File Fragmentation in the Wild: a Privacy-Friendly Approach

Vincent van der Meer*, Hugo Jonker†, Guy Dols*, Harm van Beek§, Jeroen van den Bos§, Marko van Eekelen†
* Zuyd University of Applied Sciences, {vincent.vandermeer,guy.dols}@zuyd.nl
† Open University of the Netherlands & Radboud University, {hugo.jonker,marko.vaneekelen}@ou.nl,
§ Netherlands Forensic Institute, {harm.van.beek,j.van.den.bos}@nfi.nl

***Abstract*—Digital forensic tooling should be based on reference data. Such reference data can be gathered by measuring a baseline in the real world, e.g. from volunteers. However, the privacy provisions in digital forensics tools are typically tailored for criminal investigations. This is not sufficient to ensure privacy obligations towards volunteer participants. Thus, privacy adaptations are necessary before such tooling can be used to establish or rejuvenate a baseline.**

We illustrate the feasibility of this approach by rejuvenating a baseline for file carving, via a case study of file fragmentation. We derive a set of privacy requirements to prevent deanonymisation of individuals. Atypical properties of files can nevertheless still lead to plausible deanonymisation of the file. With regards to fragmentation, we find *out-of-order fragmentation*, where a later block is stored on disk before an earlier block of the same file, occurs in nearly half of all fragmented files. This is the first study to report on prevalence of this type of fragmentation. Its high rate of occurrence has implications for the practice of file carving.

I. Introduction

Forensics is the science of finding and evaluating evidence in the context of a criminal investigation. Digital forensics is the application of this science to the digital domain. Both need to be grounded in reality. Moreover, technological developments are swift, implying that any baseline for digital forensics must regularly be recalibrated.

Consider the case of file carving, a technique to reconstruct files from their contents as blocks stored on disk, instead of from metadata. This technique is used in digital forensics to e.g. recover deleted files. However, the amount of blocks on a modern drive is enormous, which leads to a gargantuan search space for file carvers. To reduce the search space, file carvers make several assumptions, one of which is how blocks belonging to the same file are distributed over the disk: *file fragmentation*. The current de facto baseline for file fragmentation is the seminal study by Garfinkel [Gar07] from 2007. Technology has marched on since then: that study concerned mostly FAT12, FAT16 and FAT32 file systems, which have since been replaced by NTFS; disks have become (much) larger, operating systems have been updated, and laptops continue to replace desktops – and act far more as *personal* computers than traditional desktops. Finally, modern laptops often use a fast SSD drive for read-write intensive operation, and a slower HDD drive for data storage. Such a task-based division of drives did not exist at the time of Garfinkel's study.

Given these considerations, it is important to restudy file fragmentation, specifically to examine current, real-world occurrences of fragmentation, to improve the reliability and development of existing and future file carvers. Thus, a new study of file fragmentation would ideally be based on a large group of real-world, in-use systems.

However, there is a privacy problem. Studying file fragmentation requires determining the location of all blocks, which requires access to file metadata. But this metadata also contains information not pertinent to studying file fragmentation, some of which may be privacy-sensitive (e.g., filenames containing personal details). Such metadata must absolutely not be collected in measuring fragmentation. This entails that standard forensics tools cannot be used as-is, since these tools do not offer the required privacy provisions. Fragmentation tools outside the forensic domain typically focus on solving fragmentation, not on analysing it. Thus, there are currently no tools which can analyse fragmentation and simultaneously ensure the privacy of volunteer participants.

Contributions. The main contributions of this paper are:

- We establish a set of requirements to safeguard privacy when collecting file information
- We design a filtering wrapper for Fiwalk, a commonly used forensics tool, based on these requirements.
- We measure file fragmentation in the wild by a case study of 220 in-use, privately-acquired laptops and find:
  - the privacy requirements successfully prevent identification of individual users, but atypical properties of a file could lead to its plausible identification.
  - over 46% of fragmented files are fragmented out-of-order, a type of fragmentation whose occurrence was not previously reported on.

II. Background and Related Work

A. Fragmentation

NTFS divides a storage medium into partitions, volumes and, at the lowest level, blocks. NTFS organises access to blocks in a linear fashion, which introduces the possibility of file fragmentation. A file is *not* fragmented if its blocks occur adjacent (*contiguous*) and in consecutive order. In contrast, a file is fragmented when its blocks are separated by blocks not related to the file (*non-contiguous*), or when its blocks occur out of order. In Fig. 1, we illustrate how a file consisting of four blocks may be stored on disk:

- Fig. 1(a): without fragmentation,
- Fig. 1(b): with non-contiguous fragmentation,
- Fig. 1(c): with out-of-order fragmentation,
- Fig. 1(d): with both out-of-order and non-contiguous fragmentation.

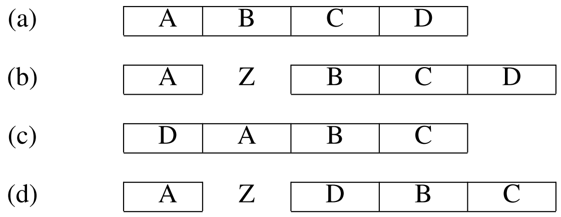

Fig. 1. How a file consisting of blocks A, B, C, and D may be stored on disk.

Note that since Windows Vista, Windows by default schedules a periodic defragmentation task.

Lastly, remark that while SSD firmware performs wear leveling (which distributes file blocks evenly over the physical storage), this is invisible to the file system.

B. NTFS features

The following NTFS features are relevant to the various ways the file fragmentation rate can be expressed.

- **Master File Table:** the Master File Table (MFT) contains the metadata of the files in a NTFS volume. For each file, the MFT holds the metadata (like timestamps, filename, size, permissions) and the allocation of blocks for the files data. The location of the MFT itself is stored in the boot sector [Car05]. To prevent fragmentation of the MFT itself, space is reserved for future growth.
- **Resident files:** resident files are files that do not have allocated blocks. Their data is fully stored in their record in the MFT. Typically, a resident file has a maximum size of about 700 to 800 bytes [Car05]. Note that resident files by definition cannot be fragmented, since no blocks are allocated to them.
- **Symbolic links:** a symbolic link (also called soft link) is an entry in the MFT that points to a path + filename. To resolve a symbolic link, the path and filename must exist. While symbolic links resemble files, they do not contain any data and thus cannot be fragmented.

The focus of this study was on collecting data on file fragmentation in a privacy-friendly fashion to benefit file carver development. Therefore, data related to other NTFS features such as junctions, Encrypting File System-files or alternate data streams, was not collected.

C. Related work

Garfinkel [Gar07] presented the seminal study on file fragmentation. He showed that if a file is fragmented, most often it is bi-fragmented (a file split into two parts). Moreover, he found that the rate of file fragmentation varies considerably between file types. Building on these findings, he constructed a file carver targeted specifically at bi-fragmented files. Following up on this, Roussev and Garfinkel [RG09] argued for a specialized carving approach: carvers tailored to specific file content types. Such carvers would be able to improve upon generic file carving techniques.

A different approach was taken by Cohen [Coh07]. He supplied a mathematical analysis of file carving. His analysis accounts for a model of fragmentation, to reduce the number of mappings from blocks to files. Notably, the approach he proposed allows to model different kind of fragmentation patterns, including contiguous out-of-order fragmentation.

Nevertheless, file fragmentation poses a problem for file carving. This was discussed in several file carving surveys, such as Pal and Memon's survey of the evolution of file carving [PM09]. They found that the first generation of carvers, based on file structure (using identification of file-header and -footer) was unable to handle fragmentation. Such carvers would typically 'recover' files that had unrelated content in their midst. This led to the development of new approaches to file carving, which aimed to overcome the difficulties of carving caused by file fragmentation. Similar findings resulted from the multimedia file carving survey of Pahade et al. [PSS15]. They concluded that multimedia files are often fragmented and compressed, and find that then-current carving tools are often unable to handle different types of fragmentation patterns.

Several more recent fragmentation studies have focused on Android. Ji et al. [JCS^{+}16] investigated file fragmentation on mobile storage systems. They studied fragmentation on four Android phones and found that file fragmentation had two extremes: either barely fragmented (mostly executable files), or highly fragmented. The latter category included database files (like SQlite). In addition, they found that file fragmentation negatively impacts read and write performance. In a follow-up study [CBJ^{+}17] on the effects of file fragmentation, the authors observed that fragmentation quickly emerged in the aging process and impacts user-perceived latencies. Another followup study [JCH^{+}18] found that traditional file system heuristics to mitigate fragmentation can fail under synthetic but realistic workloads, severely impacting I/O performance.

Finally, apart from Garfinkel's seminal study, only one other large-scale study of desktop computers reported findings on file fragmentation. In a study on file system content, Meyer and Bolosky [MB12] reported (as an aside) finding a level of file fragmentation of 4%. In addition, the most highly fragmented files within their corpus were log files. Note that it is not clear whether Meyer and Bolosky used the same definition to calculate the percentage of fragmented files as Garfinkel. Moreover, their respective corpora also differ substantially. Garfinkel used a corpus of devices ranging from 1998 to 2006, while Meyer and Bolosky's corpus consists of computers in use in late 2009.

The other area of related work revolves around *privacy-preserving digital forensics*. The notion of preserving an individuals privacy while trying to uncover relevant information about a suspected crime seems a contradiction. Nevertheless, several approaches have been suggested to enable some form of privacy during a forensics investigation. Most approaches focus on revealing only part of the (seized) data. One early use case where this emerges is in unavoidable attribution, which is sometimes proposed for privacy-enhancing technologies. For example, Olivier discussed [Oli05] how to enhance a method for obscuring web requests to assure attribution in case of investigation.

A different approach, more applicable in digital forensics, is to limit the data revealed to the investigator only to that which is necessary. Efforts in this vein seek to balance the privacy of the investigated subject (who may be innocent) with the needs of a forensic investigation. For example, Croft and Olivier [CO10] proposed an incremental model of cryptographic keys where knowing plaintext (facts) pertaining to less privacy-sensitive issues will reveal keys for higher privacy-sensitive information. The approach by Lawe et al. [LCY+11] similarly sought to limit the information disclosed to an investigator. In contrast, they chose to use cryptography to enforce regulation. They proposed to use a keyword encryption scheme to encrypt the data. Requests for decryption keys are then evaluated by a designated intermediate, who is responsible for privacy. Finally, Verma et al. [VGG18] proposed a privacy-preserving framework for analysing collected data.

While these approaches are interesting, they focus on limiting access to data in the analysis phase of a forensics investigation. In contrast, this study is focused on acquiring data to improve forensic tooling. Since we do not gather data in the context of a (criminal) investigation, privacy must be ensured *prior to* data analysis. A concept more in line with this is the concept of verifiable limited disclosure, as proposed by Tun et al. [TPB+16]. The authors recognised that more crimes are happening on social media, and proposed a way for witnesses to share their view of the social media with investigators, without providing full access. They proposed an approach where cloud providers encrypt and certify partial time lines for forensics investigation. While our setting does not use cloud providers, the idea of collecting only limited data is one that we will apply.

III. A Privacy-Friendly Approach to File System Data Collection

Ideally, all digital forensics is grounded in real-world data. However, digital forensic tools are privacy-invasive, which makes using these on volunteer participants (whose privacy must be ensured) problematic. Therefore, the development of privacy-preserving digital measurement techniques is foundational to advancing the field.

In order to gain an accurate view on modern fragmentation patterns, file fragmentation data from current, in-use devices is collected. To provide accurate data for this purpose, data should be collected from the general population. Acquiring fragmentation patterns requires a low-level view on a disk, specifically, knowing which blocks correspond to what file. Such a low-level view requires broad access to the disk. Since the data is collected from volunteer participants, privacy is a strong requirement. Measures must be taken to preserve participant privacy whilst collecting data. This will ensure privacy during analysis and further processing.

A. *Ethical aspects of data collection*

The study design was reviewed and approved by our institutional ethics review board. Participation was on a voluntary basis. All participants were explained what data would be collected, and what risks the data collection posed to their equipment. Before participation, volunteers were informed that since data collection was anonymous, their data could not be removed from the collection.

B. *Privacy Requirements and Operational Constraints*

To guarantee participant anonymity, we require that no elements of the data collection can be traced back to an individual participant. We consider an attacker that has access to the full data set, but not to any participant's device. Thus, any file system data that could potentially identify a participant must not be recorded. For each column in the MFT, we considered whether a user could, in the regular course of using a system, cause personal identifiable information to be present. These are the `Data` and `FileName` columns. Users can enter information in these columns via the file contents itself, the file name and extension, and directory names.

In addition, some entries in the MFT are special (meta-files). These may also user-settable information. We considered all special entries in the MFT. Of these, only the `$Volume` entry contains user-settable information (the volume name of a file system).

This leads to the following privacy requirements:

- **no-participant-info:** information about participants must not be recorded.
- **no-file-contents:** file contents must be excluded.
- **no-filenames:** filenames must be excluded.
- **no-file-paths:** file paths must be excluded.
- **no-volume-names:** volume names must be excluded.

However, the motivation for collecting data is to result in data that is useful for the file carving community. In particular, it is important to learn whether specific file types are fragmented differently than others. Moreover, barriers to participation should be reduced as much as possible. This leads to the following operational constraints:

- **only-safe-extensions:** only file extensions known to not contain privacy sensitive data are included.
- **performance:** data collection on a system should be finished within a reasonable amount of time.
- **no-change:** the evaluation must not change the contents of the observed file system.

C. *Implementation*

Acquiring data on file fragmentation can be done using three approaches:

- using user-grade disk analysis tools,
- creating a new tools,
- using forensic-grade disk analysis tools.

The objective of user-grade fragmentation-related tooling is to defragment rather than to report on current status of fragmentation. As such, this type of tooling is rather limited in what it reports about fragmentation – an insufficient level of detail for fragmentation analysis. In addition, these tools cause changes on the studied object (the defragmentation itself), which is neither needed nor desirable. A straightforward approach to creating our own tooling would be to copy the MFT. However, the contents of resident files are stored within each MFT-entry. To avoid copying this content would require substantial effort in parsing the MFT format. On the other hand, forensic tools already provide such functionality. They are ideally suited to provide the measurements we sought. However, privacy is typically not a consideration when designing forensic tools. Thus, such tools cannot be used out-of-the-box.

Fiwalk [Gar09] is a forensic tool for gathering details from a file system. It can be configured to omit file contents and not compute hashes of file contents. Not only does this dramatically improve scanning speed (*performance* constraint), it also happens to satisfy the *no-file-contents* requirement. To ensure the *no-change* constraint, we chose to collect data via a bootable USB stick instead of running on the host OS. We ensured privacy in the following ways:

- **no-participant-info:** No information about participants was recorded. Data was collected by visiting classes and asking for volunteers. Double participation was impossible: the researcher allowed each student in a class to only participate once, and students can only be enrolled in one class.
- **no-file-contents:** Fiwalk provides an option (for speed optimisation) to exclude file contents. This also excludes the contents of resident files. In addition, we used the Fiwalk option to not compute hashes of file contents (by default, hashes are computed).
- **no-filenames, no-file-paths, no-volume-names:** Fiwalk output contains filenames, paths and volume names. During the data collection phase, after Fiwalk completes gathering the metadata from a volume, we immediately process the data to remove these.
- **only-safe-extensions:** Fiwalk output contains the full extension. We post-process the data to only keep known extensions and extensions of three (UTF-8) characters or less. For the list of known extensions, we used a list of known extensions from Wikipedia[1].

This resulted in a script that executes Fiwalk and filters its output. This script is available from https://github.com/guydols/PriFiwalk.

D. Data Collection

Data was gathered between October 2018 and January 2019 by approaching students from one of our institutions to allow us to perform data collection on their (privately acquired) laptop. 260 students volunteered to participate. Unfortunately, our tool was unable to collect data on 40 laptops, e.g. due to absence of a USB A-port, legacy boot not being supported, or that the operating system on the USB drive was unable to boot on that system.

IV. Results

In total, we collected data from the laptops of 220 volunteers. 217 of these were running Windows 10, the other three were running Windows 7. Combined these laptops contained 334 drives. A common configuration was to find a laptop with both HDD and SSD (111 laptops), while three systems contained two SSDs. In 70 of the 106 systems with one drive, this was an SSD. All together, these drives were split into 729 volumes containing total of $\sim$86 million MFT entries, of which $\sim$69 million with allocated blocks.

Note that when reporting on file fragmentation percentages, we only take into account files that can actually fragment, that is, files of 2 or more blocks. The amount of resident files, symbolic links, and files with only one block of data allocated therefore have no influence on the reported fragmentation ratio.

A. Overall fragmentation results

Over the years, file fragmentation seems to be declining. In 2007, Garfinkel [Gar07] reported that 6% of files 'with data' was fragmented. Meyer and Bolosky [MB12] reported in 2012 that 4% of 'files' was fragmented. It is not clear if they consider this with respect to all MFT entries, all files 'with data' or some other ratio. For our study, we consider the most relevant fragmentation rate for file carving to be only files that could be fragmented, i.e., files of 2 or more blocks. We supply various fragmentation rates in Table I for comparison purposes. We distinguish between ratios for all MFT entries, for those with data (excluding symbolic links), for those with blocks (also excluding resident files) and for those with at least 2 blocks. Note that we cannot distinguish between empty and non-empty resident files – a consequence of the privacy-aware approach to data collection. The category 'files with data' therefore includes all resident files.

B. Number of fragments per fragmented file

Table II shows how many files are split into N parts, in percentages of the total number of fragmented files (rounded to two decimals, hence the numbers do not sum precisely to 100%). Note that the majority of fragmented files are fragmented into two parts (56.76%), most of which are fragmented in-order. Furthermore, remark that for all files split into 3

TABLE I. Fragmentation results

	MFT entries
All	84,390,537
With data	82,960,039
With blocks	70,320,268
With $\geq$ 2 blocks	42,671,054
Fragmented	1,871,109
Out-of-Order fragmented	868,917
% OoO of fragmented	46.4 %
% fragmented	
of all	2.2 %
of those with data	2.3 %
of those with blocks	2.7 %
of those with $\geq$ 2 blocks	4.4 %

[1] https://en.wikipedia.org/wiki/List_of_filename_extensions

TABLE II. DISTRIBUTION OF NUMBER OF FRAGMENTS (EXCLUDING NON-FRAGMENTED FILES)

# Fragments	*Total*	*In-order*	*Out-of-Order*
2	56.76 %	41.84 %	14.92 %
3	18.21 %	7.92 %	10.29 %
4	8.58 %	2.15 %	6.43 %
5	5.02 %	0.77 %	4.25 %
6	2.96 %	0.29 %	2.67 %
7	1.40 %	0.13 %	1.27 %
8	0.97 %	0.07 %	0.90 %
9	0.69 %	0.04 %	0.65 %
10	0.53 %	0.04 %	0.49 %
$\geq$ 11	4.89 %	0.34 %	4.55 %

or more fragments, out-of-order fragmentation occurs (much) more frequently than in-order fragmentation.

Files split into 11 fragments or more are over 14× more often fragmented out-of-order than in-order. Files split up in 100 parts or more make up for 0.6% of all fragmented files in the data set. The most fragmented file is a 2 GB `.bin` file that is split up in 20,464 fragments.

V. Analysis of Fragmentation Results

The presented fragmentation results provide a relevant view of the current state of file fragmentation on modern hardware. This provides valuable input into whether advanced file carving techniques remain useful in data recovery and if so, in what area their development should focus in order to increase performance.

A. Amount of fragmented data increases

When combined with earlier studies, there appears to be a steady decline in fragmentation over the years. In twelve years, fragmentation rate has decreased from 6% (Garfinkel in 2007, on all MFT entries with data) to 2.3%, a significant reduction. Whether this means that a smaller amount of data is fragmented is an entirely different matter.

We found that fragmentation of current HDD drives is very low at approximately 0.7%. However, in 2007 a typical $100 hard drive had a 500 GB capacity[2], while in Spring 2019, a hard drive in that price range has a capacity of 4 TB. So even though fragmentation is reduced by a factor of eight, the size of a typical hard drive has increased by that same factor.

SSDs were not considered in earlier studies, since they were not common in computer systems at the time. In the 2007 study, the average size of a disk was below 3 GB, so a 6% fragmentation rate would correspond to about 184 MB. The rate of fragmentation measured in this study would correspond to the same amount of data if current SSDs were only 8 GB in size. Given that the smallest SSD in our study is 16 GB, it is safe to assume that the amount of fragmented data has actually increased since 2007.

B. Relevance to file carving

Standard file carving can typically recover non-fragmented files. In our study, 97.7% of MFT entries with data were not fragmented and thus recoverable via a standard file carving approach. Nevertheless, there are cases where it is desirable to recover specific files, e.g., as they may contain key evidence. Standard techniques cannot do this. In particular, out-of-order fragmentation is not considered by current standard techniques. Yet, this type of fragmentation constitutes a significant portion of all fragmented files. For example, of all files that are fragmented into two parts, 25.75% is fragmented out-of-order. This means that a file carver focused on recovery of in-order two-part fragmented files (such as proposed by Garfinkel [Gar07]) will miss about a quarter of these files.

Carving for out-of-order fragmented files faces steep combinatorial complexity, but the amount of files fragmented in this fashion is sufficiently large (46.42% of all fragmented files) that in certain cases, recovery of such files will be worthwhile. In particular, there is a significant number of out-of-order fragmented files with two parts (14.46% of all fragmented files). This percentage is even greater than the percentage of all in-order fragmented files of three or more parts (11.89% of all fragmented files).

Cohen [Coh07] developed an advanced file carving approach to also recover contiguous out-of-order fragmented files (see Figure 1(c)). Interestingly, this type of fragmentation is nearly non-existent. In our data set of 42.3 million files with 2 or more blocks, only 8 files are fragmented contiguous out-of-order. All of those 8 files are fragmented into two parts. We conclude that while techniques for recovering out-of-order fragmented files are relevant, recovery of *contiguous* out-of-order fragmented files is currently not relevant.

VI. Discussion

A. Privacy

The goal of the privacy-filtering was to not collect any personally identifiable (PII) data. To this end, we removed filenames, path names, and unknown file extensions from the collected data. Nevertheless, the collected data does contain a lot of information about an individual drive's master file table (MFT). Thanks to the filtering, none of this information leads to an individual. Thus, the stated privacy goals have been successfully achieved.

However, in reviewing the data, we found that we could make plausible conjectures to the identification of specific *files*. In particular, an online search showed that certain combinations of exact, large file size and extension (`exe`, `mp4`) seemed to be unique. Note that plausible identification of files does not lead to identification of individuals: at best, it serves to distinguish a disk from others, not to identify its owner. Nevertheless, such plausible identification of files was unexpected and we consider this undesirable.

One theoretical mitigation would be to remove all file sizes from the data set. In this case, the file size can still be approximated from the block size and the (known) number of blocks, but the exact number of bytes is no longer available. In a small experiment, we tested whether a particular (very large) `exe` file would be harder to find when the exact size was not available. We found that the test file was still readily distinguishable amongst the search results. Thus, while removing the file size

[2] https://jcmit.net/diskprice.htm

would theoretically introduce more uncertainty, in practice the effect is negligible.

Therefore, we deem that removing the file size is insufficient to mitigate this problem in practice. Moreover, additionally removing the block size likely does not bring relief: in our data set, 97% of disks has the same block size. Thus, while participant privacy is successfully safeguarded, possible inferences regarding files must be further investigated before the data set can be made publicly available.

B. Generalisability of file fragmentation findings

The finding that the average fragmentation rate has dropped since previous studies, is corroborated by an examination of our own system's fragmentation. This is also strongly supported by the introduction of default weekly defragmentation in Windows systems since Garfinkel's study.

The second fragmentation finding is the rate of out-of-order (OoO) fragmentation. This type of fragmentation was previously not reported on. To validate this finding, we examined individual OoO rates for filesystems with more than 10,000 files in our data set. Of these, three quarters had an OoO-rate of over 40%. In addition, we measured file fragmentation on two fresh Windows 10 installations on virtual machines. Both installations were updated, one online, one with an offline update package. As expected, the fragmentation rates were low (0.2% tot 0.4%). We found that the percentage of fragmented files that was OoO-fragmented was 44.4% and 45.0%, respectively. These results show that our findings on out-of-order fragmentation are not user-dependent. Moreover, they are remarkably close to our finding of an average rate of out-of-order fragmented files of 46%.

VII. Conclusions

In this paper, we proposed a privacy-friendly approach to performing measurements using forensic tooling. We establish a set of requirements to prevent deanonymisation of participants. The requirements are derived by considering where personal identifiable information (PII) could be stored in the MFT. We developed a wrapper around Fiwalk to discard any such data. Our data collection attained the desired level of privacy: our data collection does not contain any PII. However, atypical properties of individual files may still allow the file to be deanonymised. While we believe this only applies in certain edge cases, we nevertheless consider this undesirable and will consider measures to address this in the data set.

The data presented in this study also provides new input for designing file carvers. As such, this data provides a basis for a discussion on which advanced file carving techniques to apply in specific situations. The major novel data points are that the percentage of fragmented files has reduced, the amount of fragmented data has nevertheless increased, and that there is a category of fragmentation that was not yet considered in practice: out-of-order fragmented files. In particular, 25% of files fragmented in two parts is fragmented out-of-order, and this rate quickly increases with the number of fragments. This provides actionable input into designing advanced file carving techniques.

Acknowledgements. The authors would like to thank all the volunteers (device owners) for their collaboration in this research. Van der Meer was supported by the Netherlands Organisation for Scientific Research (NWO) through Doctoral Grant for Teachers number 023.012.047.

References

[Car05] Brian Carrier. *File system forensic analysis*. Addison-Wesley Professional, 2005.

[CBJ+17] Alexander Conway, Ainesh Bakshi, Yizheng Jiao, William Jannen, Yang Zhan, Jun Yuan, Michael A. Bender, Rob Johnson, Bradley C. Kuszmaul, Donald E. Porter, and Martin Farach-Colton. File systems fated for senescence? nonsense, says science! In *Proc. 15th USENIX Conference on File and Storage Technologies (FAST'17)*, pages 45–58, 2017.

[CO10] Neil J. Croft and Martin S. Olivier. Sequenced release of privacy-accurate information in a forensic investigation. *Digital Investigation*, 7(1-2):95–101, 2010.

[Coh07] Michael I. Cohen. Advanced carving techniques. *Digital Investigation*, 4(3-4):119–128, 2007.

[Gar07] Simson L Garfinkel. Carving contiguous and fragmented files with fast object validation. *Digital Investigation*, 4:2–12, 2007.

[Gar09] Simson L. Garfinkel. Automating disk forensic processing with sleuthkit, XML and python. In *Proc. 4th IEEE Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'09)*, pages 73–84, 2009.

[JCH+18] Cheng Ji, Li-Pin Chang, Sangwook Shane Hahn, Sungjin Lee, Riwei Pan, Liang Shi, Jihong Kim, and Chun Jason Xue. File fragmentation in mobile devices: Measurement, evaluation, and treatment. *IEEE Transactions on Mobile Computing*, 2018.

[JCS+16] Cheng Ji, Li-Pin Chang, Liang Shi, Chao Wu, Qiao Li, and Chun Jason Xue. An empirical study of file-system fragmentation in mobile storage systems. In *Proc. 8th USENIX Workshop on Hot Topics in Storage and File Systems (HotStorage'16)*, 2016.

[LCY+11] Frank Y. W. Law, Patrick P. F. Chan, Siu-Ming Yiu, Kam-Pui Chow, Michael Y. K. Kwan, Hayson Tse, and Pierre K. Y. Lai. Protecting digital data privacy in computer forensic examination. In *2011 IEEE Sixth International Workshop on Systematic Approaches to Digital Forensic Engineering, SADFE 2011*, pages 1–6, 2011.

[MB12] Dutch T. Meyer and William J. Bolosky. A study of practical deduplication. *TOS*, 7(4):14:1–14:20, 2012.

[Oli05] Martin S. Olivier. Forensics and privacy-enhancing technologies - logging and collecting evidence in flocks. In *Advances in Digital Forensics, IFIP International Conference on Digital Forensics, National Center for Forensic Science, 2005*, pages 17–31, 2005.

[PM09] Anandabrata Pal and Nasir Memon. The evolution of file carving. *IEEE signal processing magazine*, 26(2):59–71, 2009.

[PSS15] Raj Kumar Pahade, Bhupendra Singh, and Upasna Singh. A survey on multimedia file carving. *International Journal of Computer Science & Engineering Survey*, 6(6), 2015.

[RG09] Vassil Roussev and Simson L. Garfinkel. File fragment classification-the case for specialized approaches. In *Proc. 4th IEEE Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE'09)*, pages 3–14, 2009.

[TPB+16] Thein Than Tun, Blaine A. Price, Arosha K. Bandara, Yijun Yu, and Bashar Nuseibeh. Verifiable limited disclosure: Reporting and handling digital evidence in police investigations. In *Proc. 24th IEEE Requirements Engineering Conference (RE'16)*, pages 102–105, 2016.

[VGG18] Robin Verma, Jayaprakash Govindaraj, and Gaurav Gupta. Df 2.0: Designing an automated, privacy preserving, and efficient digital forensic framework. In *Proc. 13th ADFSL Conference on Digital Forensics, Security, and Law (CDFSL'18)*, pages 128–149. ADFSL, 2018.