

Effectiveness of File-Based Deduplication in Digital Forensics*

Sebastian Neuner, Martin Schmiedecker
and Edgar Weippl
SBA Research, Vienna

Email: {sneuner|mschmiedecker|eweippl}sba-research.org

Abstract

Over the last decades the increasing amount of storage became a pressing problem for forensic investigators. This is caused by the computerization of everyday life and the associated increasing number of different devices in typical households. Considering multi-terabyte storage on the suspects side, even more storage requirements emerge on the side of the investigator for secure backup and working copies.

In this paper we improve the standardized forensic process by proposing to rigorously use file deduplication across devices as well as file whitelisting in investigations in order to reduce the amount of data that needs to be stored for analysis as early as during data acquisition. These improvements happen in an automatic fashion and are completely transparent to the forensic investigator. They may furthermore be added without negative effects to the chain of custody or artefact validity in court, and are evaluated in a realistic use case.

Additionally, we illustrate the effectivity of our proposed approach on a real-world corpus by showing a notable reduction in number of reduced files as well as storage.

1 Introduction

Investigators in digital forensics are facing several issues throughout their daily analysis routines. One of the biggest issues is the increasing amount of commodity hardware affordable for everybody. Commodity 3.5" SATA hard drives come with a maximum capacity of up to 8 terabytes per hard drive, while memory cards for smartphones and digital cameras can have up to 256 gigabytes. USB thumb drives have a current maximum capacity of two terabytes. Observing the past trend, Kryder et al. [18] expect this trend to increase even further than

*This is the preprint of the paper to be published with Wiley's Security and Communication Networks, DOI: 10.1002/sec.1418. This paper is the extended version of a paper published at IWCC'2015 [23]

posited by Moore’s Law [5]. Considering the stated numbers and the average number of devices per household, a forensic investigator would have to deal with tens of terabytes of data. This amount of data has to be securely acquired, processed and stored in several copies. Garfinkel et al. [13] predicted this problem years ago. However, so far this issue was not tackled in-depth in research and neither was it adapted into the forensic process of investigators nor in the corresponding standards.

This extended paper not only aims to enhance well-known standards like the RFC 3227 [4] or the NIST SP-800-86 [16] with the recommendations proposed in this paper, but also aims to proof the proposed performance improvements by reducing the to-be-acquired data in an early stage of the investigation. Those recommendations specifically tackle storage capacity issues during an investigation and therefore also reduce the needed processing power, workload and time to handle the data.

Therefore the contributions of this paper are as follows:

- We propose an advanced forensic process for digital investigations, taking into account some of the most pressing limitations for investigators.
- We discuss different analysis techniques which scale well and can be used to limit backend storage requirements for analysts.
- We evaluate our process with an exemplary use case and show that the overall storage requirement for that case can be decreased by 78%.
- Conducting an evaluation on a real-world dataset, we show that storage reductions of over a third are possible.

The remainder of this paper is organized as follows: Section 2 gives a brief background on the forensic process as well as existing guidelines for data acquisition and processing. Section 3 explains our proposed changes and improvements of the forensic process. In Section 4 these improvements are first evaluated on an artificial dataset and then proved on a real-world corpus. We discuss the results and limitations in Section 5, before we conclude in Section 6.

2 Background

The usual image acquisition approach in digital forensics as defined in [16, 4] creates two images of a hard drive of interest, while hashing the source and destination multiple times to prove that the integrity of the source hard drive has not been tainted and the underlying data has not been modified. As such, the created images are exactly the same size as the hard drive, which is one of the yet unsolved problems in digital forensics [13]: hard drives with 8 terabytes capacity are nowadays a commodity, and can be readily obtained online and in retail stores.

The problem with the current forensic process as practiced today is its lengthiness. It can take days to acquire and analyze large data repositories. Even though highly parallelized approaches have been proposed recently [14], they are not yet incorporated in commercial tools or the process itself, where open source products are heading in the right direction as shown by the well-known tool *bulk_extractor*¹. Moreover, the large set of software, data formats and as devices restrict the possibility of having fully automated approaches [8]. As such, the images created are still bit-by-bit exact copies of the hard drive to be analyzed.

File whitelisting is a common approach in digital forensics, during which each file on the hard drive is hashed using e.g. SHA-1 and compared against a list of well-known, benign files. The largest corpus of benign files is the NIST National Software Reference Library (NSRL) with their Reference Data Set (RDS). To build it, NIST installs all kinds of software in virtual machines and monitors the files that are created during installation. This allows the RDS to map each stored hash value to a specific file and furthermore to which software package containing it. The RDS is published quarterly, the most recent version at the time of writing being the RDS 2.49 from June 2015, containing more than 42 million unique hash values.

3 Improvements to the Forensic Process

This section is splitted into two parts: In the first explain theoretical techniques which can be leveraged to cope with the ever increasing case sizes. We put them in the context of an artificial case in the second part, especially where to apply them with respect to the existing recommendations.

3.1 Individual Improvements

The first proposed enhancement of the forensic process is already done in practice: **not always are two physical copies necessary**. While a working copy and at least one backup copy should always be in place, less stringent rule enforcement is needed when the data source drives are not bound to time requirements to get back to their owners or into production. This is particularly the case for investigations by law enforcement, where the data sources themselves are confiscated and no temporal pressure exists to return them to the owner. We do not have concrete numbers on how this is done in practice, but this can be very effective for reducing storage requirements. It is more like a logical enhancement to the standard processes, since it is not in all cases that the data needs to go back to production systems as soon as possible. Of course, for production systems where downtime is an issue and hard constraints exist that these systems stay online, a second copy is needed for backup. This is of

¹Online at https://github.com/simsong/bulk_extractor

relevance for e.g. all kinds of server systems like e-mail or web servers. Sometimes it can be also enough to create an image of the current files in the file system, omitting the free space and possible file slack. This depends on the context of the investigation, and the actual questions to be answered.

Another strategy which is missing so far in the process descriptions is the rigorous use of **file whitelisting**. Files irrelevant to the investigation can be easily excluded in the early stages due to the use of cryptographic hash functions like MD5 or SHA-1, whereas files of particular interest can be identified if they are known a-priori to the investigator. In the forensic community, the most notable example for the former case is the NIST national software reference library (NSRL) with their reference data set (RDS) [21]. It uses default software installations of operating systems and end user software to derive a list of hash values on a file basis. The most recent version of the RDS 2.49 (as of October 2015) contains more than 42 million hash values, for over 150 million files. An example for the latter is PhotoDNA which computes a visual fingerprint for pictures and compares it with known pictures of child abuse. It was developed by Microsoft and Dartmouth University and is used by large software companies like Facebook or Twitter. Most recently, a REST API was introduced to query the PhotoDNA database online². Due to the availability of cheap storage and processing power, we argue that any investigator could and should set up their own list of hash values for files of interest. This could include all files from intra-company file shares, possibly malicious files from anti-virus quarantine, web pages (including pictures and thumbnails) or company-wide e-mail attachments. Depending on the local privacy laws there are hardly any limitations on which files to include.

The improvement on the storage backend which this paper proposes is the creation of a **reduced working copy**. It is created as soon as all known, benign files are identified, as they can be safely excluded from the need to store them (except for their metadata). All other files are stored according to the file system metadata, and additionally all portions of the free space are extracted and stored as well. At worst, this can be a very large fraction of the original drive capacity. At best, a vast majority of files can be excluded in a fully automated process and without any interaction of the investigator. Since this process is strictly monotonous (the resulting working copy can only be at most the capacity of the drive), the resulting working copy will always be smaller than the full capacity of the storage drive. All further analysis steps can be done on this reduced working copy, and the original drive(s) can be locked securely away as the backup. If the drive(s) need(s) to go back into production use, a second copy is to be created using a bitwise copy. The second large improvement on the storage backend is the rigorous use of deduplication, at the very least across devices within each case. This step should also include the application of **fuzzy hashing** [26], since files which are similar but not the same until the very last

²Online at <http://www.microsoft.com/en-us/photodna>

bit cannot be identified using cryptographic hash functions. While the most commonly found fuzzy hash functions are *ssdeep* [17] and *sdfhash* [25], there is still no common ground which is the best for specific use cases, and specialized similarity hash functions are still an active field of research [3], for example *mrsh-v2* [2] which can identify file fragments.

Hashing each file per device by default can be used to easily **identify the same files across devices** and reduce the need for storing them multiple times. This is likely to further reduce the number of files that need inspection of any kind and save storage at the investigators backend due to deduplication. In particular with the use of cloud storage solutions like Dropbox or iCloud, many devices nowadays share local files which are kept synchronous across devices. However, the file system metadata of all copies needs to be preserved. Across cases and in the near future, efficient and privacy-preserving mechanisms will be needed to share hash value lists between multiple parties. Even though there are current mechanisms available to facilitate private set intersection [10], i.e. using zero-knowledge proofs [6], it is not yet known if they can be used for digital forensics and handle millions of hash values in practice. File system- as well as enhanced analysis of file **metadata** should be used in this step to compare file timestamps, EXIF metadata or other information sources in order to identify data sources and sinks and to reconstruct the flow of information across devices (and users). In very large environments with thousands of computers and users, this can be challenging.

Finally, the process should include the acquisition from various **online accounts** and the retrieval of the associated data and metadata using forensic methods. Online services like Facebook, Twitter, Apple or Google Services have hundreds of millions of users, and these online accounts are often tied to smartphones. While these companies have mechanisms in place to aid law enforcement, this source of information is not available to foreign civil law suits or other third-party investigations. Even though approaches have been recently proposed to acquire the data without the explicit aid of the service operators, e.g. using APIs [15] or based on observed network traffic [9] [22], they have not yet been incorporated in the standard processes. Cloud computing [1] can pose another, although related type of problem for digital forensics. Compared to online services and SaaS platforms, acquisition in IaaS cloud services is more related to the standard forensic process in direct comparison [20]. An obstacle is often how the investigator can access these services and whether or not the credentials needed for authentication can be obtained from the suspects, the hard drives or by other means. In most cases user consent is needed, and even though Great Britain is among the few countries that can convict a suspect if he/she is not releasing a password, this is not commonly found elsewhere.

3.2 Improved Forensics Process

Our improved steps for automated data analysis so far only enhance the current standards, in particular NIST SP-800 86. While RFC 3227 stops after the data acquisition, NIST SP-800 86 states specific steps to reduce the amount of files and data to analyze, i.e. using the NIST NSRL hash value collection. However, fuzzy hashing and cross-device checking are not mentioned, as well as the importance of online accounts for data storage and online services. It only exemplifies the use of multiple sources for data gathering, within a confined scope.

The core improvement in this paper is the parallelized calculation and evaluation of hash values, and the reduced working copy. As before, the data should be acquired according to the order of volatility, and using a hardware write blocker to prevent manipulations (accompanied by rigorous documentation). Before the image is created, file system metadata is parsed and all files in the file system hashed numerous times, including cryptographic hash functions like SHA-1 and fuzzy hash functions like *ssdeep* or *sdhash*. These hash values are then stored in some form of database and automatically evaluated with the proposed improvements: known, benign files are excluded using e.g. the NIST NSRL dataset, and multiple copies of the same file are detected across devices. Similarity hash values are used to detect similar files and present a set of candidates that seem related. This information can be embedded and enriched within an automatic timeline creation from file system metadata in the acquisition steps. Deleted files where the data has not yet been overwritten should be extracted and hashed similar to the other files. Furthermore, known malicious files can be found using hash value black listing. In the future, additional hash value calculations can be added as well as additional hash value sources. This can include novel fuzzy hash functions, other cryptographic hash functions like the upcoming SHA-3 or new hashing methodologies like sector hashing as proposed in [30].

After the automatic exclusion of files, the remaining files, folders and regions of free space are copied into the reduced working copy. Depending on the context of the analysis, this is expected to be sufficient for many cases. The use of cryptographic hash functions allows the argumentative exclusion of known files, since for each and every file there is a line of argumentation why this file was removed and ignored in further analysis steps. The final step is the optional extraction of online credentials from browsers, stored passwords or artefacts from online data services like e.g. Dropbox or iCloud. The entire process is visualized in Figure 1. Please note that the individual processing steps can be run concurrently: hashing the files may happen on the same byte stream as extracting the file system metadata, thus reducing the amount of read requests to the hard drive to the original bitwise copy as used today in digital forensics. Also, the extraction of online account information is considered optional, thus the different representation in the figure.

Most importantly, all of the steps discussed so far have the ability to run

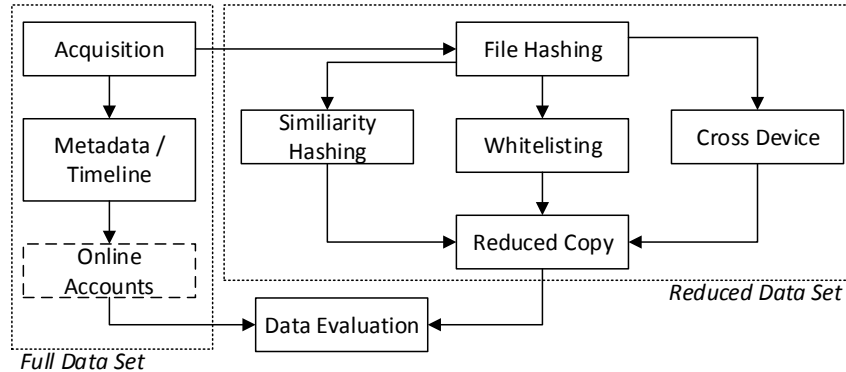


Figure 1: Improved Steps for the Forensic Process

automatically and present their findings in an understandable format to the investigator as well as in machine-readable form for further analysis steps. The computational overhead is very likely to be negligible compared to the additional insights using automated analysis as well as the possible reduction in the number of files and file fragments needing manual inspection.

4 Evaluation

In this Section we first describe the theoretical approach as described in the original paper by Neuner et al. [23]. This first part describes an artificial scenario. However, the specifics we used were derived during our ongoing informal discussions with law enforcement officials as well as forensic investigators. Building upon this theory and conducting an evaluation to prove the statement on a real-world corpus, the second part of this Section shows the practicability of the proposed deduplication approach as well as storage and performance improvements on 16 disks.

4.1 Theoretical Approach for Deduplication

On the theoretical basis we consider some form of malicious online activity as the initial reason for an investigation. The investigator is tasked with the acquisition of a relatively small number of devices from the following set, all devices which can be found in a modern household: computers or notebooks, smartphones respectively tablets, external storage devices like USB thumb drives or external hard drives, and lastly digital cameras. Furthermore numerous accounts at online services, e.g. Facebook, Google, Flickr or Twitter (just to name a few) which for the sake of brevity are omitted in our evaluation.

For the evaluation of our theoretical approach we used the following setup. We consider the investigated person to have the following devices in use: Two computers, whereas one computer is a Desktop PC and one computer is a Laptop. The windows PC is based on Windows 8 which uses roughly 160,000 files. We consider an additional total of 50,000 files to be from the user, including temporary working files and installed software. As described by Rowe et al. [28], commonly found hard drives include 18% Microsoft-related system files, 25% graphics (e.g. camera images), 4.7% documents (e.g. spreadsheets, presentations, etc.) and 4.3% executables to name the most important types. For mobility reasons the user has a Laptop computer with files daily mirrored with the Desktop computer and therefore these corpus’ share 80% of the same files. He/she uses an Android smartphone with about 13,000 multimedia files such as images, photos, videos and music files as described by Lessard et al. [19], 2,000 files which are related to different installed Apps (assuming about 300 files per App) and 20,000 files which are either related to running Google services or related to the Android operating system itself.

In addition to the mentioned computation devices (computers, smartphone) the specified setup includes two digital cameras with 2000 photos in total, split across three SD cards and several external storage devices used for backup. Those external storage devices include two external hard drives with half a terabyte and one terabyte in capacity, and three USB thumb drives from various manufacturers and with different capacities. These external hard drives contain the backed-up files from the Desktop PC as well as the notebook, respectively. The Desktop PC was used for backing up the files from the cameras and the smartphone, meaning that these files are found in the backup on the one terabyte hard drive as well. The USB thumb drives include an additional 20,000 files which are unique with respect to the other devices. Finally data is spread over the computers and the smartphone via a cloud service (e.g. Dropbox) and kept in sync with a remote copy. Therefore a large number of the user files are available on Desktop PC, Laptop as well as the smartphone. Please note that most of the specific numbers were chosen at random, they would differ in reality due to the different user’s age distribution, usage patterns, personal preferences, professional background and other factors. The overall capacity and number files for each device can be seen in Table 1.

Table 1: Overview of devices and storage capacities

Device	Storage Capacity	# of files	used capacity
Windows 8 PC	1tb	210k	250gb
Windows 8 Notebook	500gb	190k	180gb
Android Smartphone	32gb	35k	15gb
SD Cards	{8 16}gb	2k	10gb
external hard drive	{500gb 1tb}	400k	430gb
USB thumb drive	{4 8 16}gb	20k	32gb
Sum:	3.16tb	857k	917gb

4.2 Evaluation of the Theoretical Approach

The regular forensic process would need to acquire and copy each device at least once, resulting in the need to store roughly a little more than three terabytes of data only for the device images. If a backup copy is needed, this adds up to 6.2 terabytes of storage capacity needed. Overall, this would also mean extracting and analyzing 857,000 files. In the improved forensic process, however, an overall list for the entire case and thus all the devices is created which contains file names and hash values of all the unique files. This list also includes all metadata for the deduplicated files. To further reduce the number of files to be extracted for the working copy, the content is compared to available software reference lists like the NSRL. These steps allow to drastically reduce the numbers. The first reduction is caused by having redundant device contents on multiple devices. The Desktop computer and the Laptop both have their backup exclusively on their external USB backup drives. 80% of the Laptop user files (30,000) are duplicates from the Desktop PC, leaving 20% or 6,000 unique files as difference between the user files on the Desktop PC and the Laptop (due to cloud sync and working copies). As such, and starting with the acquisition on the Desktop PC, 210,000 files are to be extracted from the Desktop PC while the acquisition of the Laptop deduplicates the operating files and most of the user files. Therefore 184,000 files are duplicates and not added to the reduced working copy.





One cloud service is in use which synchronizes files over the Desktop PC, the Laptop computer as well as the smartphone, including the pictures of the user. A typical Desktop PC contains about 7.6% camera images as of [28], which would be in this particular case roughly 16,000 pictures on the PC. This is a superset of the pictures from the smartphone, including the audio and video files outside of the synced folders, leaving 22,000 files on the smartphone to be included in the reduced working copy. The 2,000 pictures on the digital cameras (stored on three different SD cards) were already synced to the Desktop and are thus duplicates in this case.

The last step is the removal of commonly found files e.g. using the NIST NSRL RDS. According to Rowe [27], 32% of a typical hard drive can be matched with files contained in the RDS set. This reduces the 210,000 files from the Desktop PC to roughly 143,000 files, and the files uniquely found on the Laptop to 4,080 files. Table 2 illustrates files to be extracted per source in the corpus. Grayscale areas mark the proportion of files that have to be extracted from that specific source, whereas white areas are duplicates that do not have to be taken into account for the created reduced working copy.

4.3 Real-World Evaluation Corpus

The evaluation was carried out on a real-world dataset from 16 participants captured in an IT consulting company and research institution respectively. This

Table 2: File extraction distribution per source in corpus.

Desktop PC		68%
Laptop		2%
ext. USB devices		5%
SD card / camera		
Cloud		
Smartphone		63%

company has a managed network, which is used to install the latest, stable Windows operating system on all clients as well as the corresponding updates. To be more precise, Windows 8 or Windows 8.1 was installed on all captured disks respectively. To capture the data stored on the disks the tool *tsk_loaddb*³ was used which is included in the well-known *sleuthkit* (TSK) [7] forensic investigation software. This tool stores important information about every file and directory on-disk to a sqlite database, above all a cryptographic checksum (MD5) and the size. Two important modifications were made to the corpus: To reduce the size of the corpus and store only the data needed for evaluation (checksums and sizes) all tables except the table *tsk_files* were dropped. Furthermore, database rows containing temporary Windows files were deleted. This deletion process includes files such as *\$BadClus:\$Bad* which is a sparse file, including a named stream created by NTFS [29]. Additionally, the columns *name* and *parent_path* were filled with *NULL* to preserve the privacy of the participants. Considering those 16 captured disks and only the remaining table, the corpus contains roughly ten million database rows in total.

4.4 Real-World Evaluation Design

Considering those millions of files the evaluation setup is as follows: The platform the evaluation was performed on is an Ubuntu Server with 16 physical cores and 72 gigabytes of memory. The developed evaluation application is written in Python. It reads all databases stored by *tsk_loaddb* and stores them in Python lists or dictionaries respectively. Keeping all of this information in memory, the comparing steps are faster and not I/O bound. Otherwise, the limiting factor would be reading the databases from disk.

The following describes the different evaluations carried out on the dataset to show the effectiveness of the proposed approach:

NSRL Reduction: The NSRL Reference Data Set (RDS)⁴ in its *minimal* version stores over 42 million unique hashes for identification of known files. Making use of this dataset, the first part of the evaluation covers the comparison of our corpus to the NSRL RDS.

³Online at: http://wiki.sleuthkit.org/index.php?title=Tsk_loaddb

⁴Online at <http://www.nsrl.nist.gov/Downloads.htm>

Cross Comparison: This evaluation step covers the comparison of one corpus database to the remaining databases.

Incremental Reduction: To show the distribution of reductions over every database, this part of the evaluation shows the incremental reduction of one database to the remaining databases. In more detail, let D be the databases from one to n , n be the total number of databases and x the database to be compared, then:

$$D = \{1..n\}$$

$$\Pi_i = \{x|x \in D \setminus D_i\}$$

$$f_{(i,j)} = D_i \Delta \Pi_{i,j}$$

Figure 5 shows this as an average distribution over all databases in the corpus.

5 Results

Considering the evaluation in Section 4 being splitted into two parts, this section is splitted in the same manner: The theoretical approach as well as the evaluation on the real-world corpus. The first part describes the results of the theoretical approach on the artificial dataset; in the second part we show that deduplication at the time of acquisition significantly reduces the number of to-be-saved as well as the required storage.

5.1 Results of the Theoretical Approach

As described in the previous section, the working copy of the artificial dataset will finally contain 189,000 files. This means a total reduction of 78% compared to the full dataset. Furthermore considering the average size of 1.1 megabytes per file, the working copy is reduced by 709 gigabytes. Since a reduction of size also means a reduction in computing power (e.g. hashing of files) an investigator experiences an overall performance enhancement. The percentages of the overall reduction in files and storage space on the artificial dataset are represented in Figure 2.

5.2 Results of the Real-World Corpus

To state the improvement of the proposed approach, Table 3 shows the number of hashes (and therefore number of hashable files and directories) as well as the corresponding sizes for each database in the corpus.

In regards to the evaluation design the results of the evaluation are splitted into three parts.

NSRL Reduction: Figure 3 shows the deduplication rate when comparing each database of the corpus to the NSRL RDS dataset. The NSRL RDS

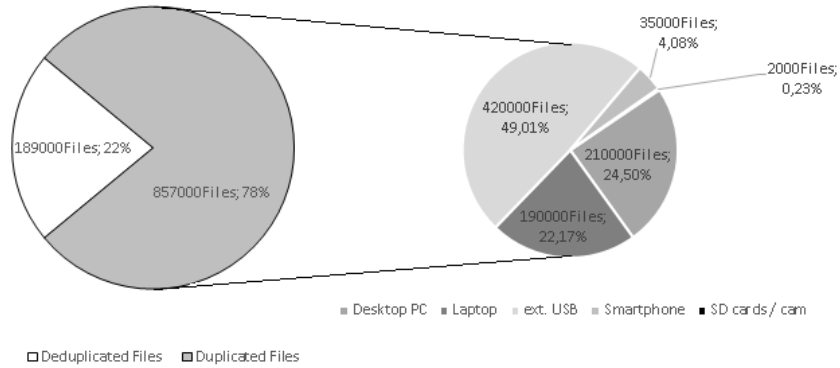


Figure 2: File reduction in the reduced working copy

dataset used in this evaluation contains 42,060,541 unique MD5 hashes (RDS minimal version) not including any size information.

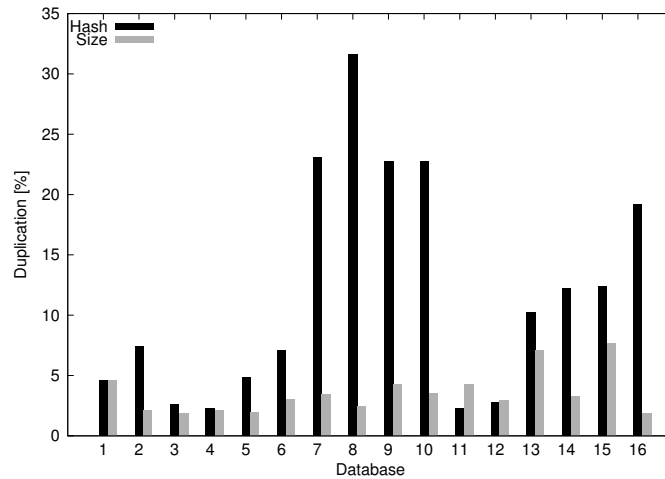


Figure 3: Deduplication rate of the corpus databases to NSRL RDS

As indicated by Figure 3 every database in the corpus can be reduced in terms of hashes as well as size when compared to the NSRL RDS dataset. The black bars show the duplication in percent for the hashes, whereas the gray bars show the duplication in percent for the size. Database 1 is the rare case of having the same relative amount of duplication in hashes and size at 4.59% (meaning a possible deduplication of 14,210 files/directories and 4.18 GB). In contrast 31.60% of the hashes within database 8 can

Table 3: Corpus Details

Database	Total # of Hashes	Total Size [GB]
1.sqlite	199255	243.63
2.sqlite	244499	238.82
3.sqlite	222561	304.11
4.sqlite	224166	238.03
5.sqlite	503764	242.16
6.sqlite	382571	122.49
7.sqlite	309853	90.90
8.sqlite	618218	171.16
9.sqlite	497088	249.42
10.sqlite	713162	147.07
11.sqlite	203187	244.03
12.sqlite	226521	244.01
13.sqlite	708268	302.76
14.sqlite	304996	470.73
15.sqlite	623331	244.63
16.sqlite	334121	83.13

also be found in the NSRL dataset, which corresponds to 2.42% of its (database 8) size (5.45 GB).

Cross Comparison: When e.g. acquiring large amounts of data a high number of semi-identical disks, duplicates exist. Figure 4 proves this assumption.

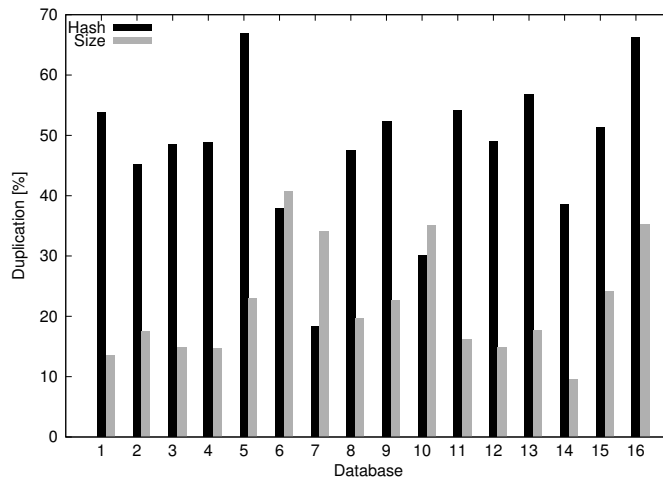


Figure 4: Deduplication rate of each corpus database to the remaining databases

It shows reductions in hashes and size for every database compared to the

sum of all remaining databases. Two strong outliers are database 5 and 16 with 66.81% (336,570 hashes) and 66.28% (221,465 hashes) possible hash deduplication respectively. Those numbers correspond to reductions of 23% and 35.16% in size, meaning absolute reductions of 55.59 GB and 29.23 GB respectively.

Incremental Reduction: To show the distribution of duplication throughout the corpus related to every database, Figure 5 illustrates an average over the incremental comparison as described in Section 4. The gray line corresponds to the average duplication in percent of the hashes, whereas the black line corresponds to the average duplication in percent in regards to the size of the databases. The x-axis describes the compared database to the initial database. To ensure the comparison of always the same sequence of databases, the Python built-in function `sort()` was used to pre-process the sequence (e.g. comparing the numbers [1,2,3,10], the sorted list according to Python `sort()` would be [1,10,2,3]).

To be more clear, describing this on the example of the first comparison: *1.sqlite* is the initial database, then $i + 1$ is database *10.sqlite* and so forth until *16.sqlite*, which is the predecessor of *2.sqlite*. In this example, $i + 15$ corresponds to *9.sqlite*, meaning the sum of all databases in the sequence from two to 16.

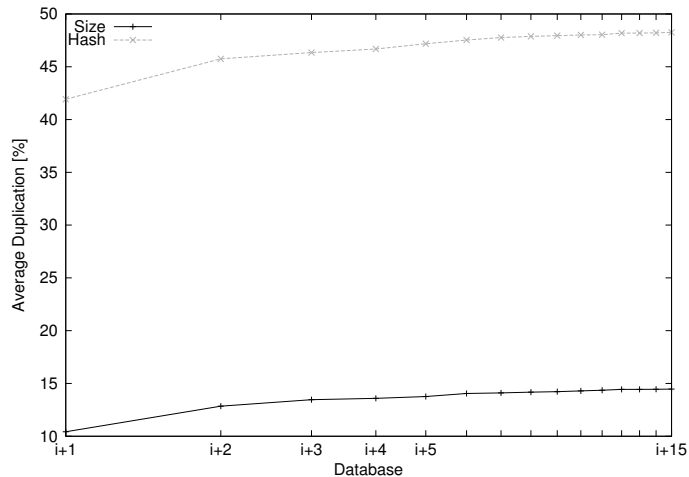


Figure 5: Average over the incremental deduplication rate

Figure 5 illustrates that the comparison of any database x_j to the remaining databases and their sum is an increasing trend. As expected, the highest number of the same hashes are found in the first compared database and slightly increasing with every incremented database. On average, the percentage of similarity is between three to five percent from the comparison of the initial database to the next and the comparison of

the initial database to the sum of all remaining databases (step i+15 in Figure 5).

5.3 Discussion

Those improvements can be applied to the data acquisition process, since they are totally transparent to the user applying them, e.g. the forensic investigator. Since we proved our stated, theoretical approach on a real-world corpus, we argue that this workflow can be beneficial to the majority of forensic cases. However, when comparing the theoretical and tested results, we have shown that the amount of file and size reduction is highly dependent on the underlying data corpus. If the number of unique files is low, or the unique files are large in size, the resulting improvements will be lower. Therefore, the NSRL RDS is a good start, but the overall reduction depends on the quality of the whitelist(s) applied by the investigator.

5.4 Limitations

Our evaluation is based on largely homogenous set of computers which share the same operating system; our findings can, however, be transferred to cross-platform investigations that are common in real-world cases. In this paper we clearly demonstrated the trend and showed the possibilities of file-based deduplication during image creation.

5.5 Future Work

For future work we implement a prototype which performs file-based deduplication as part of the data acquisition and thus during the creation of the forensic images. Ideally, this prototype will create an AFF image [12] which is smaller in size compared to the originating hard drive. The tool will be configurable to ignore the free space of the hard drive (once it has been processed and searched for possible remnant artefacts, e.g. using `bulk_extractor` [14]). If needed it is able to create bit-identical copies with the same hash values on-demand using the database of deduplicated files, similar to the recently proposed improved process by Golden et al. [24].

Based on our experience we do not expect processing power to be a bottleneck during image acquisition, since this step is usually bound by I/O capacity. Multi-core notebooks are common nowadays, and the additional overhead of creating hash values of all the files on-the-fly is expected to not increase the processing time per hard drive considerably. Alternatively, the tool could process the output of an initial run of `dfxml.py` [11]. Since the created XML contains all the needed information for file-based deduplication, including hash values and the location on disk, this would only require one additional (partial) reading of the hard drive for creating the deduplicated image, compared to multiple initial

readings of the entire disk.

6 Conclusion

In this extended paper we not only showed how an improved forensic process can be used to reduce the amount of storage requirement for forensic investigations by using file whitelisting and cross-device deduplication. While the metadata of duplicate files has to be preserved, our process is particularly useful in cases where the focus of the investigation lies on referenced files in the file system. We described an exemplary use case where file deduplication and file whitelisting were used to save 78% respectively 700 gigabytes of storage capacity.

Additionally we showed on a real-world data corpus consisting of 16 disks that file exclusion during the acquisition process saves storage and therefore processing power. When excluding files found within the NSRL RDS the reduction is about 3.5%, when cross-comparing the databases to each other the reduction is over 22% in regards to their size. Overall we hope that our improved process will lead to interesting discussions in the community as well as an improved standard forensic process in the near future.

Acknowledgements

The research was funded by COMET K1, FFG – Austrian Research Promotion Agency and by FFG grant 846070: SpeedFor.

References

- [1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, et al. A view of cloud computing. *Communications of the ACM*, 53(4):50–58, 2010.
- [2] F. Breitinger and H. Baier. Similarity preserving hashing: Eligible properties and a new algorithm mrsh-v2. In *Digital forensics and cyber crime*, pages 167–182. Springer, 2013.
- [3] F. Breitinger and V. Roussev. Automated evaluation of approximate matching algorithms on real data. *Digital Investigation*, 11:S10–S17, 2014.
- [4] D. Brezinski and T. Killalea. Rfc 3227: Guidelines for evidence collection and archiving. *Internet Engineering Task Force*, 2002.
- [5] D. C. Brock and G. E. Moore. *Understanding Moore’s law: four decades of innovation*. Chemical Heritage Foundation, 2006.
- [6] J. Camenisch and G. M. Zaverucha. Private intersection of certified sets. In *Financial Cryptography and Data Security*, pages 108–127. Springer, 2009.

- [7] B. Carrier. The sleuthkit (tsk), 2015. URL: <http://www.sleuthkit.org/sleuthkit/>.
- [8] A. Case, A. Cristina, L. Marziale, G. G. Richard, and V. Roussev. Face: Automated digital evidence discovery and correlation. *digital investigation*, 5:S65–S75, 2008.
- [9] M. Cohen. Pyflag—an advanced network forensic framework. *Digital investigation*, 5:S112–S120, 2008.
- [10] E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In *Financial Cryptography and Data Security*, pages 143–159. Springer, 2010.
- [11] S. Garfinkel. Digital forensics xml and the dFXML toolset. *Digital Investigation*, 8(3):161–174, 2012.
- [12] S. Garfinkel, D. Malan, K.-A. Dubec, C. Stevens, and C. Pham. Advanced forensic format: an open extensible format for disk imaging. In *Advances in Digital Forensics II*, pages 13–27. Springer, 2006.
- [13] S. L. Garfinkel. Digital forensics research: The next 10 years. *digital investigation*, 7:S64–S73, 2010.
- [14] S. L. Garfinkel. Digital media triage with bulk data analysis and bulk_extractor. *Computers & Security*, 32:56–72, 2013.
- [15] M. Huber, M. Mulazzani, M. Leithner, S. Schrittwieser, G. Wondracek, and E. Weippl. Social snapshots: Digital forensics for online social networks. In *Proceedings of the 27th annual computer security applications conference*, pages 113–122. ACM, 2011.
- [16] K. Kent, S. Chevalier, T. Grance, and H. Dang. Guide to integrating forensic techniques into incident response. *NIST Special Publication (SP) 800–86*, 2006.
- [17] J. Kornblum. Identifying almost identical files using context triggered piecewise hashing. *Digital investigation*, 3:91–97, 2006.
- [18] M. Kryder and C. S. Kim. After hard drives – what comes next? *Magnetics, IEEE Transactions on*, 45(10):3406–3413, Oct 2009.
- [19] J. Lessard and G. Kessler. Android forensics: Simplifying cell phone examinations. 2010.
- [20] B. Martini and K.-K. R. Choo. An integrated conceptual digital forensic framework for cloud computing. *Digital Investigation*, 9(2):71–80, 2012.
- [21] S. Mead. Unique file identification in the national software reference library. *Digital Investigation*, 3(3):138–150, 2006.

- [22] C. Neasbitt, R. Perdisci, K. Li, and T. Nelms. Clickminer: Towards forensic reconstruction of user-browser interactions from network traces. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1244–1255. ACM, 2014.
- [23] S. Neuner, M. Mulazzani, S. Schrittwieser, and E. Weippl. Gradually improving the forensic process. In *International Workshop on Cyber Crime (IWCC)*, 2015.
- [24] G. G. Richard III and J. Grier. Rapid forensic acquisition of large media with sifting collectors. *Digital Investigation*, 14:S34–S44, 2015.
- [25] V. Roussev. Data fingerprinting with similarity digests. In *Advances in digital forensics vi*, pages 207–226. Springer, 2010.
- [26] V. Roussev. An evaluation of forensic similarity hashes. *digital investigation*, 8:S34–S41, 2011.
- [27] N. C. Rowe. Testing the national software reference library. *Digital Investigation*, 9:S131–S138, 2012.
- [28] N. C. Rowe and S. L. Garfinkel. Finding anomalous and suspicious files from directory metadata on a large corpus. In *Digital Forensics and Cyber Crime*, pages 115–130. Springer Berlin Heidelberg, 2012.
- [29] R. Russon and Y. Fledel. Ntfs documentation, 2004.
- [30] J. Young, K. Foster, S. Garfinkel, and K. Fairbanks. Distinct sector hashes for target file detection. *Computer*, (12):28–35, 2012.