



Audio Engineering Society Conference Paper

Presented at the Conference on Audio Forensics
2017 June 15–17 Arlington, VA, USA

This paper was peer-reviewed as a complete manuscript for presentation at this conference. This paper is available in the AES E-Library (<http://www.aes.org/e-lib>) all rights reserved. Reproduction of this paper, or any portion thereof, is not permitted without direct permission from the Journal of the Audio Engineering Society.

Deleted Audio File Decay on a Digital Voice Recorder

James H. Jones, Jr.¹

¹ George Mason University, Department of Electrical and Computer Engineering, Fairfax, Virginia, US

Correspondence should be addressed to Jim Jones (jjonesu@gmu.edu)

ABSTRACT

Recovery of intact but previously deleted digital files is a staple of data recovery services and digital forensics investigations. Although more challenging, the recovery and analysis of fragments from deleted files remains an active area of research and application. In this work, we study the decay of MP3 formatted audio files on a portable voice recorder with internal flash storage, and we recover usable audio fragments. Our approach captures media snapshots over time, then analyzes them to measure deleted file decay. Our results in this work and elsewhere indicate that fragments of deleted files are (a) surprisingly persistent, (b) recoverable, and (c) interpretable. This work is relevant for law enforcement, the intelligence community, and data recovery specialists.

1 Introduction

The contents of digital files are usually not destroyed at the time of file deletion. Rather, a filesystem reference to the file is changed or deleted, and the storage locations used for the file's data are marked as free for future use. Over time, new files are written to the media, and these new files may overwrite part or all of the deleted file. The effect on an arbitrary deleted file is one of monotonic decay over time.

Fragments of deleted files may contain useful information such as email addresses, URLs, or account numbers, may contain recoverable imagery or audio, and may indicate the past presence of a specific file. Recovery and analysis of deleted file fragments may prove useful in criminal investigations, intelligence activities, or when the recovered fragment otherwise contains some data of value to a user or analyst.

Despite the acknowledged value of deleted file fragment contents, and ongoing work inferring

source file type and possibly identity, little research has focused on the factors affecting deleted file decay. Such research might facilitate the optimal allocation of scarce investigative resources ("should we bother looking for fragments of file X? should we expect to find any, and if so, how many?") and support the interpretation of the presence or absence of fragments under different circumstances ("what does it mean if we don't find fragments of file X, or only find N fragments, or find them in a particular pattern?").

2 Related Work

Deleted file recovery, and related efforts to carve unknown but intact files from digital media, is straightforward when the file's contents are contiguous [1]. Given known file headers, and either known trailers or file size information from metadata, intact deleted files may be recovered. Similarly, in the absence of end of file indicators, trial and error has been shown to work in many cases, essentially appending sequential sectors to the recovered start of file data until rendering errors cease, past which point errors recur. Recovery of

intact but non-contiguously stored files creates a challenge which has been solved for some file types where file continuation can be reasonably inferred from sector contents [2].

Subsector string recovery is also straightforward, as these may be extracted with no knowledge of the source file contents or location. It is the middle ground of file fragment recovery and interpretation that is more difficult, where a file fragment consists of one or more sectors from the original file, but not the entire file. Prior work over the last 10 years has attempted to identify the filetypes of recovered file fragments [3][4][5][6], and on methods to establish that a remnant is part of a specific original file vs. a false positive, a situation which is discussed in [7] and [8].

The efforts noted above support the importance of deleted file content recovery at any level, but they do not address the mechanisms by which file fragments are created nor the factors affecting deleted file decay. In 2012, Fairbanks and Garfinkel [9] posited factors which might affect the persistence of deleted file content. This paper is predicated by other work observing the effects of data persistence but not attempting to explain it beyond the immediate case. In chapter 7 of their 2005 book "Forensic Discovery", Farmer and Venema [10] published experimental data and partial explanations for the persistence of deleted file information. While useful, the experiments were limited in scope, and the discussion sought to explain the observed persistence given aspects of their particular test system rather than computer systems in general. As noted by Venema elsewhere [11], "...persistence of deleted file content is dependent on file system, activity, and amount of free space (a complex relationship)." Roussev and Quates [12] tangentially show the data persistence behaviour in a case study of the M57 dataset [13]. The case study focused on content triage using similarity digests, but the paper includes a graph of deleted file data persistence for a specific example (see Figure 1 on page S66 of that

publication). The decay effect, observed as an almost linear reduction in deleted file content over a matter of days, is explained by the user deleting the files and continuing to use the system normally.

In this work, we begin to study the persistence of deleted audio files on embedded flash media in a controlled environment. It is our aim to begin to understand the mechanisms by which deleted file content decays, and the factors affecting the rate and nature of that decay.

3 Methodology

Our method for studying deleted file decay captures sequential media snapshots over time, then processes those images to track the contents of locations where deleted file contents are stored [14]. Assuming that files to be tracked are deleted between the first two images, we identify deleted files as those that are present in the first image but not the second. Then we record the sector locations and contents of those files in the first image, prior to deletion. We check the contents of the original file's sectors over the remaining images, identifying if, when, and which sectors of each deleted file were changed. This approach was inspired by the differential analysis ideas articulated by Garfinkel et al [15]. The file and sector change data is then processed to form and test hypotheses as to which factors affect deleted file decay, and to design additional experiments. This process is summarized in Figure 1.

We conducted experiments using a Sony ICD-PX440 digital voice recorder. The device has 4GB of internal flash storage (3.63GB available for audio file storage), formatted FAT32 with 512 byte sectors and 64 sectors per cluster. Prior to the experiments reported here, we fully reformatted the device to ensure the media did not contain remnants of any prior recordings.

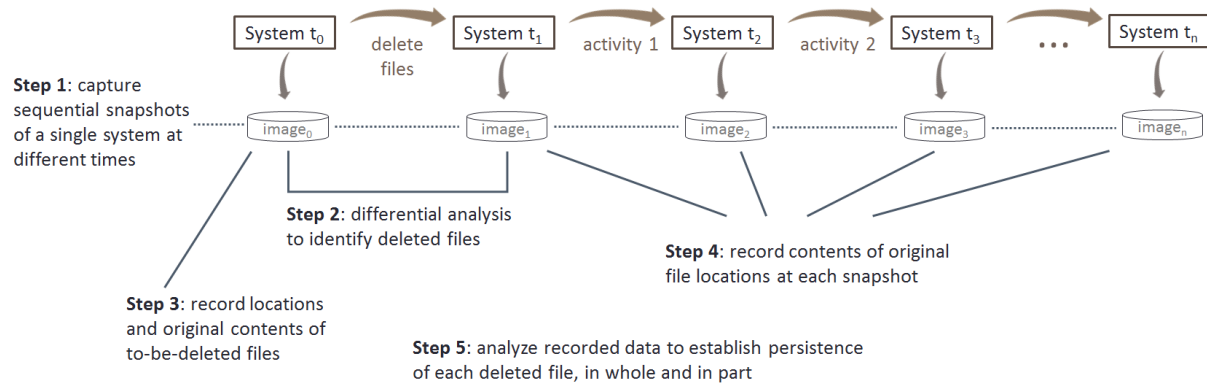


Figure 1. Deleted file decay experimental process.

We made 10 audio recordings on the device, a mix of voice and music, although both used external sources (i.e., we didn't copy music files to the device, but rather played music from another device and recorded the sound using the device's microphone). The audio was recorded using the ICD-PX440's Super High-Quality (SHQ) monaural mode with a sampling rate of 44.1kHz and bit rate of 192kbps. The files ranged in size on media from 177 KB to 1.3MB, in length from 7 to 54 seconds, and were stored as MP3 files on the internal flash storage. Viewing the files' metadata using ffprobe (<https://ffmpeg.org/ffprobe.html>) indicates encoding by SONY IC RECORDER MP3 5.1.2 and 44100 Hz, stereo, s16p, and 192 kb/s. We captured the full contents of the internal flash storage using FTK Imager, mounting the media as a physical drive using the device's built-in USB interface.

We captured an initial snapshot (Image ID 0) after creating the 10 audio files, then deleted the 10 files and captured another snapshot (Image ID 1). We continued different activities and captured snapshots as summarized in Table 1.

The 3GB of files written to the device in step 5 consisted of 2845 PNG files of size 12KB each, 39 MPG files of size 7MB each, and 39 AVI files of size 73MB each.

Image ID	Activity
0	after creating 10 audio files
1	after deleting those 10 files manually (via USB connection to PC)
2	after creating nine new audio files
3	after creating seven additional audio files
4	after deleting the newly created (all) files on the device
5	after loading files on the device manually (3GB; to near device capacity)
6	after (quick) reformatting the device storage

Table 1. Device activity and snapshots.

4 Results

Processing the 7 snapshots to trace the decay of the 10 initially deleted audio files generated the decay curves shown in Figure 2, where the y-axis is percent of the file (sectors) intact, the x-axis is the image ID with time flowing from left to right, and each file is represented by one line in the graph. In summary, four of the 10 files had surviving sectors

through the final snapshot (image ID 6). Surviving sectors for three of the four files were within the main body of the original file (neither at the beginning or end of the original file); surviving sectors for the fourth file were near the beginning of the original file and did not contain audio content.

The sharp decay rates between Image ID 1-2 and Image IDs 2-3 align with creating new recordings on the device, where the combined sizes of the 16 (nine + seven) new files exceeded the combined sizes of the original 10 files. Similarly, filling the device's storage media between Image IDs 4-5 contributed further to deleted file decay, while deleting the new files between Image IDs 3-4 and quick reformatting the device between Image IDs 5-6 had no effect on the original file decay. Table 2 (at the end of the paper) contains detailed file persistence data for the 10 deleted files. Recalling that the device media is formatted with a 32 KB cluster size (64 sectors), the surviving sectors can be explained as residing in the cluster (file) slack of the files written after the original files were deleted. That is, the new files did not entirely overwrite the deleted files, but the residual data now resided in an allocated cluster and was protected from further decay. In fact, this was manually confirmed by inspecting the raw media images directly. For example, the boundaries of the 22 surviving sectors of file 2 are shown in Figure 3 and Figure 5. In Figure 3, the first surviving sector of file 2 is shown as it exists in Image ID 6, beginning at offset 0x06975400 and sector 215978, which is not a cluster boundary. In Figure 5, the last surviving sector of file 2 is shown also as it exists in Image ID 6, ending at offset 0x06977FFF and just before the start of sector 216000, which is a cluster boundary (note the start of a clearly new file, a PNG image, at the start of sector 216000). Audio content in an MP3 file is stored in frames, where only complete frames are playable [16]. The MP3 frame header is 0xFFFBB204, which begins 63 bytes into the recovered sectors for file 2 (see Figure 3), and the last complete header ends 31 bytes before the end of the second to last recovered sector (see Figure 4). When replaying recovered audio we will only hear complete frames, i.e., audio from just under 21 of the 22 recovered sectors.

Given that encoded data from different sources will rarely produce files of exactly the same size or aligning with cluster boundaries, the end of an arbitrary allocated file will usually coincide with an internal and full cluster of a previously deleted file. In essence, the final cluster of every allocated file on the media will likely have residual data from a previously deleted file.

Individual file survivability is highly dependent on the nature of the files that are written to the device after deletion and the order in which they are written. For example, in our experiment step 5, we wrote many small files (12KB each) to the media first, then wrote larger files (7MB then 73MB). Had a single large file (greater than 5.5MB) been written directly over the 10 deleted audio files (a total of 5.5 MB), then no file slack would exist in which the deleted file content could survive. We are assuming in most cases that the deleted and overwriting files are of similar but not identical size and the media was previously full, so the file slack of each newly written file is likely to have the remnants of some previously deleted file.

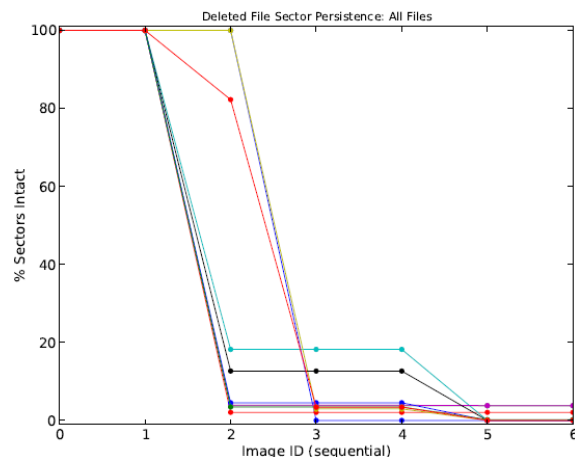


Figure 2. Decay curve for 10 deleted audio files.

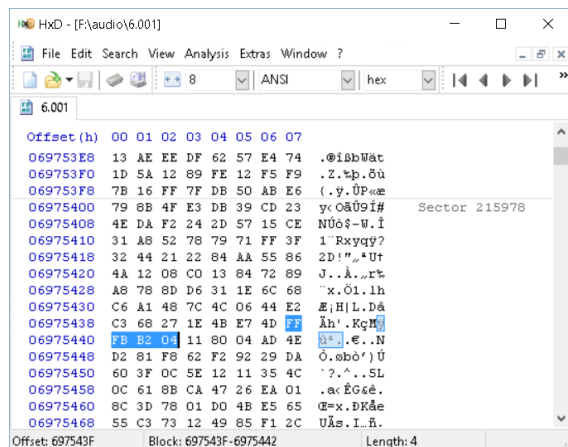


Figure 3. Start of surviving file 2 sectors.

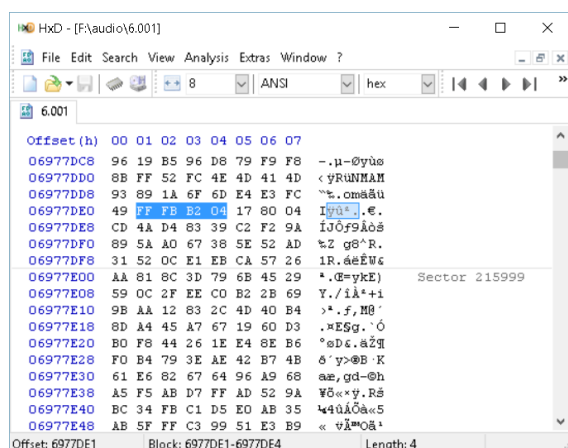


Figure 4. End of playable file 2 sectors.

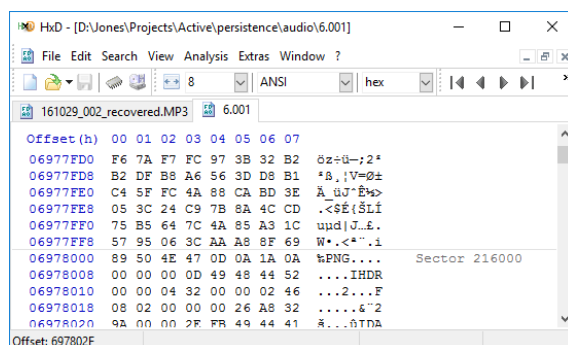


Figure 5. End of surviving file 2 sectors.

We extracted the surviving sectors from each of the four files, and saved the extracted sectors as files with MP3 extensions. We were able to play three of the four resulting audio files as MP3 files (files 1, 2, and 3 in Table 2). The surviving fragments of the fourth file (file 5 in Table 2) did not contain audio content, originally or in their recovered and unchanged state. The three original files and recovered audio extracts are available on GitHub at:

<https://github.com/jjonesu/DeletedFilePersistence/tree/master/audio>

5 Conclusions

Fragments of deleted audio files (MP3) persist on the embedded flash storage media of audio recording devices after deletion, even after multiple overwrites. This persistence is driven in part by file system parameters such as sector and cluster size, and the relationship between these parameters and the sizes of files written to the device's media. These fragments may be recovered and may contain intelligible audio. The potential to recover portions of previously deleted audio recordings long after deletion has implications for privacy, law enforcement, the intelligence community, and data recovery specialists.

Future work will extend these experiments to include additional devices, configuration parameters, and post-deletion activity in order to establish factors affecting deleted file decay and potential inferences to be drawn from the nature and patterns of residual deleted file contents.

file	sectors	sectors intact at snapshot:							final persistence	original length (sec)	surviving length (sec)
		0	1	2	3	4	5	6			
0	346	346	346	63	63	63	0	0	0.00%	7	0.00
1	746	746	746	28	28	28	28	28	3.75%	15	0.56
2	1065	1065	1065	22	22	22	22	22	2.07%	22	0.45
3	1055	1055	1055	41	41	41	40	40	3.79%	22	0.83
4	1435	1435	1435	65	65	65	0	0	0.00%	30	0.00
5	2554	2554	2554	88	88	88	5	5	0.20%	54	0.11
6	395	395	395	50	50	50	0	0	0.00%	8	0.00
7	1387	1387	1387	1141	47	47	0	0	0.00%	29	0.00
8	611	611	611	611	0	0	0	0	0.00%	12	0.00
9	1665	1665	1665	1665	49	49	0	0	0.00%	35	0.00

Table 2. Deleted file decay data.

References

- [1] Richard III, G. G., & Roussev, V. (2005, August). Scalpel: A Frugal, High Performance File Carver. In DFRWS.
- [2] Pal, A., Shanmugasundaram, K., & Memon, N. (2003, July). Automated reassembly of fragmented images. In Multimedia and Expo, 2003. ICME'03. Proceedings. 2003 International Conference on (Vol. 1, pp. I-625). IEEE.
- [3] Li, B., Wang, Q., & Luo, J. (2006, December). Forensic analysis of document fragment based on SVM. In Intelligent Information Hiding and Multimedia Signal Processing, 2006. IIH-MSP'06. International Conference on (pp. 236-239). IEEE.
- [4] Calhoun, W. C., & Coles, D. (2008). Predicting the types of file fragments. Digital Investigation, 5, S14-S20.
- [5] Beebe, N. L., Maddox, L. A., Liu, L., & Sun, M. (2013). Scedan: Using concatenated n-gram vectors for improved file and data type classification. IEEE Transactions on Information Forensics and Security, 8(9), 1519-1530.
- [6] Roussev, V., & Quates, C. (2013). File fragment encoding classification—An empirical approach. Digital Investigation, 10, S69-S77.
- [7] Garfinkel, S., Nelson, A., White, D., & Roussev, V. (2010). Using purpose-built functions and block hashes to enable small block and sub-file forensics. digital investigation, 7, S13-S23.
- [8] Garfinkel, S. L., & McCarrin, M. (2015). Hash-based carving: Searching media for complete files and file fragments with sector hashing and hashdb. Digital Investigation, 14, S95-S105.
- [9] Fairbanks, K., & Garfinkel, S. (2012). Column: Factors Affecting Data Decay.

- Journal of Digital Forensics, Security and Law, 7(2), 7-10.
- [10] Farmer, D., & Venema, W. (2005). Forensic discovery (Vol. 6). Upper Saddle River: Addison-Wesley.
 - [11] Reust, J., & Friedburg, S. (2006). DFRWS 2005 Workshop Report.
 - [12] Roussev, V., & Quates, C. (2012). Content triage with similarity digests: The M57 case study. Digital Investigation, 9, S60-S68.
 - [13] Garfinkel, S., Farrell, P., Roussev, V., & Dinolt, G. (2009). Bringing science to digital forensics with standardized forensic corpora. digital investigation, 6, S2-S11.
 - [14] Jones, J., & Khan, T. (2017). A Method and Implementation for the Empirical Study of Deleted File Persistence in Digital Devices and Media. Proceedings of the 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC).
 - [15] Garfinkel, S., Nelson, A. J., & Young, J. (2012). A general strategy for differential forensic analysis. Digital Investigation, 9, S50-S59.
 - [16] Koenig, B. E., Lacey, D. S., & Reimond, C. E. (2014). Selected Characteristics of MP3 Files Re-encoded With Audio Editing Software. Journal of Forensic Identification, 64(3), 304.