

Master Research Project Random Sampling applied to Rapid Disk Analysis

Nicolas Canceill <Nicolas.Canceill@os3.nl>

Master System and Network Engineering University of Amsterdam, The Netherlands

July 20, 2013

Abstract

This document is a Research Project report for the Master education in *System and Network Engineering* at the University of Amsterdam, The Netherlands. The main focus of the study is on how random sampling can contribute to rapid forensics investigation of hard disk drives.

Forensically investigating several terabytes of data, bit per bit, requires large computing resources. Not only is it inherently long to extract all the data of a hard drive, but any automation tool aimed at parsing, organizing, and encoding all the data for human readability, also demands high amount of memory and computation capabilities.

The ever-increasing capacity of digital storage devices, along with new forensics and law-enforcement challenges, raise new concerns: time has become a critical investigation factor. This requires techniques of rapid analysis for digital storage devices, and also balances a trade-off between time and certainty.

This document presents a thorough modeling and evaluation of random sampling applied to data analysis on hard disk drives. The influence of most parameters in play are determined, based on general observations on modern digital storage mechanisms, and the precision and scalability of the technique are evaluated.

Instructor Prof. C. de Laat University of Amsterdam, The Netherlands

Supervisors Dr. E. van Eijk, Dr. Z. Geradts Netherlands Forensics Institute, The Hague, The Netherlands

Acknowledgements

I would like to thank my Instructor for this project, Prof. Cees de Laat, for his advice and concern. More generally, I am grateful to the entire team of the Master education in *System and Network Engineering*, from the University of Amsterdam, and particularly to Prof. Marcel Worring for his insight on the statistical aspect of the research.

I am also very thankful to the Nederlands Forensisch Instituut for offering me the opportunity of this Research Project. There I had the support of my Supervisors, Dr. Erwin van Eijk and Dr. Zeno Geradts, and I am very grateful for it: this work would not have been possible without them.

I would like to strongly thank Assoc. Prof. Simson Garfinkel, from the American Naval Postgraduate School. This research mostly based on his previous work in digital forensics; moreover, he kindly provided me with precious comments and enlightenments.

I would also like to thank James Taguchi, for accepting to share with me a draft on his Master's thesis, based on recommendation from S. Garfinkel. His work confirmed most of my results and goes even further, and his approach of the method helped me clarify and formalize my own study.

Finally, I would like to thank my classmates from the University of Amsterdam, for helping me review this report and offering their advice.

Nicolas Canceill

Contents

Abstract									
A	Acknowledgements								
\mathbf{N}	n content	5							
1	troductionISampling methods2Background3Research scope	5 6 7							
2	ethodology1Modeling2.1.1Simple scenario2.1.2More realistic assumptions2.1.3Complex scenario2Experimental process2.2.1Experimental data set2.2.2Examination process2.2.3Measurement details	8 8 9 11 13 13 14 14							
3 4	esults 1 Statistical distribution	 15 16 17 19 							
Δ	2 Contributions	20 20 21							
Li	List of Figures								
Li	List of Tables								
Li	List of Acronyms								
Bi	Bibliography								

1 Introduction

We know very little, and yet it is astonishing that we know so much, and still more astonishing that so little knowledge can give us so much power. Bertrand Russel

The necessity of rapid techniques to analyze digital storage devices raises as the storage capacity of the devices increase, and as their prices decrease. A typical 2010's 1TB desktop hard disk drive (HDD) costs less than 100 euros, and can sustain an average data transfer rate up to 150MB/s, so thoroughly inspecting 1TB takes at least two hours.

The proliferation of cheap storage devices, combined with increasingly-longer analysis time, has a drastic effect on the duration of forensics procedures. However, the circumstances of today's forensics investigations often pose critical timeframe challenges.

Traditional methods are growing out of time scale, so a way to rapidly identify high-value evidence could be extremely helpful to forensics examiners — e.g. law enforcement, intelligence, counter-terrorism... In a lot of real-world situations, quickly obtaining indications about suspicious data can assist in taking educated decisions: for scanning devices at checkpoints (airport security, border crossing), or for quickly classifying a large amount of devices.

The trade-off between certainty and time is a direct consequence of rapid analysis: as long as the disk is not fully read, there is still a possibility of error. As a result, the question becomes: can such a trade-off be efficiently balanced?

Even if full certainty is not achieved, it may still be possible to gather valuable information in a short time period. There are various way to identify valuable evidence: a suspiciously empty device, the presence of relevant target data, or unusual amounts of a specific data type, can become precious indications.

The current timeframe challenges require forensics examiners to perform such an identification from a small limited data sample. Hence that sample should represent the full data as accurately as possible: the measurements on the sample should be an estimate of the real value.

Of course, sampling comes with several considerations. Some information may be lost in the process; the results might not always be consistent; the efficiency could be variable — there could be worst-case scenarii. The adaptability, the scalability, the stability of the technique need to be questioned.

This research is founded on basic sampling theory and techniques; it draws inspiration from previous work on forensics analysis of digital evidence. After modeling the technique, the expression of meaningful values allows to determine the parameters in play. Then, experiments are conducted to study the influence of those parameters. Finally, the efficiency is evaluated in the context of forensics investigation.

1.1 Sampling methods

To keep a general scope, the forensics inspection of a HDD can be modeled as a *Space Filling Design* problem. Let a numerical experiment be a deterministic evaluation of the following type of model:

$$f : \begin{cases} \mathcal{S} \subseteq \mathbb{R}^d & \mapsto & \mathbb{R} \\ x & \mapsto & f(x) \end{cases}$$

Studying such an experiment means evaluating f. However, computing f(x) everywhere in S depending on all the d dimensions is often costly, which leads to the problem of representing f based on a limited number N of computations.

Space Filling Design is the process of representing the whole space S by a finished set of vectors:

$$\mathfrak{P}_n = \{x_1; \ldots; x_n\}, n \leq \mathfrak{N}$$

When S is countable, it is called a "population", and \mathfrak{P}_n is called a "sample". Achieving a *Space* Filling Design requires an adequate sampling method.

By essence, sampling only provides an estimate; increasing precision usually raises the costs. As a result, any sampling method has an inherent trade-off between costs and precision: for the forensics examiner, it becomes a trade-off between the time spent and the degree of certainty.

The quality of the estimate depends on the statistical sampling process. In the scope of a forensics investigation, it is very important to avoid any deterministic sampling process due to the risk of "safe havens": if all data is not equally likely to be sampled, then a malicious entity could abuse that knowledge to try to hide evidence.

Simple random sampling is a method such that every possible sample is equiprobable [1]. This is easily implemented by a uniform random selection without replacement: n items are drawn successively; at any draw, the process must give an equal chance of selection to any item not already drawn.

As a result, the probability of every possible sample is the same. Let N be the number of elements in S, then the number of possible samples is

$$\binom{N}{n} = \frac{N!}{(N-n)!n!}$$

Given a fixed set of n items, the probability to draw the first element is $\frac{n}{N}$. More generally, the probability of drawing a specific sample is:

$$\prod_{i=0}^{i=n-1} \left(\frac{n-i}{N-i}\right) = \frac{(N-n)!n!}{N!}$$

Therefore, every possible sample is indeed equiprobable. This model ensures that the sampling method does not create "safe havens".

1.2 Background

The main foundation of this research is the work of Simson Garfinkel, Associate Professor at the Naval Postgraduate School, USA. Garfinkel *et al.* have been investigating new methods of digital forensics, particularly techniques of rapid data analysis; they are conducting leading research on random sampling for data/feature carving [2, 3].

Garfinkel recently gave a talk entitled *Searching A Terabyte of Data in 10 minutes* [4], explaining how random sampling can speed up digital forensics procedures. He also presented his research at the Nederlands Forensisch Instituut (NFI). However, the exact process of rapid analysis is not entirely published.

For this project, only two open-source tools from Garfinkel's research were publicly available.

- frag_find is a hash-based carving tool: it looks for traces of a specific file in a target filesystem using block-wise Bloom filters [5], and was initially developed to carve the Reference Data Set (RDS) from the National Software Reference Library (NSRL) [6]. Besides hash-based filtering and some "intelligent search" mechanisms (gives a higher search score when finds consecutive blocks), it does not support random sampling.
- **bulk_extractor** is a generic forensics carving tool: it runs a broad search for various types of signatures (system and network configuration files, emails and mailboxes, ELF files, archives, specific Windows formats...). It supports basic block-wise random sampling: a global sampling fraction, and the number of sampling passes.

The latter is mainly a large library of signatures, using several searching techniques to detect and extract features of a known type. The former is a simple search tool, which is very suitable for a strict experimental process. In the course of this project, "frag_find" is used as a working basis for implementing and evaluating random sampling techniques.

Through communication with Garfinkel in the course of this project, he offered enlightening comments on his work. Moreover, his MSc. student James Taguchi accepted to share his own master thesis on the subject [7]; unfortunately, it was received a few days before the end of the project, but it provided very useful insight, and confirmed most of the results presented in this report.

1.3 Research scope

The main goal of this project was to answer the following research question:

How can random sampling help forensically investigate hard disk drives?

The problematics can be thereafter divided into several sub-questions:

- What information about a HDD's content can be characterized by random sampling?
- Which parameters have influence on the speed or precision of random sampling? How much impact can they have?
- What indications may such information provide for further investigation?

The focus of this research is the work of Garfinkel *et al.* because it is one of the most advanced results concerning fast HDD analysis. Considering the importance of ethical implications in forensics, the above questions are very legitimate. The objective is to analyze and evaluate the method, in order to clearly determine its power, and its limits.

The ideal outcome of research on random sampling of HDD would be a finished forensics tool: based on a HDD's characteristics, examiners could automatically get the lowest possible cost for a specified precision, or vice-versa. However, the timespan of this project does not allow to build a fully functional tool. The scope of this research is to make the first step: in order to bring down the problem to only time and certainty, all other parameters must be analyzed, and efficiency must be precisely evaluated.

2 Methodology

Sampling theory is a prolific branch of statistics. In a world of big numbers and surveys, it proves extremely helpful. This is why the beginning of the methodology is dedicated to modeling the problematics.

There is always a gap between academic studies and real-world situations, and the following is an attempt to make a compromise. By modeling the search as an elaborated urn problem, it was possible to design a simple experimental process based on Garfinkel's "frag_find" tool. Through this process, the influence of sampling parameters and data characteristics are evaluated.

2.1 Modeling

This section presents gradually-complex models of data carving through sampling. The general situation is the search for *target data* on a HDD. The question is not how much of the target data is present, but simply wether any of the target data is present.

The goal of this theoretical study is to determine the variables in play. This will allow to explore their meaning and evaluate their influence.

2.1.1 Simple scenario

In the simplest case, random sampling can be used to determine if a HDD has been properly wiped. Such an experiment can be modeled easily:

Firstly, randomly select a sample — divide the disk into chunks, and select some of them. Then examine the sample: if, and only if, all chunks in the sample are empty, then the disk is declared empty.

This can be brought down to the classic "urn problem":

There is an urn containing N balls of two colors: K red and N - K blue. If n balls are drawn without replacement, what is the probability that k of them are red?

The answer is called hypergeometric probability distribution:

$$P(k,n) = \frac{\binom{K}{k} \times \binom{N-K}{n-k}}{\binom{N}{n}}$$

Back in the context of HDD analysis, the meaningful result is the probability of error: when there is data on the disk, but the sample only contains empty chunks. Assuming all chunks are equiprobable, and neither replaced nor drawn twice, then the probability of error is:

$$P(0,n) = \frac{\binom{K}{0} \times \binom{N-K}{n}}{\binom{N}{n}}$$
$$= \frac{(N-K)!}{(N-K-n)!} \times \frac{(N-n)!}{N!}$$
$$= \prod_{i=0}^{i=n-1} \left(\frac{N-K-i}{N-i}\right)$$

This model actually scales extremely well: as shown on Table 2.1, by dividing a HDD into two billion chunks, with 10,000 improperly wiped (that is 0.001%), and sampling 0.05% of the drive, there is only 0.673% chance of missing the data.

	Probability of error	Probability of error
Sampled chunks	with $10,000$ chunks of data	with $1,000,000$ chunks of data
1	99.999%	99.950%
100	99.950%	95.123%
1,000	99.501%	60.645%
10,000	95.123%	00.673%
100,000	60.652%	
200,000	36.786%	
500,000	08.206%	
750,000	02.350%	
1,000,000	00.673%	

Table 2.1: Probability of not finding data among 2,000,000,000 chunks

Although the basic example is with empty chunks, this techniques is well suited when combined with any mean of identifying distinct unique chunks of data. It has been proven that using cryptographic hashes of individual chunks could allow to identify target data [8]. So, instead of distinguishing between empty and non-empty chunks, it could be possible to test wether chunks contain target data or not.

2.1.2 More realistic assumptions

The previous reasoning is confusing several parameters as "chunks". The size of chunks written by a HDD, of chunks from target data, of chunks sampled, need not be the same. Those three parameters are the size of the following data units:

- Sector The smallest data unit addressed by a HDD. Historically the sector size was 512 bytes, but now more and more HDDs use 4096 bytes.
- **Block** A piece of target data, organized in contiguous sectors, and hashed for comparison. Any smaller data chunk is considered to small to be identified.
- **Transaction** Contiguous sectors read from disk, and searched for target blocks: so it should be as large as a block, or more.

This is necessary due to multiple assumptions on how data is written on a HDD. Firstly, there are various way to write files, which could be spread over several sectors. Secondly, files are split in chunks of several contiguous sectors, but with an offset: files chunks may not always be block-aligned, only sector-aligned.

The first assumption is very reasonable: files are usually larger than 4096 bytes. In order to properly identify target data, a big enough part of the file should be read: using a small block size means that target data does not have to be spread over a large number of contiguous sectors; however, it also gives less information for identifying distinct files.

The second assumption breaks the conditions of the classical "urn problem". If the transaction size and the block size are equal, sampling is done per block: as a result, only block-aligned blocks can be found; otherwise the transaction just contains a part of a target block, which is not enough for identifying. So there may be blind spots: blocks that can never be found, independently of the sample selection.



Figure 2.1: The risk of blind spots: a target block is not found

Figure 2.1 illustrates this: target blocks are shown in grey, the first one is aligned with the transactions, the other is not. As a result, even if all transactions are sampled, the second block will not be found: it is in a blind spot. There are several ways to solve this.

The naive technique would be to consider every possible sector offset while keeping transaction size equal to block size: if the sampling process allows transactions to be selected at any sector offset, there are no more blind spots. Nevertheless, this creates new issues: either sampling becomes redundant, or an elaborated algorithm is required for proper sampling. In both cases, this really makes the model more complex, and the error rate harder to compute.

However, the transaction size can be larger than the block size. In that case, when a transaction is selected for sampling, all possible sector offsets within that transaction are checked, as shown on Figure 2.2: within a transaction, all possible blocks are compared to target data, and the target block would be found at the highlighted position.

As a result, the randomness and the non-redundancy of the sampling process are preserved, and chances of blocks being hidden are lowered. For instance, Figure 2.3 shows that if the transaction size is twice the block size, then the probability of a blind spot is reduced by half.

In order to completely void those chances, and to bring back the model closer to the "urn problem", the only solution is: not to allow any block to be caught between two contiguous transactions. A simple way to achieve that is to overlap contiguous transactions by a few sectors, just enough so that all blind spots are covered.



Figure 2.2: The benefits of a large transaction size on block alignment



Figure 2.3: The benefits of a large transaction on blind spots

2.1.3 Complex scenario

The presented assumptions can be formalized in a broader model, based on the initial scenario.

The parameters of the simple model take a new meaning, because there are now three classes of "chunks". N is the total amount of possibilities: the number of all possible transactions; and K is the number of transactions containing identifiable target data. Finally, n is the number of transactions.

Then the model can be completed by using the following parameters, as shown on Figure 2.4:

 ${\cal S}\,$ the sector size

B = bS the block size

T = tS the transaction size



Figure 2.4: The parameters of the model

Z = zS the image size

 $D=dS\,$ the amount of target data

So K can be expressed as:

$$K = \left\lceil \frac{D}{T} \right\rceil = \left\lceil \frac{d}{t} \right\rceil$$



Figure 2.5: Overlapping transactions avoids blind spots

In order properly cover blind spots, transactions need to intersect each other by almost B. Overlapping by a full block would create redundancy; the proper size is B - S because the size of the blind spot is 2(B - S), as shown on Figure 2.5. Thus N can be expressed as:

$$N = \left\lceil \frac{Z}{T - (B - S)} \right\rceil = \left\lceil \frac{z}{t - (b - 1)} \right\rceil$$

Injecting those expressions in the previous model results in the new probability of error:

$$\prod_{i=0}^{i=n-1} \left(\frac{\left\lceil \frac{z}{t-(b-1)} \right\rceil - \left\lceil \frac{d}{t} \right\rceil - i}{\left\lceil \frac{z}{t-(b-1)} \right\rceil - i} \right)$$

The model is consequently independent of the sector size S. The variable z depends on the situation, and d is unknown. Whereas t, b and n are the sampling parameters, and can be tuned depending on external requirements.

2.2 Experimental process

As aforementioned, the experimental process is based on Garfinkel's "frag_find" program. An automated chain of tools was built to perform the measurements.

For every experiment, the process is the same:

- 1. Generate experimental data
- 2. Run the search process
- 3. Gather the results

Each measurement is averaged from successive experiments.

2.2.1 Experimental data set

The scope of this research was necessarily limited due to the timeframe: experiments were conducted on an average HDD based on a simple data set. In order to repeat and vary experiments, an automated data generation tool was designed.

The National Institute of Standards and Technology is an American agency dedicated to science. They support the NSRL project:

The NSRL is designed to collect software from various sources and incorporate file profiles computed from this software into a Reference Data Set RDS of information. The RDS can be used by law enforcement, government, and industry organizations to review files on a computer by matching file profiles in the RDS.

Based on that data set, it was possible to design a simple process to obtain data of a target type. Since the data set is composed of hashes, the target type is a range of hashes. Then, block of data can be randomly generated to match the specified range.

That block generator was incorporated into a simple script to create experimental disk images. The tool supports various parameters:

- Image size
- Sector size
- Block size
- Source of target blocks
- Amount of target blocks

- Source of non-target data
- Proportion of empty sectors

The different layouts (empty sectors, non-target data, target blocks) are uniformly random. Firstly, the image is fully written with non-target data, along with the appropriate proportion of empty sectors; data is written sector by sector in a random order. Then the specified amount of target blocks is inserted at random positions on the image. Finally, some information is logged — positions of target blocks, number of empty sectors...

2.2.2 Examination process

The examination process is adapted from the "frag_find" tool; the main addition to the original implementation is the sampling selection process. Sampling selection is performed beforehand, and can be tuned by adjusting the sampling fraction.

For the purpose of the experiments, the tool supports several parameters:

- The image to analyze
- The target data to look for
- The sector size
- The block size
- The transaction size
- The sampling fraction

The search is performed in several steps as follows:

- 1. Read the size of the image
- 2. Enumerate the total number of possible transactions
- 3. List transactions for the sample
- 4. Go through the list and perform every transaction

For each transaction, all possible sector offsets are checked: the block starting at each offset is hashed, and compared with target blocks.

2.2.3 Measurement details

Two durations were measured: the full execution time, and the duration of the search itself. Results are simply logged on-the-fly, at the end of each execution. Afterwards, they are formatted through a set of parsing patterns through AWK scripting. This allows to compute averages and rates directly from the script.

The search duration is measured through the standard C library sys/time.h, and the full execution time is measured through shell built-in time, both directed at the system clock.

Measurements were conducted on a standard SATA 7,200 rpm HDD. Data was systematically wiped and generated again for every experiment.

3 Results

Measurements mentioned as "average" are drawn from 60 successive independent experiments, run on unique automatically-generated images with the specified parameters.

3.1 Statistical distribution

The general shape of the results for a series of experiments is shown on Figure 3.1. The measures converge towards the real value of the fraction of target data:

$$f_{\rm real} = \frac{d}{z} = 0.203$$

The experiments were run with f, t and b fixed, and various values of n and z. The results present some characteristics of an unbiased normal estimate.



Figure 3.1: Graph: Distribution of results for a series of experiments

Moreover, the experimental process seems to behave similarly to a normal estimate. It is possible to observe the effect of "distribution walls": values impossible to go beyond. For instance, the fraction of target data is obviously bounded: $f \in [0, 1]$. Consequently, 0 and 1 act like "distribution

walls": a normal distribution is limited by the standard error σ ; however, when the average is less than σ away from a wall, a part of the normal distribution is impossible.

As a result, when a normal distribution is "close to a wall", the standard error decreases. Such a behavior is shown on Figure 3.2. The experiments were run with t and b fixed, various values of n and z, and three distinct values of f_{real} : one far from the walls, and the two others close to the walls. The results present a lower error when close to a wall.



Figure 3.2: Graph: The effect of "distribution walls"

3.2 Precision scaling

The envelope of the distribution on Figure 3.1 is the general shape of the measured standard error rate s(n). That shape is preserved with the average error rate $\bar{s}(n)$, and can be estimated as an inverted square root.

Figure 3.3 demonstrates that behavior:

$$\bar{s}(n) \sim \frac{1}{\sqrt{n}}$$

The experiments were run with t and b fixed, various values of n and z, and averaged with several values of d.

Using a transaction size larger than b can speed up the search process: for a fixed fraction $\phi = \frac{nt}{z}$ of the image being sampled, numerous small transactions will generally be slower than fewer long transactions. However, increasing t also augments the error rate — for instance, doubling t roughly



Figure 3.3: Graph: Influence of the number of transactions

halves the total number N of transactions, and thus requires a higher ϕ to achieve the same error rate.

Such behavior is shown on Figure 3.4. The experiments were run with d and b fixed, and various values of ϕ and t.



Figure 3.4: Graph: Influence of the transaction size

3.3 Time scaling

Aside from the error rate, the most meaningful variable is the search duration. As a reasonable assumption, the disk seeking time should be averaged over multiple experiments, and the search duration should be proportional to the sample size — thus proportional to n for a fixed transaction size.

Indeed, such a behavior is observed: Figure 3.5 shows that the search time scales proportionally to the sample size. The experiments were run with t and b fixed, various values of n and z, and averaged with several values of d.



Figure 3.5: Graph: Search duration scaling

For small values of n, an inflection of the curve can be observed. A detail of that phenomenon is shown on Figure 3.6. The experiments were run with t and b fixed, various values of n and z, and averaged with several values of d.



Figure 3.6: Graph: Search duration for small samples

4 Conclusions

4.1 Results summary

The general shape of the results, shown on Figure 3.1, presents most characteristics of the *simple* random sampling technique: an unbiased estimate of a normal distribution. The measure is taken as a simple average f over the sample, as the ratio between the number of successful transactions and the total number of possible transactions.

For a specified HDD, the variance V of the data distribution is fixed. The theoretical standard error for simple random sampling is:

$$\sigma(n) = \frac{V}{\sqrt{n}}$$

This is very close to the statistical model $\bar{s}(n)$ observed, so it is coherent with the results. However, it was mentioned that the context of HDD analysis introduces additional constraints on the model.

Based on that elaborated model, the relevant parameters of the experiments are determined. Table 4.1 presents a summary of those parameters and their meaning.

External parameters				
Capacity Z of HDD	Easily determined			
Sector size S	Easily determined			
Amount of target data D	Unknown			
Sampling parameters				
Block size B	Depends on data layout, affects data identification			
Transaction size T	Depends on data layout, affects search duration			
Amount of transactions n	Depends on required precision, affects search duration			

Table 4.1: Parameters of the experiments

The influence of the various sampling parameters is analyzed, and summarized thereafter.

- The block size depends on the data layout, and determines the identification efficiency: small blocks will not require target data to be in numerous contiguous sectors, but large blocks will provide more data for identification efficiency.
- The transaction size depends on data spreading along the HDD, and have influence on the search duration: a larger transaction size means less total transactions, so it requires a larger sample to achieve the same error rate, but large transactions can usually be read faster.
- The sample size has direct influence on the search duration and the error rate: the search time is proportional to the sample size, but the error rate is inversely proportional to its square root.

All of this provides a framework of indications to adapt to external parameters, and tune the sampling process.

4.2 Contributions

This Research Project focuses on the study of random sampling in order to forensically investigate hard drives. In this report, the investigation method is modeled and evaluated.

The problematics can now be answered:

• What information about a HDD's content can be characterized by random sampling?

The presence or absence of specific blocks of target data.

• Which parameters have influence on the speed or precision of random sampling? How much impact can they have?

There are two kinds of variables involved: the sampling parameters, and the characteristics of the HDD — capacity, sectorization, data layout.

Depending on the context, they may prove to scale extremely well in precision and time. In order to properly tune the sampling process, it may be critical to obtain some preliminary information about the target evidence.

• What indications may such information provide for further investigation? The simple scope of searching for specific target data can provide very valuable indications. The tremendous speed of analysis to obtain a specified error rate on the presence of data could prove extremely useful: for classifying evidence, prioritizing investigations... The value of quickly-retrieved indications about digital evidence has already been proven [9].

Random sampling an array of HDDs can give indications about which one to study first, at which addresses, and provide insight about target data.

Finally the original research question can be addressed:

How can random sampling help forensically investigate hard disk drives?

As a general conclusion, random sampling has been proven to be a powerful, scalable, adaptive technique for rapid HDD analysis. Based on a limited insight on the target evidence, it is possible to search the HDD very efficiently. Suitable sampling parameters can achieve excellent results, and allow to adapt to various situations.

4.3 Further research

As mentioned in the Introduction, the ideal outcome of research on random sampling for HDD analysis would be a fully functional tool. The model outlines multiple parameters, but what ultimately matters is the balance between the duration of the process, and the tolerance for error. This project aims at paving the way to an automated decision process, that could suggest sampling parameters based on the situation.

Such a process could then be integrated in a portable forensics toolkit: with a proper interface to a HDD, it would be able to estimate the search duration based on the target error rate, and vice versa. That would constitute a profitable asset for forensics investigators.

List of Figures

2.1	The risk of blind spots: a target block is not found	10
2.2	The benefits of a large transaction size on block alignment	11
2.3	The benefits of a large transaction on blind spots	11
2.4	The parameters of the model	12
2.5	Overlapping transactions avoids blind spots	12
0.1		
3.1	Graph: Distribution of results for a series of experiments	15
3.2	Graph: The effect of "distribution walls"	16
3.3	Graph: Influence of the number of transactions	17
3.4	Graph: Influence of the transaction size	17
3.5	Graph: Search duration scaling	18
3.6	Graph: Search duration for small samples	18

List of Tables

2.1	Probability of not finding data among 2,000,000,000 chunks	9
4.1	Parameters of the experiments	19

List of Acronyms

HDD hard disk driveNFI Nederlands Forensisch InstituutNSRL National Software Reference LibraryRDS Reference Data Set

Bibliography

- [1] William G. Cochran. "Simple Random Sampling". In: *Sampling Techniques*. Third Edition. John Wiley & Sons, 1977. Chap. 2.
- [2] Simson L. Garfinkel. "Fast Disk Analysis with Random Sampling". In: *CENIC 2010.* Corporation for Education Network Initiatives in California. 2010.
- [3] Simson L. Garfinkel. "Cross-Drive Analysis with bulk_extractor and CDA tool". In: OSDF 2012. Open Source Digital Forensics. 2012.
- [4] Simson L. Garfinkel. "Digital Forensics Innovation: Searching a Terabyte of Data in 10 Minutes". In: DCACM 2013. Association for Computing Machinery. Washington, DC, USA, 2013.
- [5] Burton H. Bloom. "Space/Time Trade-offs in Hash Coding with Allowable Errors". In: C. of the Association for Computing Machinery 13.7 (1970), pp. 422–426.
- [6] Simson L. Garfinkel, Paul Farrell, and Douglas White. "Practical Applications of Bloom Filters to the NIST RDS and Hard Drive Triage". In: ACSAC 2008. Computer Security Applications Conference. Anaheim, CA, USA, 2008.
- [7] James K. Taguchi. "Optimal Sector Sampling for Drive Triage". MA thesis. Naval Postgraduate School, 2013.
- [8] Simson L. Garfinkel et al. "Using purpose-built functions and block hashes to enable small block and sub-file forensics". In: *Science Direct* 7.S (2010), pp. 13–23.
- [9] Brian Jones, Syd Pleno, and Michael Wilkinson. "The use of random sampling in investigations involving child abuse material". In: *Digital Investigation* 9.S (2012), pp. 99–107.