SCIENCES

J Forensic Sci, September 2018, Vol. 63, No. 5
doi: 10.1111/1556-4029.13722
Available online at: onlinelibrary.wiley.com

PAPER

DIGITAL & MULTIMEDIA SCIENCES

Eoghan Casey,[1] *Ph.D.*

Digital Stratigraphy: Contextual Analysis of File System Traces in Forensic Science

ABSTRACT: This work introduces novel methods for conducting forensic analysis of file allocation traces, collectively called digital stratigraphy. These in-depth forensic analysis methods can provide insight into the origin, composition, distribution, and time frame of strata within storage media. Using case examples and empirical studies, this paper illuminates the successes, challenges, and limitations of digital stratigraphy. This study also shows how understanding file allocation methods can provide insight into concealment activities and how real-world computer usage can complicate digital stratigraphy. Furthermore, this work explains how forensic analysts have misinterpreted traces of normal file system behavior as indications of concealment activities. This work raises awareness of the value of taking the overall context into account when analyzing file system traces. This work calls for further research in this area and for forensic tools to provide necessary information for such contextual analysis, such as highlighting mass deletion, mass copying, and potential backdating.

KEYWORDS: forensic science, digital forensics, digital evidence, digital stratigraphy, contextual forensic analysis, file system analysis, file allocation strategies, next-available file allocation, best-fit file allocation, valid data length slack, file initialization, file tunneling

When computing devices are used to create, alter, and destroy or conceal data by deleting, reformatting, wiping, or backdating files, it might be possible to determine the sequence of associated events using digital traces, even when the original file contents are not recoverable. Specifically, it can be useful to perform a type of contextual analysis using knowledge of file systems and nuances of allocation mechanisms. This work introduces novel approaches to contextual analysis of file allocation traces, collectively referred to here as *digital stratigraphy* because it has similarities with stratigraphy performed in geology and archeology.

Stratigraphy is the scientific study of layers (a.k.a. strata) with the aim of determining the origin, composition, distribution, and time frame of each stratum (1) Stratigraphic analysis is one of the cornerstones of archeology. Recent advances in digital forensic analysis have concentrated on temporal analysis (2–4). Prior to the present work, digital stratigraphy has not been systematically defined and applied in forensic science (5). Without these in-depth analysis methods, valuable clues remain buried within storage media.

Such contextual, stratigraphic analysis is particularly useful in digital forensic science when data were intentionally destroyed or falsified (e.g., backdating a document). When the creation time of a document is at issue, an examination of how data are positioned and overlaid on the disk might give a sense of when the document was created. Under certain conditions, it is possible to infer when deletion occurred on the basis of file allocation activities and significant gaps, even when the deleted files and their metadata are not recoverable. However, in some situations, it can be challenging to distinguish between concealment behavior versus normal system activity such as file initialization and file tunneling. Forensic analysts must be able to recognize such peculiarities in file allocation to avoid incorrect conclusions about digital stratigraphy.

This paper formalizes digital stratigraphy as a form of forensic analysis and provides illustrative case examples and empirical studies. Capabilities and challenges associated with digital stratigraphy are covered. File allocation operations that can confuse even experienced forensic analysts are explained. Potential areas of future research are proposed.

The novel contributions of this work are as follows:

- establish digital stratigraphy as a forensic analysis method;
- demonstrate that forensic analysis can reach verifiable conclusions about data destruction even when file contents or metadata are unrecoverable; and
- define requirements for digital forensic tools to support digital stratigraphy.

Furthermore, this work strengthens understanding of file system behavior, thus reducing the risk that forensic analysts will misinterpret file system traces.

File Allocation Strategies

Digital stratigraphy involves contextual analysis of how data are arranged on storage media. The arrangement of data on storage media is heavily dependent on file system allocation strategies. A potential misconception of forensic analysts is that new files are always saved onto storage media in an orderly fashion, using the next-available location (6). In fact, under controlled conditions, many operating systems save new files in the next-available unoccupied area on a hard drive. However, when operating systems are performing multiple simultaneous tasks, files may not be saved in a linear, contiguous sequence on the hard

[1]Ecole des Sciences Criminelles (ESC), Université de Lausanne, Batochime, CH-1015 Lausanne-Dorigny, Switzerland.

Received 8 Aug. 2017; and in revised form 29 Oct. 2017; accepted 30 Nov. 2017.

drive. Furthermore, there are some peculiarities and nuances in particular file systems that generate traces that can be misinterpreted. Before assuming next-available allocation strategy, it is important to take into consideration the specific context of the digital device and surrounding activities.

Next-available File Allocation

In the simplest scenarios, such as on a memory card in a digital camera, when files are saved in quick succession, there can be an orderly next-available allocation. In reference to FAT 16 file systems on earlier versions of Windows, including Windows 2000, Microsoft documentation states that "In a FAT folder structure, files are given the first available location on the volume" (7). Source code available on Microsoft's Web site provides insight into the FAT file system, including a next-available allocation strategy (8). This allocation strategy is also encountered on Linux-based devices that use FAT file systems such as GPS devices (9).

Such an orderly allocation strategy can allow recovery of deleted data even when the link between the deleted data and the corresponding file system entry is lost. For instance, Fig. 1 shows files salvaged from a digital video recorder that was reformatted. The reformatting operation obliterated all prior FAT32 file system metadata, including the file name, file size, 1st cluster, and created time of each video. However, because this device saved files using a next-available allocation strategy, each file ends where the next file begins. Employing content carving techniques to search for the AVI (RIFF) header at the start of each video successfully salvages the complete contents of every file on the device. In addition, the sequential order of these files indicates the order in which they were recorded on the device.

However, as noted in the following quote, the next-available allocation strategy does not always hold true, even on FAT file systems.

> I tested Windows 98 and Windows XP systems, and it appeared that a next available algorithm was being used in both. The next-available algorithm searches for the first available cluster starting from the previously allocated cluster. For example, if cluster 65 is allocated to a new file and then cluster 62 is unallocated, the next search will start at cluster 66 and will not immediately reallocate cluster 62... There are many factors that could affect the allocation of clusters, and it is difficult to identify the exact algorithm used.(10)

There are situations that will cause file allocation to deviate from such a deterministic, predictable next-available allocation strategy (11).

> Case example: Windows NT was used to save Microsoft Word documents onto a FAT formatted floppy diskette. Forensic analysis revealed that the documents were separated on the disk by areas filled with zeroes. One forensic analyst assumed a next-available allocation strategy, and misinterpreted these areas of zeroes as evidence of file wiping. However, experiments performed using a Windows NT system and the same version of Microsoft Word revealed

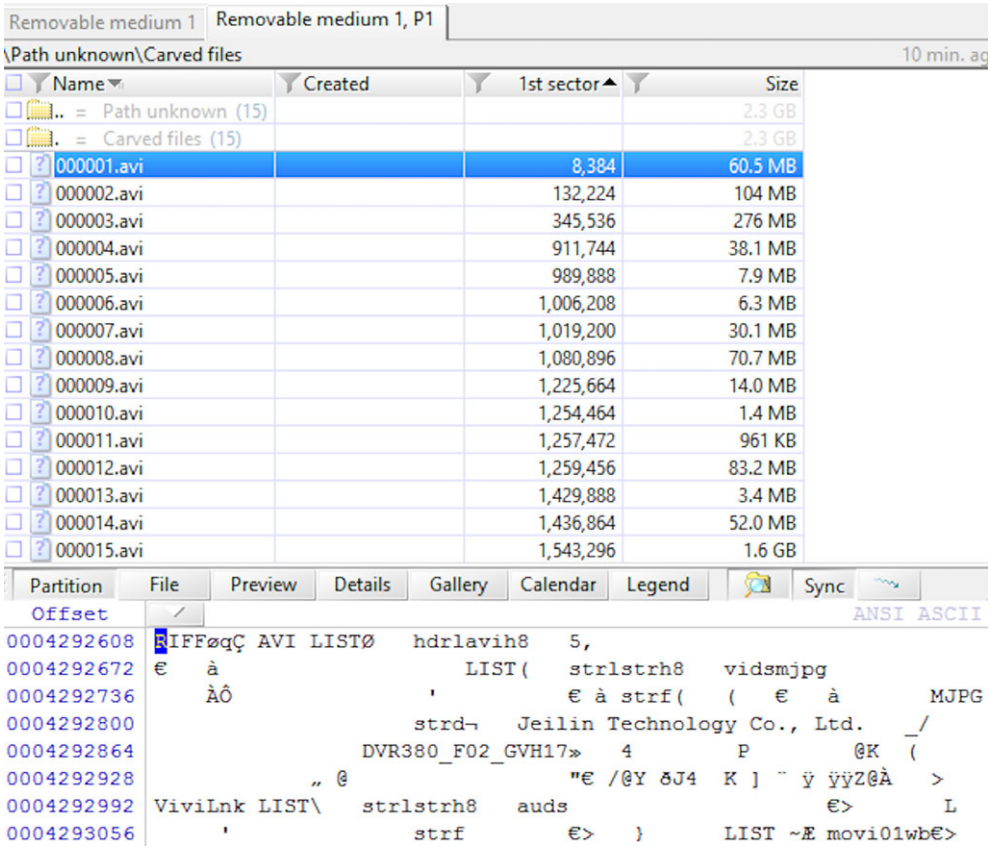


FIG. 1—*Video files salvaged from a reformatted digital video recorder using content carving techniques. Because this device used a next-available allocation strategy, each file ends where the next file begins, aiding recovery and indicating creation sequence. [Color figure can be viewed at wileyonlinelibrary.com]*

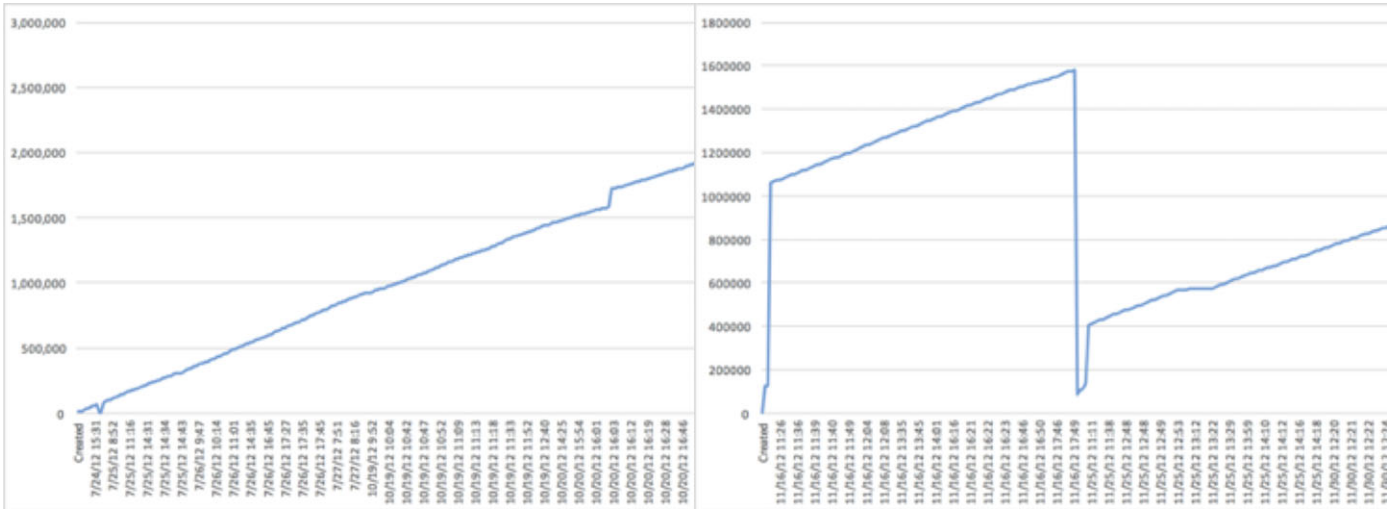


FIG. 2—*Plot of physical locations of files (first cluster on the* y*-axis) over time (created timestamps on the* x*-axis) from two FAT32 formatted SDCards used in digital cameras: (left) Canon Powershot ELPH 100 HS and (right) a Nikon D70s on the right. [Color figure can be viewed at wileyonlinelibrary.com]*

that normal file allocation left similar areas of zeroes between files (12). This file allocation behaviour could be due to new files being saved on cylinder boundaries.

In more recent operating systems, deviations from this strategy on FAT file systems appear to be caused by circumstances and usage of the storage media rather than any complexity in the allocation algorithm. Memory cards that were used in different digital cameras and mobile devices were analyzed in this study. For example, Fig 2. depicts files that were created on two FAT16 formatted media cards used in digital cameras. Depicted in Fig. 2 (left), photographs were taken without any deletion. Depicted in Fig. 2 (right), a series of photographs were taken and then deleted, then a second series of photographs were taken and subsequently deleted, and finally some additional photographs were taken. Each file is represented as a dot on the chart, with a line between sequential files to show the progression over time. The *x*-axis shows the created time of each file, and *y*-axis shows the physical location of the first cluster of each file. Similarly, Fig. 3 plots the chronological progression and corresponding physical locations of files saved on two FAT32 formatted media cards used in digital cameras. These graphs were generated by opening the storage media in a digital forensic tool (XWays Forensics) and exporting all recoverable file system metadata, including the created time and the physical location of the first cluster of each file. Both Figs 2 and 3 show that files being created over time are saved in steadily increasing clusters, with sudden drops when photographs are deleted from the device.

Removable storage media used in mobile devices typically store data from a range of applications and user activities, resulting in less orderly allocation of files. To observe file system activities in real-world contexts, six SDCards used in mobile devices were examined. The next-available allocation strategy can still be observed on FAT16 and FAT32 formatted SDCards that were used in mobile devices as shown in Figs 4 and 5.

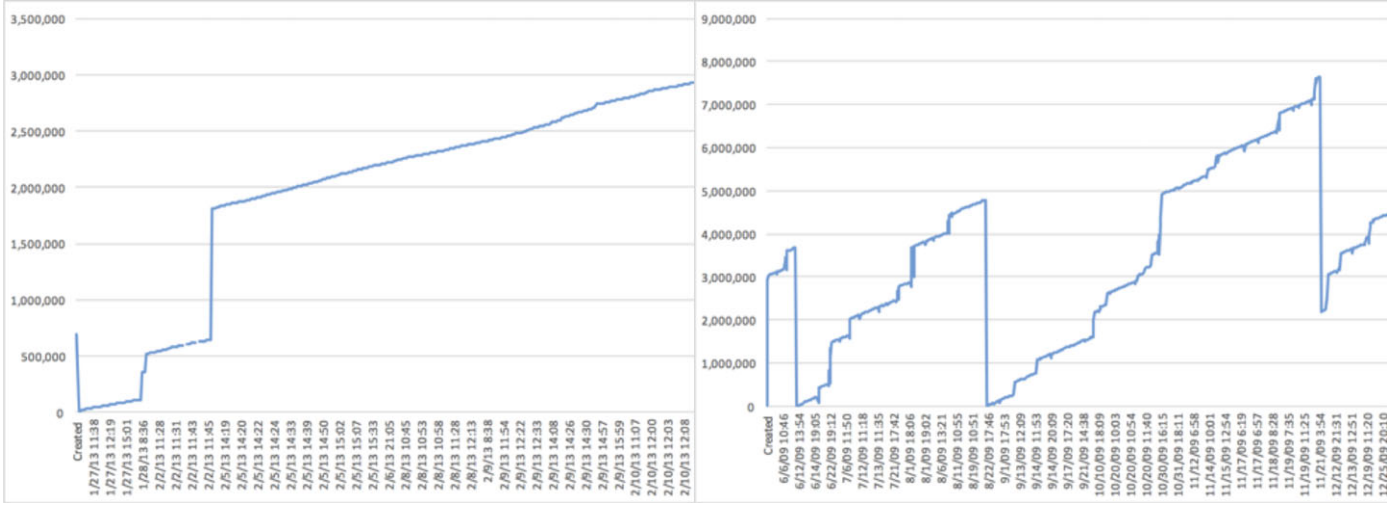


FIG. 3—*Plot of physical locations of files (first cluster on the* y*-axis) over time (created timestamps on the* x*-axis) from two FAT32 formatted SDCards used in digital cameras: (left) Canon Powershot SD1200 IS and (right) Nikon COOLPIX S600. [Color figure can be viewed at wileyonlinelibrary.com]*

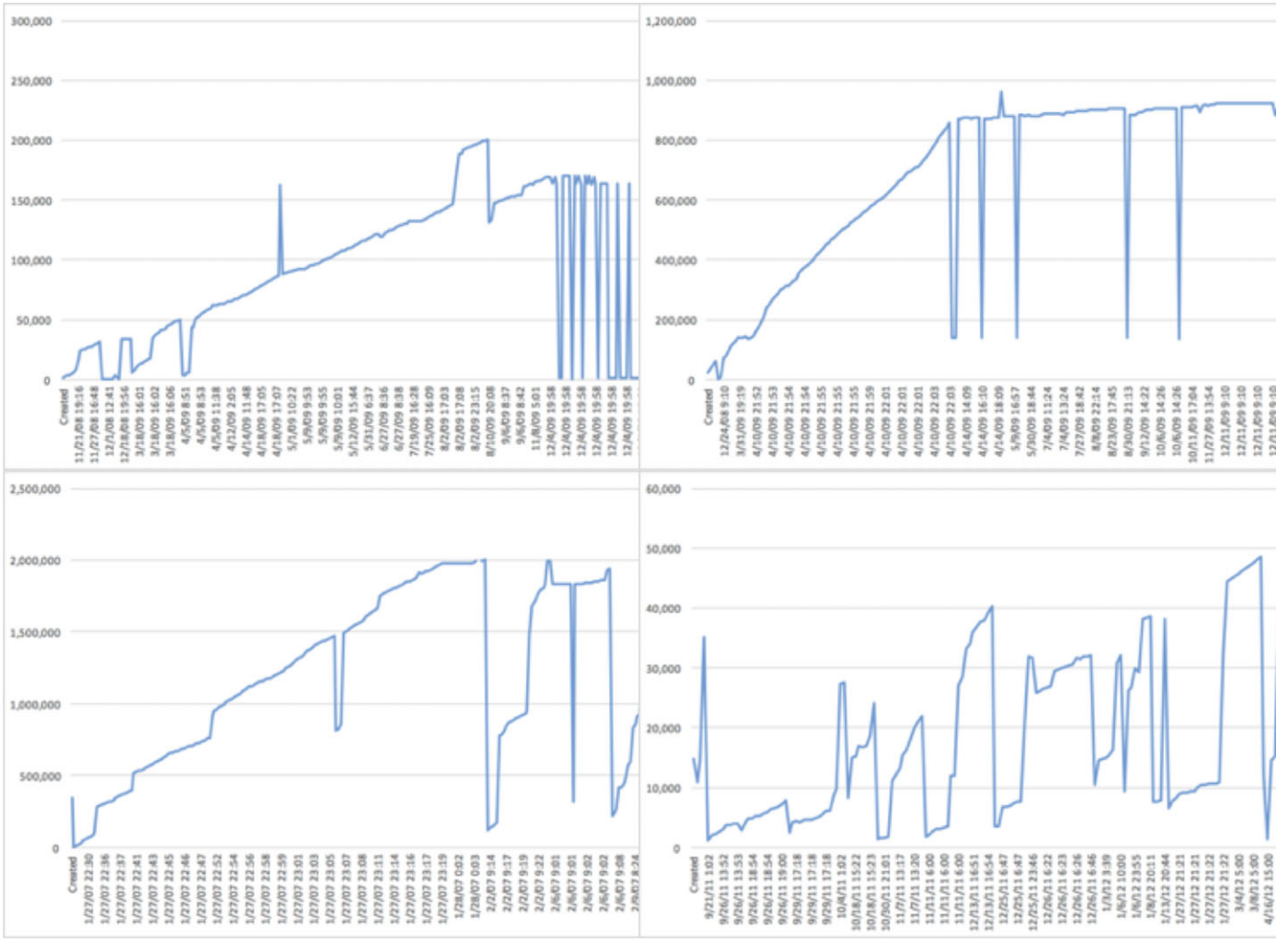


FIG. 4—*Plot of physical locations of files (first cluster on the* y*-axis) over time (created timestamps on the* x*-axis) from FAT16 formatted SDCards in mobile devices. [Color figure can be viewed at wileyonlinelibrary.com]*

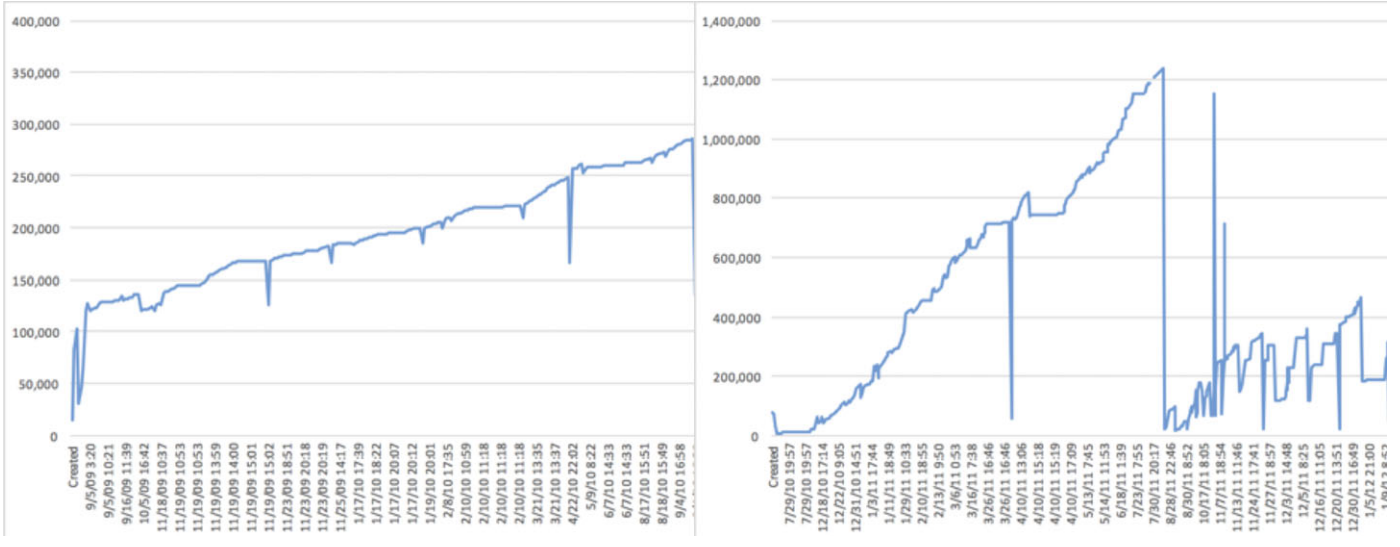


FIG. 5—*Plot of physical locations of files (first cluster on the* y*-axis) over time (created timestamps on the* x*-axis) from FAT32 formatted SDCards used in Android mobile devices, showing upward trends in allocation with some significant drops. [Color figure can be viewed at wileyonlinelibrary.com]*

When such an allocation pattern is in operation and a group of files are deleted, this can leave a gap that is discernible using forensic analysis.

Case example: In one matter involving alleged mass deletion, files had originally been copied onto the Windows system in bulk. These files were copied onto a FAT file system which used a next-available allocation pattern with the physical location on the hard drive increasing linearly over time. When the physical location of files that remained on the hard drive was plotted over time, there were significant gaps between these files. Even though file system metadata for the deleted files was not recoverable, contextual analysis of these gaps provided high strength of evidence apropos the claim that mass deletion had occurred, and low strength of evidence apropos the claim that mass deletion had not occurred.

The next-available allocation strategy of FAT file systems becomes less pronounced when files are regularly deleted from storage media or on system drives that have experienced regular reboots and user activities over an extended period of time. However, on such file systems, the next-available allocation strategy might still be discernible over narrow time periods, particularly when large amounts of data are deleted. In one matter, the file system allocation over 1-week time told a compelling story of mass deletion (significant drop in the physical location of created files), followed by a period of new files being saved onto the hard drive (linear increase in physical location over time), followed by reformatting (another significant drop).

Although usage of a file system over time can complicate the arrangement of files on storage media, it can also create opportunities for forensic analysis. For instance, creation of temporary files when editing a document can leave traces that reveal the chronology of these activities. In addition, use of disk fragmentation programs can group files together at a distinct point in time. In some cases, files that are found outside of this orderly grouping can be shown to have been saved to disk after the disk defragmentation occurred.

Case example: Disk defragmentation processes are commonly used to rearrange files on a hard drive for performance purposes. In one case of backdating, a questioned document could be shown to be backdated because it was saved onto the hard drive after the defragmentation process was run, on a date after the claimed creation date of the document.(13)

Another circumstance that can complicate forensic analysis on FAT file systems is when storage media is nearly full. In this scenario, file allocation cannot find enough storage space in higher numbered clusters and wraps around to the beginning of the disk to continue its search for next-available clusters (14).

Best-fit Versus Next-available File Allocation

Unlike FAT file systems, NTFS exhibits a best-fit strategy to allocated files, as described in Carrier.

The best-fit algorithm is when the data [are] placed in a location that will most efficiently use the available space, even if it is not the first or next-available. Therefore, if a small amount of data [are] being written, it will be placed in the clusters that are part of a small group of unallocated clusters instead of in a large group where larger files could be stored.(10)

Although NTFS file allocation is more sophisticated than FAT, a predictable increase can still occur under certain circumstances. For example, a backup storage device on which files are saved and rarely deleted will still exhibit the gradual increase in file allocation location as shown in Fig. 6.

However, NTFS file systems that are actively used do not follow such predictable allocation patterns because NTFS is designed to make efficient use of storage space. In fact, NTFS sometimes goes “backwards” when saving fragments of a file, allocating later pieces of a file in lower numbered sectors (Fig. 7). Although not easily replicated in a controlled manner, this allocation pattern occurs during normal use of a computer.

To perform digital stratigraphy and analyze deletion activities, forensic analysts and tool developers need a strong understanding of file allocation strategies. Understanding file allocation methods can provide insight into such concealment behavior, but

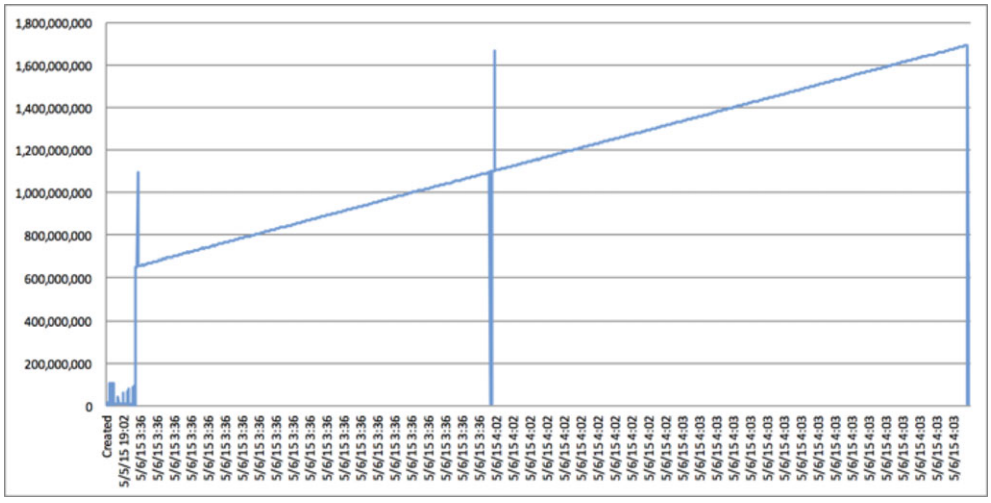
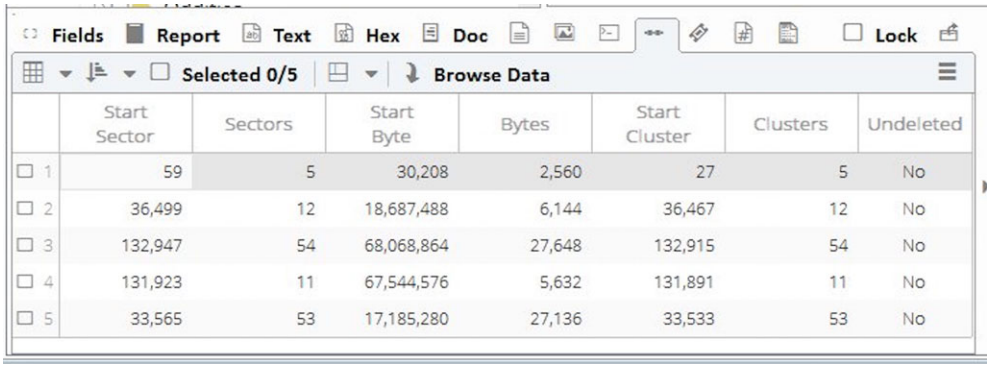


FIG. 6—*NTFS formatted storage media used for system backups shows increasing allocation locations over time. [Color figure can be viewed at wileyonlinelibrary.com]*



	Start Sector	Sectors	Start Byte	Bytes	Start Cluster	Clusters	Undeleted
1	59	5	30,208	2,560	27	5	No
2	36,499	12	18,687,488	6,144	36,467	12	No
3	132,947	54	68,068,864	27,648	132,915	54	No
4	131,923	11	67,544,576	5,632	131,891	11	No
5	33,565	53	17,185,280	27,136	33,533	53	No

FIG. 7—*Start clusters of five parts of a single file on NTFS shown using EnCase 8. The last two parts of the file were saved in lower number clusters than earlier parts of the file. [Color figure can be viewed at wileyonlinelibrary.com]*

real-world computer use introduces complexity that complicates forensic analysis. Each file system tells a story about the use of that storage media, and analyzing the allocation of files over time can sometimes provide insight into deletion or other concealment activities. As a result, general understanding of file allocation methods can only be used as a starting point, and it is necessary to take the overall context into account when analyzing traces of such concealment behavior.

It is important to note that not all digital forensic tools provide access to the file system details necessary to perform digital stratigraphy. Forensic analysts must use tools that are fit for this purpose, and developers can enhance digital forensic tools features that enable digital stratigraphy. In addition, when performing digital stratigraphy, it is important to consider peculiarities of file allocation that can create confusion: file initialization and file tunneling.

File Initialization

When Microsoft Windows is preparing to create a new file, a file system entry is created before any data are written to disk. In other words, a file system entry is created and disk space is reserved before the file contents are saved to disk. This process is called file initialization (15). Under certain circumstances, the storage space reserved for the new file may not be used in its entirety or at all. Cases involving the download of many large files from the Internet (e.g., pirated movies or software) often include interrupted downloads, resulting in incomplete files that occupy only part of the disk space that was originally initialized and reserved for the file. The portion of the initialized space that is actually used to store the new file contents is called the valid data length (VDL) (see Fig. 8).

Forensic analysts can expect to encounter initialized, incomplete files on both NTFS and FAT file systems. NTFS captures the difference between logical file size and valid data length in two MFT fields; this difference is called VDL slack (15,16). FAT file systems do not have fields to capture the difference between logical file size and valid data length, making it more difficult to detect when VDL slack is present.

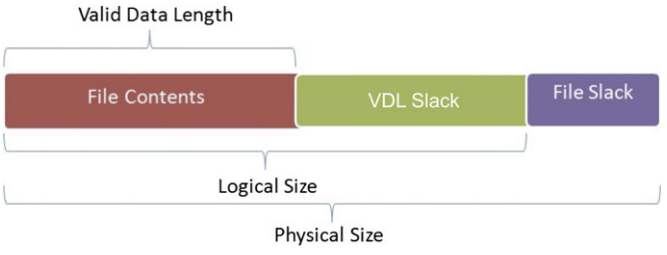


FIG. 8—*Diagram of file with a logical size that is larger than its valid data length, leaving uninitialized space. [Color figure can be viewed at wileyonlinelibrary.com]*

From a forensic analysis perspective, VDL slack can be beneficial because it can contain remnants of deleted data. VDL slack is similar in concept to file slack except that it is contained within the logical file size (see Fig. 8). Unlike file slack which is no longer associated with a file, data in VDL slack are in a kind of limbo, trapped at the end of an allocated file but not actually part of that file, effectively freezing deleted data within allocated files. Although various disk cleaning utilities can be configured to wipe file slack, they generally do not reach data in VDL slack. For example, the Eraser (version 6.2) and CCleaner (version 5.22) disk cleaning utilities have an option to wipe file slack (a.k.a. cluster tips), but not VDL slack. As a result, deleted data can remain in VDL slack indefinitely, even surviving after data destruction methods and tools have been used.

From a digital stratigraphy perspective, the data in VDL slack existed on the computing device before the containing file. For example, Fig. 9 shows a file that has an initialized size of 1,744,830,464 but a valid data length of 1,073,741,824 bytes (displayed at bottom right). So, this file has 671,088,640 bytes (640 MB) of VDL slack, which contains various fragments of deleted data that existed on the disk before this initialized, incomplete file was created. Extracting and searching this 640 MB of VDL slack for characteristics of common file types found portions of deleted pictures, audio, and Web-related traces completely unrelated to the new initialized, incomplete file.

When a file is initialized but the associated content was not saved to disk, the new (initialized) file system entry may point to a cluster that contains old data. This temporal discrepancy makes analysis of digital stratigraphy more difficult and can be misinterpreted as backdating, particularly when it is unclear whether VDL slack is present.

Case example: A dispute over ownership of a company hinged on the date that shares were transferred. Forensic analysis of FAT formatted floppy disks found a series of deleted digitized facsimiles (fax) of share transfer agreements. Each fax had the date of transmission imprinted at the top of the page. Forensic analysis also found an unrelated deleted file system entries ("NEWFILES") pointing to the same cluster that contained the deleted faxes. The NEWFILES had created dates that were after the transmission dates marked at the top of the faxes. Forensic analysts were

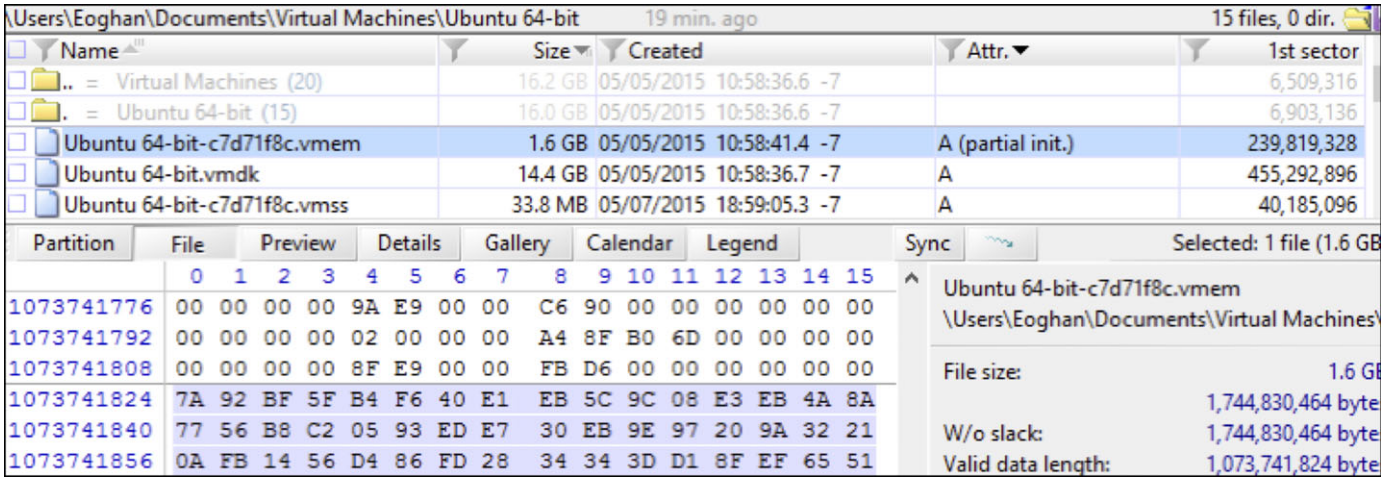


FIG. 9—*File with 400MB of VDL slack viewed using XWays Forensics 18. [Color figure can be viewed at wileyonlinelibrary.com]*

asked to weigh the two possibilities: (1) had the NEWFILES been overwritten by backdated faxes, or (2) had the NEWFILES been initialized but not saved, leaving the legitimate faxes in the cluster. It was not possible to prove that the NEWFILES were partially initialized because FAT file systems do not have fields to distinguish VDL. Ultimately, the judges decided that forensic analysis was "not conclusive of the issue of when the documents stored on the floppy disks were created or otherwise accessed."(12)

This situation can arise on NTFS as well. For instance, the first two files listed in Fig. 10 are partially initialized files on an NTFS file system. The first file (MVI_6420.AVI) is easily detected as an initialized, incomplete file because its logical size is larger than its valid data length. The second file (MVI_6362.AVI) appears to be zero bytes in size, but this is erroneous information due to an interruption before the file system entry was completed (partial initialization). In such situations, it can be difficult to ascertain whether the file was partially initialized. Even when such a partially initialized file is detected, it can be difficult to determine the provenance of the cluster contents. Did the cluster contents belong to a previously deleted file, to the new (partially initialized) file, or to an intentionally backdated file?

The challenges associated with analyzing initialized, incomplete files and partially initialized files can be more complicated with forensic tools that display zeroes instead of the actual disk contents as shown in Fig. 11. This zeroing of data occurs when the Microsoft operating system is relied on to read cluster contents. As explained in Microsoft's fsutil documentation "Any reads between VDL and EOF automatically return 0 in order to preserve the C2 object reuse requirement" (17). Forensic analysts

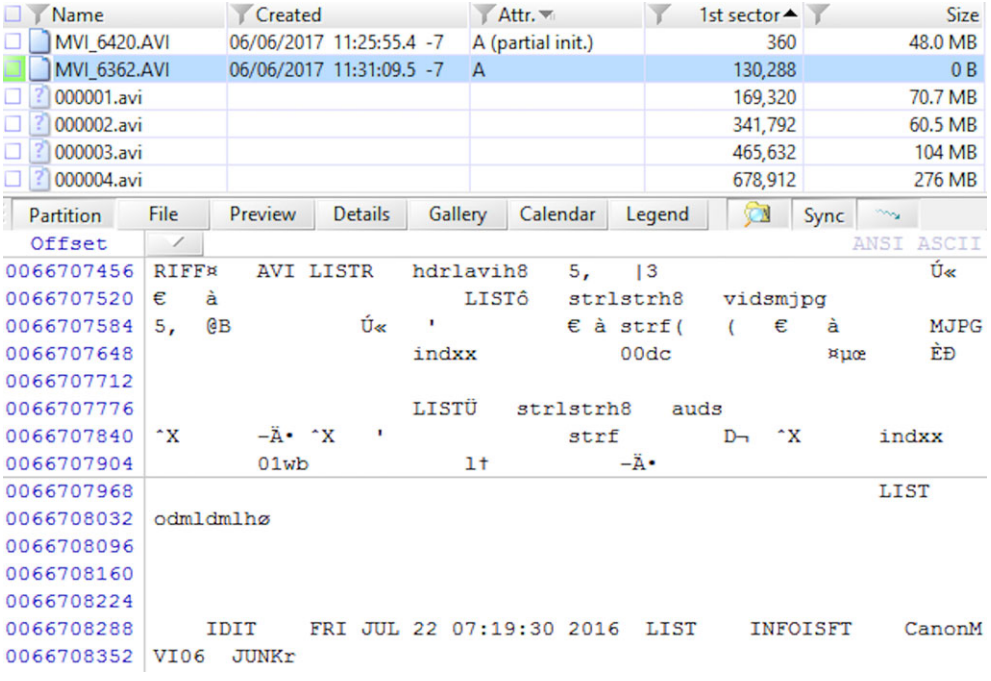


FIG. 10—*Two files that are partially initialized on NTFS shown in XWays Forensics 18. The first file (MVI_6420.AVI) is detected as partially initialized because its logical size is larger than its valid data length. The second file (MVI_6362.AVI) is not detected as initialized because no file size information is available. [Color figure can be viewed at wileyonlinelibrary.com]*

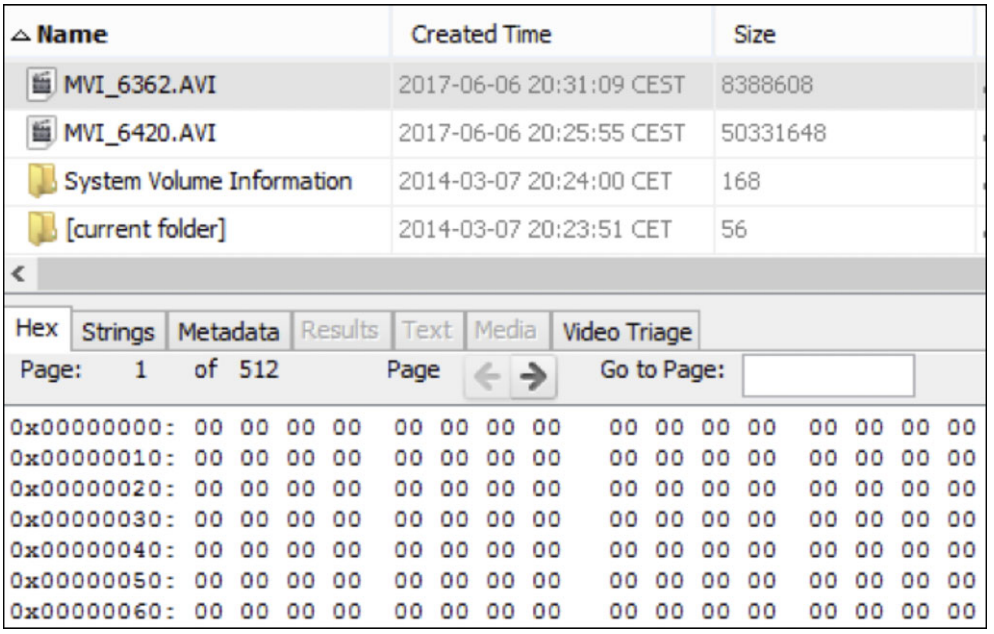


FIG. 11—*The same partially initialized file as Fig. 10, but viewed using Autopsy version 4.1.1. Zeroes are displayed instead of the actual cluster contents. The Autopsy graphical user interface (GUI) hides slack space by default to emulate what the Windows operating system presents. Autopsy version 4.3 and above have added an option to unhide data in slack space, presenting it as a separate file in the GUI. [Color figure can be viewed at wileyonlinelibrary.com]*

must be aware of such features in digital forensic software in order to obtain information needed for digital stratigraphy.

Given the challenges of forensic analysis when file initialization is involved, great care must be taken when reaching conclusions about the provenance of deleted information that is recovered from partially initialized files.

File Tunneling

When Microsoft Windows creates a new file that has the same name as a recently deleted file, the file system metadata from the old, deleted file is reassigned to the new file (18). This phenomenon is called file tunneling and can be replicated reliably on all versions of Windows, including Windows 10. File tunneling can result in newer content being stored in what appears to be an older file, which can be misinterpreted as backdating.

Case example: While providing expertise in a contract dispute, forensic analysts found what appeared to be backdating of a document that was pertinent to the case. Further analysis found that the discrepancy between the created date embedded within the document compared with the created date on the file system was due to file tunnelling resulting from normal use of Microsoft Word. Controlled experiments using Microsoft Word revealed that the file tunnelling phenomenon occurs when the 'Save As' menu item is selected to resave a file using the same name, effectively overwriting the file with the newest version of itself. Before the 'Save As' occurs, the created date embedded within the file matches the created date on the file system. After the file is modified and saved with the same name using the 'Save As' function, the embedded created date is updated but the created date on the file system remains unchanged due to file tunnelling. Table 1 compares the properties of the Word document before and after this Save As operation is performed, including the distinctive trace of the Revision number being set to 2.

TABLE 1—*Properties of an MS Word (Office version on Windows 8) document that has been resaved using the same name, effectively overwriting itself. File tunneling causes the new file to retain the old created date on the file system.*

Property Name	Before Save As	After Save As
File system created date	02/25/2015	02/25/2015
File system modified date	02/25/2015	6/5/2017
Embedded created date	02/25/2015	6/5/2017
Revision number	6	2
Editing time	5 min	0

File tunneling has also been observed in the context of Internet activities, with the contents of files being updated while retaining the old file system metadata. Without knowledge of file tunneling, temporal discrepancies between the file system metadata and dates in the cluster contents can lead to incorrect conclusions about digital stratigraphy.

Discussion

This work demonstrates the applicability of studying digital strata on storage media in forensic analysis to uncover the time frame, origin, composition, and distribution of each stratum. These novel approaches to conducting contextual analysis of file allocation traces, collectively called *digital stratigraphy*, translate concepts from stratigraphy in geology and archeology into the digital realm.

Among other insights, forensic practitioners and researchers can employ digital stratigraphy to discern concealment and destruction activities through a deeper understanding of file system behavior and the contextual analysis of file systems. When next-available file allocation strategies are exhibited, the

resulting traces can be utilized to infer their meaning and significance. However, when performing digital stratigraphy analysis, forensic practitioners must be aware that some traces of file allocation will not follow next-available allocation strategies. Furthermore, forensic practitioners must remember that apparent temporal discrepancies can be caused by normal file system behavior such as file initialization and file tunneling.

Digital stratigraphy can reveal traces of deletion when only metadata is recoverable or when no metadata for the deleted file content is recoverable. In addition, digital stratigraphy can uncover deleted data buried within a file system, such as in VDL slack. Digital stratigraphy can also be applied to recycled flash memory chips to differentiate activities of the current user from those of prior uses of the chip in previous, recycled devices (19).

There is a need for forensic tools to provide necessary information for contextual analysis, such as highlighting mass deletion, mass copying, VDL slack stratigraphy, file tunneling, and potential backdating. For example, in addition to plotting location information together with temporal details as shown in the present work, the Harris matrix approach could be adapted to digital stratigraphy (1). As another example, a proof of concept visualization tool developed to examine changes of file state (active vs. deleted) over time using information in Shadow Volumes demonstrates a form of digital stratigraphy that could be adapted to other file system metadata (20).

Other file systems need to be studied from a digital stratigraphy perspective, including exFAT, EXT4, and APFS. Further research is needed to develop novel digital stratigraphy methods that utilize the position of data on disk (e.g., scattered vs. concentrated), the origin of various fragments (e.g., from one source vs. many sources), or the composition of the data. New digital stratigraphy methods can be explored by performing controlled experiments to observe the file allocation traces that are created by specific actions.

Acknowledgments

I am grateful to Geoff Fellows for sharing his knowledge of file systems, for generating the example in Fig. 9, and most of all for his friendship.

References

1. Harris EC. Principles of archaeological stratigraphy, 2nd rev. edn. London, U.K./San Diego, CA: Academic Press, 1989.
2. Chabot Y, Bertaux A, Nicolle C, Kechadi T. A complete formalized knowledge representation model for advanced digital forensics timeline analysis. Digit Investig (Fourteenth Annual DFRWS Conference) 2014;11(2):S95–105.
3. Chabot Y, Bertaux A, Nicolle C, Kechadi T. An ontology-based approach for the reconstruction and analysis of digital incidents timelines. Digit Investig 2015;15:83–100.
4. Hales G. Assisting digital forensic analysis via exploratory information visualisation [dissertation]. Dundee, U.K.: Abertay University, 2016.
5. Casey E. Digital stratigraphy: analyzing file allocation methods to uncover concealment behavior. Proceedings of the 69th Annual Scientific Meeting of the American Academy of Forensic Sciences; 2017 Feb 13–18; New Orleans, LA. Colorado Springs, CO: American Academy of Forensic Sciences, 2017.
6. Microsoft. How FAT works. Microsoft product documentation, 2003; https://technet.microsoft.com/en-us/library/cc776720.aspx.
7. Microsoft. FAT file system. Microsoft product documentation, 2010; http://technet.microsoft.com/en-us/library/cc938438.aspx.
8. Microsoft. Fatfs, 2011; http://research.microsoft.com/en-us/um/redmond/projects/invisible/src/storage/fatfs/fatfs.c.htm.
9. Minnaard W. The Linux FAT32 allocator and file creation order reconstruction. Digit Investig 2014;11(3):224–33.
10. Carrier B. File system forensic analysis. Upper Saddle River, NJ: Addison-Wesley, 2005.
11. Tse WHK. Forensic analysis using FAT32 cluster allocation patterns [Master's thesis]. Hong Kong, China: University of Hong Kong, 2011.
12. Libananco Holdings Co. Ltd. v. Republic of Turkey. Case No. ARB/06/8. Washington, DC: International Centre for Settlement of Investment Disputes, 2001;
13. Friedberg E. To cache a thief: how litigants and lawyers tamper with electronic evidence and why they get caught. The American Lawyer 2004.
14. Lee WY, Kwon H, Lee H. Comments on the Linux FAT32 allocator and file creation order reconstruction. Digit Investig 2015;15:119–23.
15. Casey E. Digital evidence and computer crime: forensic science, computers, and the internet. 3rd rev. edn. Waltham, MA: Academic Press, 2011.
16. Ferguson D. Redefining file slack in Microsoft NTFS. J Digit Forensic Pract 2008;2(3):140–56.
17. Microsoft. Fsutil file. Microsoft product documentation, 2012; https://technet.microsoft.com/en-us/library/bb490642.aspx, update https://technet.microsoft.com/en-us/library/cc788058.aspx.
18. Microsoft. Windows NT contains file system tunneling capabilities. Microsoft Knowledge Base Article 172190, 2008; https://support.microsoft.com/en-us/help/172190/windows-nt-contains-file-system-tunneling-capabilities.
19. Westman M. eMMC Chip off – benefits and risks workshop. Digit Investig (DFRWS EU 2017).
20. Leschke TR, Nicholas C. Change-link 2.0: a digital forensic tool for visualizing changes to shadow volume data. In: Goodall J, Ma K-L, Engle S, Fischer F, editors. VizSec 2013: Proceedings of the Tenth Workshop on Visualization for Cyber Security; 2013 Oct 14; Atlanta, GA. New York, NY: AMC, 2013.

Additional information and reprint requests:
Eoghan Casey, Ph.D.
Ecole des Sciences Criminelles (ESC)
Université de Lausanne
Batochime CH-1015
Lausanne-Dorigny
Switzerland
E-mail: eoghan.casey@unil.ch