# Quantifying the Benefits of File Size Information for Forensic Hash Matching

Johan Garcia
*Department of Computer Science, Karlstad University, Karlstad, Sweden*

Keywords: Hashing, Digital Forensics, File Size Distributions.

Abstract: Hashing is a widely used technique in the digital forensic practice. By using file size information in addition to hashes, hash matching can potentially be made more effective since there is no need to calculate a hash value if there is no file in the hash set that has the same file size as the file being examined. Based on an examination of 36 million file sizes from five different data sets, this paper provides a quantification of the obtainable improvements. For the evaluated data sets the file reduction, i.e the fraction of files that can be skipped without hash calculations, ranged from 0.009 to 0.525. The byte reduction, i.e. the fraction of bytes that can be skipped, ranged from 0.514 to 0.992. Simulation results showed that these reductions in many cases could decrease the time necessary for hash scanning by 50% or more.



## 1 INTRODUCTION

In computer forensics the use of hash sets to compare the contents of a storage device to a set of known files have long been commonplace(Roussev, 2009). Various hash-sets are created by organizations to be used internally or shared externally to aid in different types of forensic investigations. Hash sets can be used to perform positive matching, i.e. identifying files on a storage media that do correspond to a file in the hash set. Additionally, negative matching can also be done to identify those files that *do not* correspond to a file in the hash set. The information structure of a hash set varies with the hash set provider, and the tool used to create the hash set. In many instances the hash-set is just a text file containing only one hash value per line. Other hash sets provide additional information, such as the name and directory position of the hashed file, the file size, a file type classification etc.

Forensic examinations are done differently according to the particular need of the individual case, legal requirements, available tools, etc. One frequent use of hashes and hash sets are to assist in examinations by identifying files that are either of particular interest in an investigation, or files that can be removed from an investigation as they are files known to be uninteresting such as unmodified operating system files, application files, and similar. Such scenarios are the main motivator of this work, as the inclusion of file size information has the potential to considerably improve the performance of hash matching. We do not consider hashing for uses such as providing forensic integrity or as a basis for naming in unified evidence repositories, as these applications typically requires hashes to be computed for all files regardless if they might match a hash set or not.

As storage devices continue to increase in size, the need to improve the speed of hash matching grows larger and larger. In order to quantitatively examine what potential gains the use of file size information can provide this paper performs an analysis of five different data sets from the perspective of file size distribution and overlap. Around 36 million file sizes were processed to generate reduction values that show what fraction of files, and bytes, that can be skipped from hash calculation because the examined file has a size for which there is no corresponding size match in the hash set. The results show file reduction values up to around 0.5 and byte reduction values above 0.9, implying that half of the files and more than 90% of the bytes need not be processed. Simulations modeling a mechanical hard drive show that these reductions often result in a 50% or larger decrease in the time required for hash processing.

## 2 BACKGROUND

Hashing is based on the use of hashing algorithms that produce a unique hash of a file. Typically the objec-

Garcia J..
Quantifying the Benefits of File Size Information for Forensic Hash Matching.
DOI: 10.5220/0004077303330338
In *Proceedings of the International Conference on Security and Cryptography* (SECRYPT-2012), pages 333-338
ISBN: 978-989-8565-24-2

tive is to make an exact match which is the case for this paper, but there also exists hash variations that do fuzzy matching such as ssdeep (Kornblum, 2006), with variations (Baier and Breitinger, 2011).

When performing positive matching the precomputed hash set contains hashes of files that are of particular interest to the examiner. Locating any occurrence of these files on the media under examination is thus the purpose of these kinds of examinations. One example where this methodology is commonly used are investigations related to Child Sexual Abuse (CSA) material. By performing locating hashing against a hash set of known illegal material such material can easily be found on the storage media.

Negative matching is instead used to remove files from further examination. In this case the hash set contains hash values from files known to be benign in relation to the examination conducted, such as unmodified application and operating system files. One example where this approach can be used is in the examination of infections of new malware. In these examinations it is necessary to work broadly as the malware could potentially have modified a number of files on the storage media in various ways to perform the various spreading, hiding, and anti-forensics functionalities that have been designed into it. By using excluding hashing files which are known to be unmodified can be safely excluded from further investigation, thus considerably decreasing the effort required.

File size information is a potentially useful piece of information to have in a hash-set in addition to the hash value. It is easily concluded that it is only necessary to compute hashes for files on the storage media which has a file size identical to a file size which exist in the hash set. Depending on the size distribution of the files in the hash set and of the files on the storage media, a smaller or larger fraction of the files on the storage media can be skipped without having to compute a hash value for them. This contributes to a corresponding decrease in the amount of time needed to process all the information on the storage media. Time can be saved both from the perspective of not having to read in all the file contents and compute the hash as well as avoiding a seek operation to the location of the storage media where the file is located.

The purpose of the work reported here is to provide some empirically based intuition on the order of magnitude of the improvements that can be obtained by using side file size side information when performing hash matching. To make these examinations a number of evaluation data sets were used which are described in the next section.

# 3 EVALUATION DATA SETS

To perform the evaluation five different file size data sets from different sources were used. These data sets are of two different categories, the first category being hash data sets. These data sets provide file size information for files that are used to create hash sets. The second category are scan data sets that reflect actual contents of a number of storage devices. In an investigation hash sets are used when examining files on storage devices. The file sizes on a device follows a particular distribution, such as the distributions exemplified by the scan data sets. Some general statistics on the data sets are provided in Table 1.

Table 1: Data set characteristics.

| Data set | Number of files | Unique sizes | Total file size (GB) |
|---|---|---|---|
| **Hash sets** | | | |
| CSA | 180057 | 88498 | 255 |
| NSRL | 22502929 | 896382 | 5382 |
| **Scan sets** | | | |
| PC | 9984693 | 188545 | 1140 |
| GOVDOCS | 986278 | 340955 | 466 |
| RDC | 2689123 | 149012 | 4930 |

## 3.1 Hash Data Sets

### CSA Data Set

This file size data set was obtained from a law-enforcement organization in a European country. The organization keeps their own collection of files that are known to be illicit Child Sexual Abuse (CSA) material according to their national legal rules. This collection is used to create hash sets employed to perform positive, or locating, hashing when new incoming material is examined. At the date of data collection, the collection consisted of 180057 files. Since a number of those files have exactly the same size, the number of unique file sizes is lower (88498). These unique file sizes make up the CSA data set, for which the unique file size distribution is shown in Figure 1.

The left histogram shows the file size distribution in the range 0 - 100 000 bytes, with each bar representing the number of unique file sizes in a particular interval. For example, the interval 0 to 1000 bytes has quite a small number of unique file sizes, only around 100. Furthermore, we can see that in the range from approximately 10,000 to 40,000 the number of unique file sizes is very high. Following the downwards slope we can see that around a file size of 70,000 the number of occurrences are around 500.

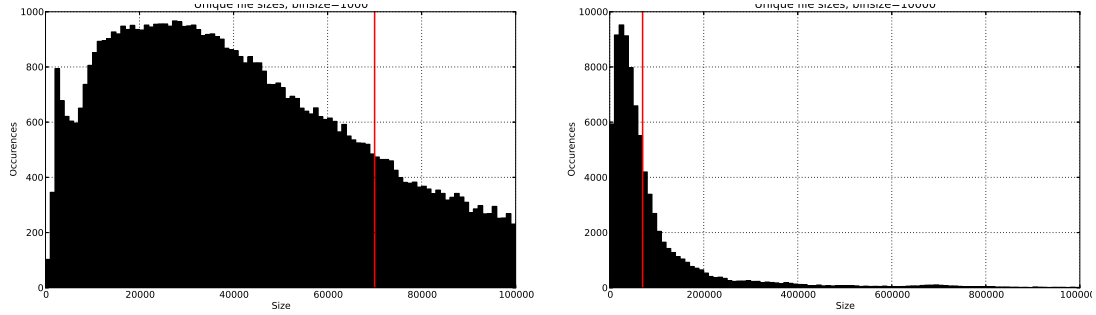


Figure 1: CSA data set characteristics.

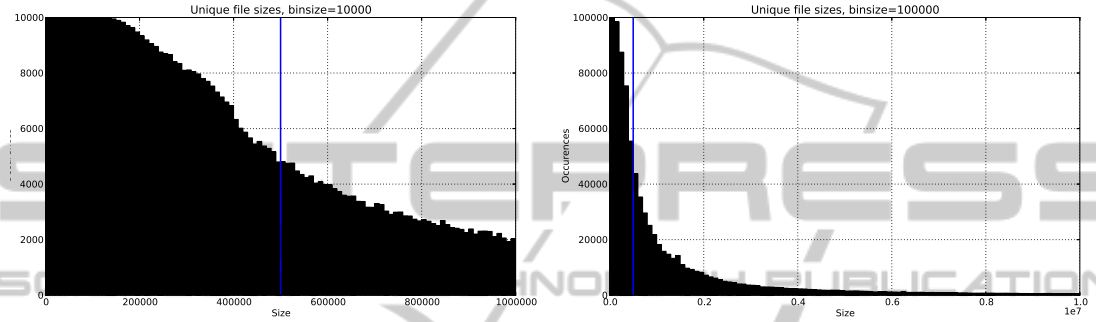


Figure 2: NSRL Data set characteristics.

When a hash set has 500 unique file sizes in a particular file size interval of size 1000 it means that when performing scanning, on average half of the scanned files that fall within that particular interval will have a size that is identical to a file in the hash set. This 50% point can thus serve as an visual anchoring point, illustrating the file size border where scanned files of larger size has less than 50% probability of requiring hash computation. The 50% point is marked by a red line. The right histogram shows the same data, but in this figure the X axis has been extended to 1 million bytes, and each bar now represents a file size interval of 10,000. It is clearly visible that for larger file sizes the number of unique file sizes that are present in this hash set is very low.

**NSRL Data Set**

A very large public hash set is provided by the National Software Reference Library (NSRL) project(NSRL, 2007). The stated goal of this project is to collect software from various sources and incorporate file profiles and hashes for the software into a Reference Data Set (RDS). The RDS hash set is typically used to perform excluding, or negative, hashing. For this examination, version 2.35 of the RDS was used which contains 74,555,829 files of which 22,502,929 are unique. These 22 million unique files together had 896382 unique file sizes, and the distribution of these file sizes are shown in Figure 2.

Again, the figure shows a histogram of the number of unique file sizes within file size intervals. Note, however, that in this figure the file intervals are a factor 10 larger than in Figure 1. If the same scale had been used the left picture would be completely black as is visible from the left part of the left figure. Clearly, this is an effect of the fact that this data set has many more files and thus a higher probability of all file sizes within a given file size interval. For the NSRL hash set the 50% point is around 700,000 and is marked by a blue line.

## 3.2 Scan Data Sets

In order to be able to evaluate the potential gain from using file size information when hashing, knowledge of the file size distribution of the media that is being scanned is also necessary. The purpose of the scan data sets are to provide file size distributions that could be similar to what an investigator may face when doing different kinds of forensic investigations. In this evaluation three different scanned data sets are used which are briefly presented here, before their examination results are discussed in the next section.

**PC Data Set**

The PC data set is based on file size information

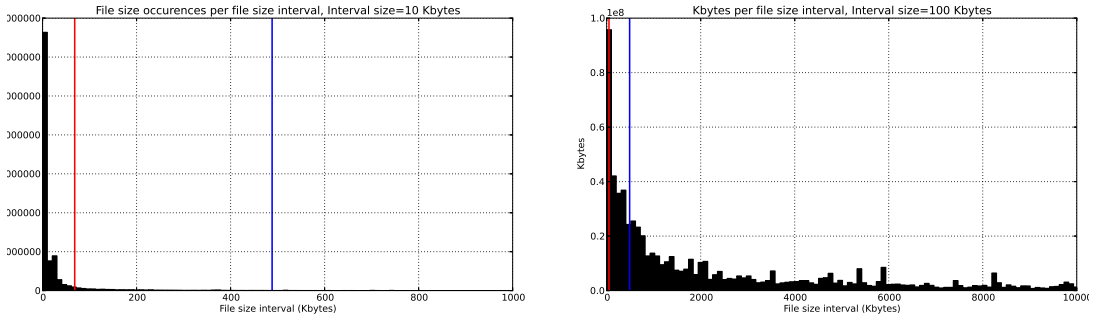


Figure 3: PC Data set characteristics.

collected from 57 public PCs located in a European Technical University. These PCs have operating system and application files installed. Some of them also to some extent contain additional files which were stored there temporarily, for example when performing downloads. On average, each PC had 20 Gbytes of files stored on it which in produced a total data set containing almost ten million different files with 188545 unique file sizes.

**GOVDOCS Data Set**

The GOVDOCS data set is based on the publicly available corpus of files that has been collected from public web servers as described in (Garfinkel et al., 2009). As such, this data set can be viewed as one representation of the file size distribution of files stored on public web servers such as those which were sampled during the collection.

**RDC Data Set**

The Real Data Corpus (RDC) data set comes from a corpora collected mainly by purchasing hard drives on the second hand market(Garfinkel and Shelat, 2003; Garfinkel et al., 2009). The file sizes in this data set are based on actual file size distributions found on the obtained storage devices, and are based on data retrieved by the `fiwalk` forensic utility. The version of RDC used to construct the file size data used in this examination has been sanitized in order to avoid privacy concerns, and is thus a somewhat reduced version of the complete RDC.

# 4 RESULTS

## 4.1 Matching of Distributions

We have seen from the above description that the majority of files in the hash sets are fairly small. Thus, the number of unique file sizes in relation to the file size interval is the largest for small file sizes. The point where 50% of the file sizes in an interval exists in the hash set is around 70 kB for the CSA data set and around 500 kB for the NSRL data set. This needs to be related to the distribution of file sizes for the three different scan sets.

The distribution of file sizes for the PC data set is shown in Figure 3. On the left the number of file occurrences are shown, and also shown in the figure are the scan set 50% points discussed above. The red line shows the position of the CSA 50% point and the blue line the NSRL 50% point. As can be seen the overwhelming majority of the files are so small that they fall below both points. Note that the interval size is 10Kbytes, and that the graph shows occurrences and not number of unique file sizes.

The right part of Figure 3 provides a graph showing the total number of bytes consumed by all files within each file size interval. The graph uses an interval size of 100Kbytes and shows that, while the majority of files seemed to be of a size smaller than both of the 50% points, this is not as true for the total number of bytes stored in the files. Larger files by definition hold more bytes, which drives up the number of bytes for the higher file size intervals even though the number of files in the interval is small. From the figure it can be reasonably assumed that the majority of bytes for this scan set belongs to files that would not have their hash computed. The corresponding graphs for the GOVDOCS and RDC scan sets are shown in Figures 4 and 5.

While the pictures provide a visual intuition, they can only show a restricted range. To provide a fuller picture numerical values were computed using all size values for each particular hash and scan set combination, which are shown in Table 2. File reduction is used as a metric and is defined as the fraction of files for which hashes need not be computed because there are no file in the hash set with a corresponding file size. For the PC data set, when CSA is used as the hash set, it is seen that the file reduction is 0.497. Thus, almost half of all files need not to have their

Table 2: File, byte and simulated time reduction results.

| Data set | File reduction | Byte reduction | File reduction (min 0.99) | Byte reduction (min 0.99) | Complete scan (s) | Size-assisted scan (s) | Time reduction |
|---|---|---|---|---|---|---|---|
| **CSA** | | | | | | | |
| PC | 0.497 | 0.887 | 0.492 | 0.667 | 65200 | 30535 | 0.532 |
| GOVDOCS | 0.487 | 0.947 | 0.481 | 0.922 | 20125 | 7640 | 0.620 |
| RDC | 0.525 | 0.992 | 0.521 | 0.890 | 9503 | 1787 | 0.812 |
| **NSRL** | | | | | | | |
| PC | 0.009 | 0.519 | 0.004 | 0.078 | 65124 | 61594 | 0.054 |
| GOVDOCS | 0.125 | 0.679 | 0.118 | 0.554 | 20129 | 14377 | 0.286 |
| RDC | 0.082 | 0.959 | 0.076 | 0.529 | 9515 | 3619 | 0.619 |

hash computed, which may at first seem strange given the shape of the PC file occurrence graph. When reviewing the CSA distribution in Figure 1 it however becomes clear that the very lowest file size intervals actually have a smaller number of unique file sizes, and this is where the majority of the PC file size occurrences fall. Going down to the NSRL hash set results, the results are markedly different. The PC file reduction is less than 1%, i.e. almost all files will have to have their hashes calculated. This is a result of the very dense coverage of files sizes shown in Figure 2. Turning to the byte reduction, which is the fraction of bytes on a storage media belonging to files which could be skipped, it can be seen that a useful reduction of the number of bytes that need to be processed can be achieved in all cases.

To examine the sensitivity of these results to the existence of a small number of large files in the scan sets a sensitivity analysis was performed. The top 1% of the files which had the largest file sizes were removed and the calculations redone. These results are marked in the table as (min 0.99). For the particular NSRL-PC combination, the heavy-tailed nature of the file size distribution and the actual number of files removed (99847) results in a significantly lower byte reduction for the sensitivity analysis case.

Considering the GOVDOCS and RDC scan sets, it can be seen that they provide similar reduction values as the PC scan set, mirroring fairly high values for the CSA hash set and lower for the NSRL hash set. They are also considerably less sensitive to removal of the largest files as shown by the sensitivity analysis.

### 4.2 Simulation Results

A simulation was also performed to estimate the time required to scan a 500 Gb hard drive completely full with files. The simulator randomly draws file sizes from the file size distribution of each scan set to fill up 500Gb. It then computes the times necessary to compute hashes for all files assuming that each file needs one random access to position the head, and that the disk read speed is the limiting factor rather than computing the hashes.

The hard drive characteristics were modeled from measurements on a Western Digital 2.5inch hard drive (WD500BEKT) with a measured read access time of 14.5 ms and average throughput of 81.6 Mb/s. 100 replications were performed for each hash and scan set combination, using a normal complete scan as well as a size-assisted scan. The average results are shown in the right side of Table 2. The 95% confidence intervals were less than 1% for all results. As can be seen in the results considerable reductions in the required time is achieved in a majority of cases, saving hours of processing times.

## 5 CONCLUSIONS

This paper aims to quantify the benefit of performing size-information assisted hash matching in the context of forensic hash processing. Contributions include the characterization of five different large scale data sets as a way to increase the knowledge surrounding file size distributions. Two different hash data sets were examined, in conjunction with three different scan data sets. Furthermore, simulations were used to quantify the potential impact on total hashing time when using file sizes to enhance hash processing.

File and byte reduction are used as primary metrics to signify the fraction of files and bytes that do not need to be processed due to the lack of a file with a corresponding size in the hash set. For file reduction, the values were between 0.009 and 0.525, while the byte reduction value ranged between 0.514 and 0.992. Thus, size information has the potential to considerably decrease both the number of files and the amount of bytes that needs to be processed when performing hash matching. Simulation results further showed that these reductions translated to considerable time gains, in many cases decreasing the time necessary for hash scanning by 50% or more. While the presence of large files have a relatively large impact on the byte reduc-

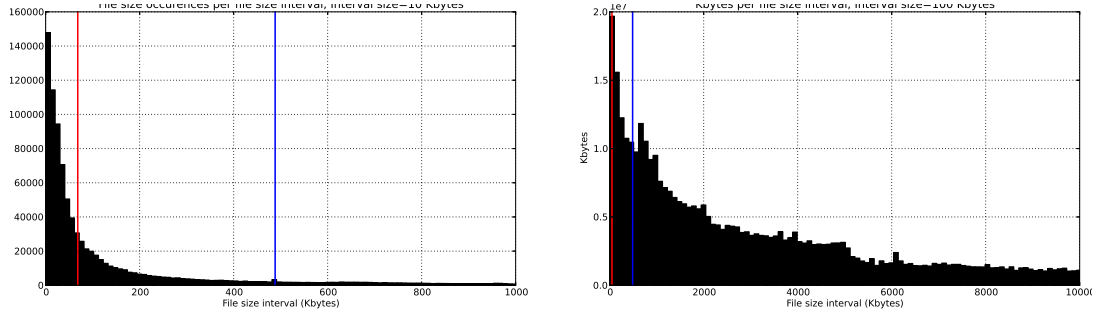


Figure 4: GOVDOCS Data set characteristics.

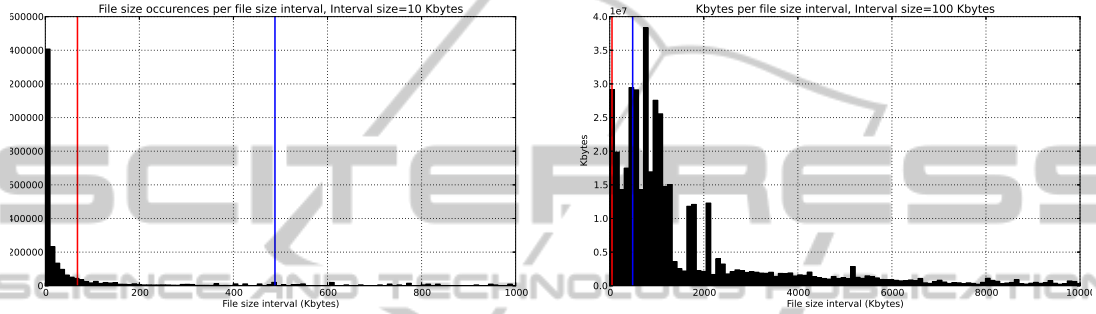


Figure 5: RDC Data set characteristics.

tion, a sensitivity analysis showed that the obtainable reduction is typically not dependent only on the presence of a very small number of very large files.

Possible avenues for future work includes further examinations using actual storage devices such as mechanical hard drives, SSDs, and USB memory devices. As shown in this paper the reduction factors are markedly different between file reduction and byte reduction. Different types of storage devices have different combinations of seek times and read speed and thus benefit differently from file and byte reduction. Further examination of how the reductions from this study translate into actual time gains for hash processing of different physical storage device types is thus one interesting topic for further study.

## REFERENCES

Baier, H. and Breitinger, F. (2011). Security aspects of piecewise hashing in computer forensics. *IT Security Incident Management and IT Forensics, International Conference on*, 0:21–36.

Garfinkel, S., Farrell, P., Roussev, V., and Dinolt, G. (2009). Bringing science to digital forensics with standardized forensic corpora. *Digital Investigation*, 6, Supplement(0):S2 – S11.

Garfinkel, S. L. and Shelat, A. (2003). Remembrance of data passed: A study of disk sanitization practices. *IEEE Security and Privacy*, 1:17–27.

Kornblum, J. (2006). Identifying almost identical files using context triggered piecewise hashing. *Digital Investigation*, 3, Supplement(0):91 – 97.

NSRL (2007). National Software Reference Library (NSRL). National Institute of Standards and Technology (NIST). U.S. Department of Justice's National Institute of Justice (NIJ), http://www.nsrl.nist.gov/.

Roussev, V. (2009). Hashing and data fingerprinting in digital forensics. *IEEE Security and Privacy*, 7:49–55.