaced by selective imaging, because they are used to file level selection which can not acquire data that needs to be recovered first.

## 6.2 Future Work

A drawback of the selective imager, which we observed during the evaluation, is the raw transfer speed. An intelligent scheduling algorithm can mitigate this drawback by re-ordering the read requests to be sequential in regard to their position on the storage device. This approach will require a little more upfront computation, but significantly increase speed, especially on hard disks.

Furthermore, the interviews with forensic practitioners revealed some acquisition scenarios, which are currently not fully supported by the selective imager. One scenario is the acquisition of evidence over a network. Examiners sometimes do not have direct access to a computer and have to copy evidence over standard network services like SMB or CIFS. These protocols do not provide any reliable provenance documentation, which weakens the legal reliability of images acquired from shared network volumes. Also network based acquisition does not allow any recovery techniques, because it limits access to the file level. A program that provides raw access to a systems storage devices can solve this dilemma, by enabling the selective imager to directly access the storage device and acquire accurate provenance documentation for the selected data objects. Also raw access to the device enables the selective imager to operate outside of the file level and perform advanced recovery operations such as file-carving.

Finally, forensic practitioners reported that selective acquisition can relieve them from a lot of insignificant work if the tools were usable by laymen. Many cases only require the assessment of files that are easily accessible. If for example a lawyer could operate the selective imager, these cases could be closed without involving a forensic examiner. To simplify the usage of the selective imager, the pre-acquisition analysis has to be automated as far as possible. Synergies with projects such as FiWalk [26] should be evaluated.

# A Questionnaire

The following 4 pages show a copy of the questionnaire, that was handed out to forensic examiners.

# Fragebogen zur Selektion vor der Sicherung

Die Kapazität von Festplatten wächst unaufhaltsam. Gleichzeitig nimmt die Menge der “irrelevanten” Daten die darauf gespeichert werden zu. Bei Kapazitäten von 1-2 TB machen sich die wenigsten die Mühe die Daten sorgfältig zu verwalten. Je nach Fokus einer Untersuchung sind heutzutage im Schnitt etwa 10% der gespeicherten Daten relevant. Private Urlaubsfotos und Videos interessieren Ermittler bei Untersuchungen zur Wirtschaftskriminalität in der Regel wenig.

Besonders bei Untersuchungen im gewerblichen Umfeld ist auch die Datenschutzfrage von großer Relevanz. Serversysteme speichern meistens Daten der unterschiedlichsten Quellen, wovon viele auch Personen zugeordnet sind die vom Durchsuchungsbeschluss nicht abgedeckt sind. In solchen Fällen wird meist, entgegen aller forensischen Prinzipien, eine simple Kopie der relevanten Daten mit Windows-Bordmitteln durchgeführt.

In ihrem Artikel “Selektion vor der Sicherung” [1] beschreiben M. Bäcker, F. Freiling und S. Schmitt Zeit- und Speicher-Schonende Methoden zur forensischen Sicherung von digitalen Speichermedien. Leitidee ist die Abkehr vom kompletten Image hin zur selektiven Sicherung genau der Daten, die für die Ermittlung relevant sind. Johannes Stüttgen entwickelt im Rahmen seiner Diplomarbeit ein Programm, mit dem eine forensisch korrekte Akquisition von Daten auch selektiv auf Dateiebene möglich ist.

Dieser Fragebogen dient der Abschätzung der Akzeptanz und Notwendigkeit selektiver Sicherung bei Forensik-Praktikern. Die Ergebnisse sollen zeigen ob ein Bedarf für Software in diesem Bereich besteht, sowie eine Einschätzung der Anerkennung von Beweisen die auf diese Weise erfasst wurden vor Gericht ermöglichen.

## 1. Wie groß ist ihrer Erfahrung nach der Zeitanteil, den das Imagen in einer durchschnittlichen Untersuchung beansprucht?

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%

## 2. Gibt es Fälle, in denen die Dauer des Imagens sich negativ auswirkt?

Ja \_\_\_\_\_ ... ☐

*Beispiele*

Nein ..... ☐

3. Wieviel Prozent der Daten auf untersuchten Datenträgern sind im Durchschnitt für eine Untersuchung relevant?

☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐  
 0%   10%   20%   30%   40%   50%   60%   70%   80%   90%   100%

4. Wie groß schätzen Sie die Zeit-Ersparnis in Prozent, zu der das Selektive Imagen in durchschnittlichen Untersuchungen führen könnte?

☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐  
 0%   10%   20%   30%   40%   50%   60%   70%   80%   90%   100%

5. Wie groß schätzen Sie die Speicher-Ersparnis in Prozent, zu der das Selektive Imagen in durchschnittlichen Untersuchungen führen könnte?

☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐  
 0%   10%   20%   30%   40%   50%   60%   70%   80%   90%   100%

6. Wie schätzen Sie die Akzeptanz partieller Images vor Gericht ein?

Bitte antworten Sie auf einer Skala von 0 bis 10 - wobei 0 bedeutet, dass Sie der Meinung sind dass das partielle Image vor Gericht als Beweismittel abgelehnt würde, und 10 bedeutet, dass Sie es für wahrscheinlich halten dass es zugelassen würde. Mit Werten dazwischen können Sie Ihre Meinung abstufen.

Schlecht Gut  
☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐  
 0   1   2   3   4   5   6   7   8   9   10

7. Unter welchen Umständen könnten Sie sich den Einsatz der Selektion vor der Sicherung vorstellen?

---



---



---



---

8. Wie groß ist der Anteil an Fällen in ihrer Arbeit, in denen eine Selektion vor der Sicherung theoretisch möglich wäre?

☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐   ☐  
0%   10%   20%   30%   40%   50%   60%   70%   80%   90%   100%

9. Um was für Arten von Untersuchung handelt es sich hierbei?

---

---

---

---

10. Falls die Selektion vor der Sicherung ein vor Gericht anerkanntes Verfahren wäre, könnten Sie sich vorstellen es einzusetzen?

Ja ..... ☐  
Nein ..... ☐

11. Haben Sie schon einmal erlebt dass im Rahmen einer Untersuchung kein komplettes Image erstellt wurde?

Ja ..... ☐  
*Aus welchem Grund?*  
Nein ..... ☐  
*Aus welchem Grund?*

12. Haben Sie schon einmal erlebt dass Beweise, die einzeln (also nicht als komplettes Image) erhoben wurden, vor Gericht Verwertung fanden?

Ja ..... ☐  
Nein ..... ☐

13. Haben Sie schon einmal selektiv Daten kopiert anstatt ein komplettes Image zu erstellen?

Ja ..... ☐  
*Aus welchem Grund?*  
Nein ..... ☐

14. Falls Sie schon einmal selektiv kopiert haben, welche Werkzeuge haben Sie eingesetzt?

Werkzeuge?

15. Falls Sie schon einmal selektiv kopiert haben, wie oft haben Sie dies getan?

1-2 Mal ..... ☐3-5 Mal ..... ☐5-20 Mar ..... ☐

Mehr als 20 Mal ..... ☐

16. Sind Sie der Meinung dass der Hauptteil der Forensischen Untersuchungen in Zukunft weiterhin mit herkömmlichen Images durchgeführt werden kann?

Ja \_\_\_\_\_ .. □

*Aus welchem Grund?*

Nein ..... ☐

17. Für wie wichtig halten Sie das Thema Selektion vor der Sicherung in der Zukunft?

Bitte antworten Sie auf einer Skala von 0 bis 10 - wobei 0 bedeutet, dass Sie der Meinung sind dass Selektion vor der Sicherung in Zukunft völlig unwichtig sein wird, und 10 bedeutet, dass Sie es für wahrscheinlich halten dass Forensiker in Zukunft ständig Selektiv Sichern. Mit Werten dazwischen können Sie Ihre Meinung abstufen.

Unwichtig

## Wichtig

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
0	1	2	3	4	5	6	7	8	9	10

18. Haben Sie irgendwelche Anmerkungen die bisher nicht berücksichtigt wurden? Falls Sie beispielsweise Erfahrung vor Gericht haben beschreiben Sie bitte in Stichworten wie umfangreich diese sind.

---

---

---

---

---

---

## B Source Code

A copy of the source code of all software developed in course of this thesis can be found on the accompanying CD-ROM labeled “Appendix” in the Directory `/code/`

## C Live DVD

The tools developed in course of this thesis have been integrated into a Live-DVD. The system contains a working copy of libAFF4, DFF 0.9, the selective imager, the aff4 connector and the partial image verifier. All tools are installed in the directory: `/home/demo/forensic/`. The partial image verifier is located in `/home/demo/forensic/dff/tools/`.

The login credentials for the system are:

<b>User:</b>	demo
<b>Password:</b>	aff4

## D Installation and Usage Manual

A copy of the usage and installation manual can be found on the accompanying CD-ROM labeled “Appendix” in the Directory `/manual/`

# Bibliography

- [1] Accessdata LLC. FTK v3.2 - The Forensic Toolkit. <http://accessdata.com/products/forensic-investigation/ftk>, 2010.
- [2] Air Force Office of Special Investigations. Foremost Filecarver. <http://foremost.sourceforge.net/>, 2006.
- [3] Arxsys. <http://www.arxsys.eu/>.
- [4] ArxSys. DFF Git Repository. <git://git.digital-forensic.org/dff.git>, 2011.
- [5] ASRData. Smart. <http://www.asrdata.com/forensic-software/our-software/>, Apr. 2002.
- [6] ASRData. Expert witness format. <http://web.archive.org/web/20031216052156/http://asrdata.com/BeSMART/whitepaper.html>, Apr. 2002.
- [7] D. Ayers. A second generation computer forensic analysis system. *digital investigation*, 6:S34–S42, 2009. ISSN 1742-2876.
- [8] F. Baguelin, S. Jacob, J. Mounier, and F. Percot. The Digital Forensics Framework. <http://digital-forensic.org/>, 2010.
- [9] D. Beckett and T. Berners-Lee. Turtle - Terse RDF Triple Language. <http://www.w3.org/TeamSubmission/turtle/>, 2008.
- [10] Bundesministerium der Justiz. Arbeitszeitgesetz. <http://www.gesetze-im-internet.de/arbzg/>, 2011.
- [11] M. Bäcker, F. Freiling, and S. Schmitt. Selektion vor der Sicherung. *Datenschutz und Datensicherheit*, 34(2):80–85, 2010.
- [12] W. Bär. *Handbuch zur EDV-Beweissicherung*. Boorberg, 2007. ISBN 3415038823.
- [13] B. Carrier. Autopsy forensic browser. <http://www.sleuthkit.org/autopsy/>, 2003.
- [14] B. Carrier. *File System Forensic Analysis*. Addison-Wesley, 2005. ISBN 0321268172.
- [15] B. Carrier. The Sleuth Kit. <http://sleuthkit.org/sleuthkit/>, 2003.
- [16] D. Cary. Endian FAQ . [http://david.carybros.com/html/ endian\\_faq.html](http://david.carybros.com/html/ endian_faq.html), July 2007.

- [17] E. Casey. *Digital evidence and computer crime: forensic science, computers and the Internet*. Academic Pr, 2004. ISBN 0121631044.
- [18] M. Cohen. LibAFF4. <http://code.google.com/p/aff4/>, 2009.
- [19] M. Cohen. Pyflag-an advanced network forensic framework. *digital investigation*, 5:S112–S120, 2008.
- [20] M. Cohen and B. Schatz. Hash based disk imaging using AFF4. *Digital Investigation*, 7:S121–S128, 2010. ISSN 1742-2876.
- [21] M. Cohen, S. Garfinkel, and B. Schatz. Extending the advanced forensic format to accommodate multiple data sources, logical evidence, arbitrary information and forensic workflow. *digital investigation*, 6:S57–S68, 2009.
- [22] G. Combs. Wireshark. <http://www.wireshark.org/lastmodified>, Dec. 2007.
- [23] P. Deutsch. RFC 1952 - GZIP file format specification version 4.3. <http://tools.ietf.org/html/rfc1952>, May 1996.
- [24] Deutsche Telekom. Data privacy protection guideline. <http://telekom.com/datenschutz>, Feb. 2011.
- [25] F. C. Freiling and B. Schwittay. A common process model for incident response and computer forensics. In *Proceedings of Conference on IT Incident Management and IT Forensics*, 2007.
- [26] S. Garfinkel. Automating disk forensic processing with sleuthkit, xml and python. In *Proceedings of the 2009 Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering*, pages 73–84. Citeseer, 2009.
- [27] S. Garfinkel. Digital forensics research: The next 10 years. *Digital Investigation*, 7:S64–S73, 2010. ISSN 1742-2876.
- [28] S. Garfinkel, D. Malan, K. Dubec, C. Stevens, and C. Pham. Advanced forensic format: an open extensible format for disk imaging. *Advances in Digital Forensics II*, pages 13–27, 2006.
- [29] K. Garloff. dd\_rescue. <http://www.garloff.de/kurt/linux/ddrescue/>, 2007.
- [30] P. Gauravaram and L. Knudsen. Cryptographic Hash Functions. *Handbook of Information and Communication Security*, pages 59–79, 2010.
- [31] M. Gercke and P. W. Brunst. *Praxishandbuch Internetstrafrecht*. W. Kohlhammer Verlag, Sept. 2009. ISBN 9783170191389.
- [32] A. Geschonneck. *Computer-Forensik*. dpunkt-Verl., 2006. ISBN 3898643794.

- [33] E. Grochowski and R. Halem. Technological impact of magnetic hard disk drives on storage systems. *IBM Systems Journal*, 42(2):338–346, 2010. ISSN 0018-8670.
- [34] Guidance Software. Encase forensics. <http://www.guidancesoftware.com/forensic.htm>.
- [35] Guidance Software. Encase Forensics v6. <http://www.guidancesoftware.com/forensic.htm>, 2008.
- [36] N. Harbour. dcfldd. <http://dcfldd.sourceforge.net/>, 2005.
- [37] Internet Engineering Task Force. RFC 2141 - Uniform Resource Name. <http://tools.ietf.org/html/rfc2141>, May 1997.
- [38] E. Kenneally and C. Brown. Revisiting Risk Sensitive Digital Evidence Collection. In *Proceedings of the 2005 DFRWS*, 2005. Available from [http://dfrws.org/2005/proceedings/keneally\\_risk.pdf](http://dfrws.org/2005/proceedings/keneally_risk.pdf).
- [39] E. Kenneally and C. Brown. Risk sensitive digital evidence collection. *Digital Investigation*, 2(2):101–119, 2005. ISSN 1742-2876.
- [40] D. Manson, A. Carlin, S. Ramos, A. Gyger, M. Kaufman, and J. Treichelt. Is the open way a better way? Digital forensics using open source tools. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, page 266b. IEEE, 2007.
- [41] J. Metz. The ewf file format. <http://sourceforge.net/projects/libewf/files/documentation/EWF%20file%20format/>, Jan. 2011.
- [42] L. Meyer-Goßner. *Strafprozessordnung: Gerichtsverfassungsgesetz, Nebengesetze und ergänzende Bestimmungen*. Beck Juristischer Verlag, May 2010. ISBN 3406606008.
- [43] Microsoft. Windows Server 2003 Ressource Kit. <http://www.microsoft.com/downloads/en/details.aspx?familyid=9d467a69-57ff-4ae7-96ee-b18c4790cffd&displaylang=en>, 2003.
- [44] Miniwatts Marketing Group. Internet Usage Statistics. <http://www.internetworldstats.com/stats.htm>, June 2010.
- [45] R. Morris. *Forensic handwriting identification: fundamental concepts and principles*. Academic press, 2000. ISBN 0125076401.
- [46] National Institute of Standards and Technology. National Software Reference Library. <http://www.nsrl.nist.gov/>, 2003.
- [47] S. Ng. Advances in disk technology: Performance issues. *Computer*, 31(5):75–81, 1998. ISSN 0018-9162.

- [48] G. Palmer. A road map for digital forensics research-report from the first Digital Forensics Research Workshop (DFRWS). *Utica, New York*, 2001.
- [49] D. Patterson. Latency lags bandwidth. *Communications of the ACM*, 47(10):71–75, 2004. ISSN 0001-0782.
- [50] S. Perumal. Digital forensic model based on Malaysian investigation process. *IJC-SNS*, 9(8):38, 2009.
- [51] Pew Research Center. Pew Global Attitudes Report. <http://pewglobal.org/reports/pdf/258.pdf>, 2007.
- [52] M. Pollitt. An ad hoc review of digital forensic models. In *Systematic Approaches to Digital Forensic Engineering, 2007. SADFE 2007. Second International Workshop on*, pages 43–54. IEEE, 2007. ISBN 0769528082.
- [53] Python Software Foundation. The Python Standard Library 14.1: hashlib. <http://docs.python.org/library/hashlib.html>, Sept. 2006.
- [54] G. Richard III and V. Roussev. Scalpel: A frugal, high performance file carver. In *Proceedings of the 2005 digital forensics research workshop (DFRWS 2005)*. Cite-seer, 2005.
- [55] G. Richard III and V. Roussev. Digital forensics tools: the next generation. *Digital crime and forensic science in cyberspace*, page 75, 2006.
- [56] G. Richard III and V. Roussev. Next-generation digital forensics. *Communications of the ACM*, 49(2):76–80, 2006. ISSN 0001-0782.
- [57] J. Stüttgen. Selective Imager Users Manual. [http://wiki.digital-forensic.org/index.php/Selective\\_Imaging](http://wiki.digital-forensic.org/index.php/Selective_Imaging), 2010.
- [58] The IEEE and The Open Group. IEEE Std 1003.1-2008: fcntl.h. [http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/fcntl.h.html#tag\\_13\\_11](http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/fcntl.h.html#tag_13_11), 2001.
- [59] The Linux Kernel Organization, Inc. The Linux Kernel Documentation. <http://kernel.org>, 2011.
- [60] B. Turnbull, R. Taylor, and B. Blundell. The anatomy of electronic evidence-Quantitative analysis of police e-crime data. In *2009 International Conference on Availability, Reliability and Security*, pages 143–149. IEEE, 2009.
- [61] P. Turner. Digital provenance-interpretation, verification and corroboration. *Digital Investigation*, 2(1):45–49, 2005. ISSN 1742-2876.
- [62] P. Turner. Unification of digital evidence from disparate sources (digital evidence bags). *Digital Investigation*, 2(3):223–228, 2005. ISSN 1742-2876.

- [63] P. Turner. Selective and intelligent imaging using digital evidence bags. *digital investigation*, 3:59–64, 2006. ISSN 1742-2876.
- [64] Volatile Systems. The Volatility Framework. <https://www.volatilesystems.com/default/volatility>, 2006.
- [65] X. Wang, D. Feng, X. Lai, and H. Yu. Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD. Technical report, Cryptology ePrint Archive, Report 2004/199, 2004.
- [66] X. Wang, Y. Yin, and H. Yu. Finding collisions in the full SHA-1. In *Advances in Cryptology–CRYPTO 2005*, pages 17–36. Springer, 2005.
- [67] D. White and M. Ogata. Identification of known files on computer systems. *National Institute for Standards and Technology, February*, 24, 2005.
- [68] World Wide Web Consortium. Ressource Description Framework. <http://www.w3.org/TR/rdf-concepts/>, Feb. 2004.
- [69] World Wide Web Consortium. XML Schema Document. <http://www.w3.org/TR/xmlschema-1/>, Oct. 2004.

All online ressources in this bibliography were accessible on March 30th, 2011.