The Impact of SHA-1 File Hash Collisions On Digital Forensic Imaging: A Follow-up Experiment

Gary C. Kessler
Emory Riddle Aeronautical University - Daytona Beach

Follow this and additional works at: https://commons.erau.edu/jdfsl

Part of the Computer Engineering Commons, Computer Law Commons, Electrical and Computer Engineering Commons, Forensic Science and Technology Commons, and the Information Security Commons

Recommended Citation
Kessler, Gary C. (2016) "The Impact of SHA-1 File Hash Collisions On Digital Forensic Imaging: A Follow-up Experiment," Journal of Digital Forensics, Security and Law. Vol. 11 , Article 10.
DOI: https://doi.org/10.15394/jdfsl.2016.1433
Available at: https://commons.erau.edu/jdfsl/vol11/iss4/10

This Article is brought to you for free and open access by the Journals at Scholarly Commons. It has been accepted for inclusion in Journal of Digital Forensics, Security and Law by an authorized administrator of Scholarly Commons. For more information, please contact commons@erau.edu.
THE IMPACT OF SHA-1 FILE HASH COLLISIONS ON DIGITAL FORENSIC IMAGING: A FOLLOW-UP EXPERIMENT

Gary C. Kessler
Embry-Riddle Aeronautical University
Daytona Beach, Florida
386-226-7947
gary.kessler@erau.edu

ABSTRACT

A previous paper described an experiment showing that Message Digest 5 (MD5) hash collisions of files have no impact on integrity verification in the forensic imaging process. This paper describes a similar experiment applied when two files have a Secure Hash Algorithm (SHA-1) collision.

Keywords: SHA-1 hash collisions, forensic imaging, computer forensics, digital forensics

INTRODUCTION

An earlier paper (Kessler, 2017) discussed the impact on the hash value of two disk images that contain the same set of files except for one -- one file that has the same Message Digest 5 (MD5) hash value as another file of the same size but different content. That paper showed that the resulting disk image hash values were, in fact, different even though all of the component files and spaces on the disk had the same hash.

That paper was specific to MD5 hash collisions. As it was coming to press, Stevens, Bursztein, Karpman, Albertini, and Markov (2017) announced a SHA-1 hash collision between two files of the same size with different content.

This paper will use the same methodology as the earlier paper to address the impact of SHA-1 hash collisions on validating the results of the computer forensics imaging process.

Section 2 will state the research question. Section 3 will describe the experimental framework with which to test the research hypothesis, followed by test results in Section 4. Section 5 will offer conclusions.

2. RESEARCH QUESTION

The earlier paper (Kessler, 2017) described a scenario that can be summarized as follows: Suppose we have two files, A and B, that have different content but are the same size and have the same SHA-1 hash value. What is the effect on the hash value of two disk images that differ only in that one disk contains File A and the other disk contains File B (where Files A and B occupy the same location on the two disk images)? SHA-1 is described in Eastlake and Jones (2001) and NIST (2015).

The research question is to test the following null hypothesis (H₀) as follows:

- The resultant two disk images will have the same hash value.
The alternative hypothesis \( (H_1) \) is as follows:

- The resultant two disk images will have different hash values.

3. EXPERIMENTAL SETUP

![Figure 1. shattered-1.pdf (left) and shattered-2.pdf (right).](image)

Examined in a hex editor, these files are found to be different in 92 nibbles (in 62 bytes), all within a single 128-byte block starting at byte offset 0x00C0; the differences are indicated by the **bolded** nibbles below:

```
hash1.bin (from shattered-1.pdf)
0000000C: 73 46 DC 91 66 B6 7B 11 BF 02 9A B6 21 B2 56 0F
000000DD: F9 CA 67 CC A9 C7 F8 5B A8 4C 79 03 0C 2B 3D E2
000000F0: 1B F8 6D B3 A9 09 01 D5 DF 45 C1 4F 26 PF DF E3
000000F0: DC 38 E9 6A C2 2F E7 BD 72 8F 0E 45 BC E0 46 D2
00000100: 3C 57 0F EB 14 13 98 BB 55 2E F5 A8 A2 2B E3 31
00000110: FE A4 80 37 B8 B5 D7 1F 0E 33 2E DF 93 AC 35 00
00000120: EB 4D DC 0D EC C1 A8 64 79 0C 7C 2C 76 21 56 60
00000130: DD 30 97 91 D0 6B 0D AF 3F 98 CD A4 BC 46 29 B1
hash2.bin (from shattered-2.pdf)
0000000C: 7F 46 DC 93 A6 B6 7E 01 3B 02 9A AA 1D B2 56 0B
000000DD: 45 CA 67 D6 88 C7 F8 4B 8C 4C 79 1F E0 2B 3D F6
000000F0: 14 F8 6D BI 69 09 01 CI 6B 45 C1 53 0A FE DF E7
000000F0: 60 38 E9 72 72 2F E7 AD 72 8F 0E 49 04 E0 46 C2
00000100: 3D 57 0F E9 DC 13 98 AB B1 2F FB 94 2B E3 35
00000110: 42 A4 80 2D 9B B5 D7 0F 2A 33 2E C4 7F AC 35 14
00000120: E7 4D DC 0F 2C C1 A8 74 CD 0C 78 30 5A 21 56 64
00000130: 61 30 97 89 60 6B 0D BF 3F 98 CD A8 04 46 29 A1
```

The `fc` (file compare) command confirms these differences. Details of this comparison, including the 150 bits that differ, can be found in Appendix 1.

As shown below, the two files have the same 160-bit SHA-1 hash, although their 128-bit MD5 hash values differ. This confirms that the contents of the two files are actually different and that there is a bona fide SHA-1 hash collision:

File: shattered-1.pdf
MD5: EE4AA52B139D959B91D1C6D8D8884402B0A750C
SHA: 38762CF7F55934B34D179A6A4C80CADCCBB7F0A

To address the research questions, two files were needed that were the same size, had the same SHA-1 hash, and had different content. Centrum Wiskunde & Informatica (2017) provides such a pair of 422,435-byte files, called `shattered-1.pdf` and `shattered-2.pdf` (Figure 1).
4. TESTS AND RESULTS

Four tests were conducted on the media described above. The results described in this section are summarized in Table 1.

In Test #S1, the thumb drive was imaged using FTK Imager (v3.1.3.2). The purpose of this test was merely to prepare a baseline disk image and set of hash values. The image verification SHA-1 hash of the thumb drive was 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18 -- the same as in Test #S1. This result confirms that overwriting data in this way is an adequate process and changes nothing else on the drive. A portion of the FTK Imager report can be found in Appendix 3. The FTK file listing showed that hash1.pdf had the expected MD5 and SHA-1 hash values for the shattered-1.pdf file.

For Test #S3, the thumb drive was mounted in WinHex and the contents of hash2.bin were copied over the 128-byte "collision block" where hash1.pdf resided on the thumb drive, thus creating the shattered-2.pdf file. This test was really the crux of the hypothesis experiment since hash1.pdf now contained the "hash-equivalent, content-different" file. The thumb drive was re-imaged, yielding an image verification SHA-1 hash of a00b80e17de1677d34d21c6e53ff9e0603eadbe6 -- different than Tests #S1 and #S2. A portion of the FTK Imager report can be found in Appendix 4. The FTK file listing showed
that *hash1.pdf* had the expected MD5 and SHA-1 hash values for the *shattered-2.pdf* file.

For Test #S4, the thumb drive was mounted with WinHex and the contents of *hash1.bin* were copied back over the "collision block" where *hash1.pdf* resided on the thumb drive, now recreating the *shattered-1.pdf* file. The purpose of this test was to restore the drive to its original state and confirm that Test #S3 changed nothing more than the 128 bytes where the test data resided. The image verification SHA-1 hash was 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18 -- the same as in Tests #S1 and #S2. This result confirms that Test #S4 had restored the disk to its initial state and that Test #S3 changed nothing more than the file data. A portion of the FTK Imager report can be found in Appendix 5. The FTK file listing showed that *hash1.pdf* had the expected MD5 and SHA-1 hash values for the *shattered-1.pdf* file.

### 5. CONCLUSIONS

The image verification SHA-1 hashes in Tests #S1, #S2, and #S4 -- images that each held the *shattered-1.pdf* (*hash1.bin*) content -- had the same value, whereas the image verification SHA-1 hash value in Test #S3 -- when the image held the *shattered-2.pdf* (*hash2.bin*) content -- was different from the other tests. The fact that Tests #S1, #S2, and #S4 had the same hash proved that the test process worked as desired; the fact that Test #S3 had a different result shows that the hash value of the imaged drive depends upon the actual bit content of the entire drive. Since the hash values of the two images are not the same, the null hypothesis (H₀) is disproven and the alternate hypothesis (H₁) is proven.

| Description of Test                                                                 | Image SHA-1 Hash Value                      |
|-------------------------------------------------------------------------------------|---------------------------------------------|
| #S1 - Drive with *shattered-1.pdf*                                                  | 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18   |
| #S2 - Overwrite bytes 0x8490C0-0x84913F with *hash1.bin* data (*shattered-1.pdf)* | 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18   |
| #S3 - Overwrite bytes 0x8490C0-0x84913F with *hash2.bin* data (*shattered-2.pdf)* | a00b80e17de1677d34d21c6e53ff9e0603eadbe6   |
| #S4 - Overwrite bytes 0x8490C0-0x84913F with *hash1.bin* data (*shattered-1.pdf)* | 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18   |

As in the prior paper, disproving the null hypothesis is the expected result because the hash value of a disk image is based upon the bit contents of the disk rather than the hashes of the individual files -- including file system structures and unallocated space -- that compose the disk contents. Thus, even if all of the file hashes on two disks are the same, the disk image hashes will be different if the contents of the files are different. Given this result, the scenario described in Section 2 cannot be realized.

It is hoped that this result will lay the concern about file hash collisions to rest as they apply to digital forensic imaging. As long as both individual files and the entire image are hashed, the theoretical occurrence of individual file collisions is not a factor in confirming the evidentiary integrity of a forensic copy.

This said, the fact that SHA-1 collision can be forced is significant. Although the SHA-1 standard was deprecated in 2013, it is still in wide use.
As noted in the prior paper, the MD5 hash values are different for the\textit{shattered-1.pdf} and\textit{shattered-2.pdf} files, although the SHA-1 hash value is the same. Since the MD5 and SHA-1 algorithms are different, the manipulation that can create an MD5 collision cannot create a SHA-1 collision -- indeed, note the complexity of the SHA-1 collision compared to the relative simplicity of the MD5 collision. To date, no one has yet shown a practical method with which to cause both an MD5 and SHA-1 collision in the same file.

**NOTE**

All FTK Imager reports, FTK reports, and ancillary files are available for examination at http://www.garykessler.net/gck/sha_test.zip.

**AUTHOR BIOGRAPHY**

Gary C. Kessler, Ph.D., is a professor of cybersecurity and chair of the Security Studies & International Affairs Department at Embry-Riddle Aeronautical University in Daytona Beach, Florida. He is a Certified Computer Examiner (CCE), Certified Cyber Forensics Professional (CCFP), and Certified Information Systems Security Professional (CISSP), and a member of the Hawaii and North Florida Internet Crimes Against Children (ICAC) Task Force. Additional information can be found at http://www.garykessler.net.
REFERENCES

Centrum Wiskunde & Informatica (CWI). (2017). Shattered. Retrieved from https://shattered.it/

Eastlake, D., 3rd, & Jones, P. (2001, September). US Secure Hash Algorithm 1 (SHA1). Requests for Comments (RFC) 3174. Retrieved from https://www.rfc-editor.org/rfc/rfc3174.txt

Kessler, G.C. (2017). The Impact of MD5 File Hash Collisions on Digital Forensic Imaging. *Journal of Digital Forensics, Security & Law, Vol. 11: No. 3*, pp. 129-140.

National Institute of Standards and Technology (NIST). (2015, August). *Secure Hash Standard (SHS)*. Federal Information Processing Standards Publication FIPS PUB 180-4. Retrieved from http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

Stevens, M., Bursztein, E., Karpman, P., Albertini, A., & Markov, Y. (2017). The first collision for full SHA-1. Retrieved from https://shattered.it/static/shattered.pdf
APPENDICES

Appendix 1: Comparison of 128-Byte Difference of the Two Files

Comparing files shattered-1.pdf and shattered-2.pdf

| Hash difference | File 1 | File 2 | Difference |
|-----------------|--------|--------|------------|
| 0000000C: 73 7F | 01110011 | 01111111 |             |
| 0000000C: 91 93 | 10100001 | 10100011 |             |
| 0000000C: 66 A6 | 01101110 | 10100110 |             |
| 0000000C: 11 01 | 00010001 | 00000001 |             |
| 0000000C: 8F 3B | 10001111 | 00111011 |             |
| 0000000C: B6 AA | 10110110 | 10101010 |             |
| 0000000C: 21 1D | 01000001 | 00111001 |             |
| 0000000C: F9 45 | 11111101 | 01001011 |             |
| 0000000D: CC D6 | 11001100 | 11010110 |             |
| 0000000D: A8 88 | 10101000 | 10010000 |             |
| 0000000D: 5B 4B | 01011011 | 01001011 |             |
| 0000000D: A8 8C | 10101000 | 10001100 |             |
| 0000000D: 03 1F | 00000011 | 00011111 |             |
| 0000000D: 0C E0 | 00001100 | 11110000 |             |
| 0000000D: E2 F6 | 11100100 | 11110110 |             |
| 0000000E: 18 14 | 00011000 | 00010100 |             |
| 0000000E: B3 B1 | 10110011 | 10110001 |             |
| 0000000E: A9 69 | 10101001 | 01101001 |             |
| 0000000E: D5 C5 | 11010101 | 11000101 |             |
| 0000000E: DF 6B | 11011111 | 01101011 |             |
| 0000000E: 4F 53 | 01001111 | 01010011 |             |
| 0000000E: 26 0A | 00101010 | 00010010 |             |
| 0000000E: B3 B7 | 10110011 | 10110111 |             |
| 0000000F: DC 60 | 11011100 | 11000000 |             |
| 0000000F: 6A 72 | 01101010 | 01101000 |             |
| 0000000F: C2 72 | 11000010 | 01100010 |             |
| 0000000F: BD AD | 10111011 | 10101011 |             |
| 0000000F: 45 49 | 01001001 | 01001001 |             |
| 0000000F: BC 04 | 10111100 | 00001000 |             |
| 0000000F: D2 C2 | 11010010 | 11000100 |             |
| 0000000F: 3C 30 | 00111100 | 00110000 |             |
| 0000000F: 4B 99 | 11101011 | 11101001 |             |
| 0000000F: D4 6D | 00010100 | 11020100 |             |
| 0000000F: BB AB | 10111011 | 10101111 |             |
| 0000000F: 55 E1 | 01010101 | 11100001 |             |
| 0000000F: A0 BC | 10100000 | 10111100 |             |
| 00000010: 9D 94 | 10101000 | 10010100 |             |
| 00000010: 31 35 | 01100001 | 01100101 |             |
| 00000010: 4F 42 | 11111110 | 01000010 |             |
| 00000010: 37 2D | 00101111 | 00101101 |             |
| 00000010: 8B 98 | 10111000 | 10100000 |             |
| 00000010: 17 0F | 00011111 | 00001111 |             |
| 00000010: 0E 2A | 00001110 | 00101010 |             |
| 00000010: DF C3 | 11011111 | 11000011 |             |
| 00000010: 93 7F | 10010011 | 01111111 |             |
| 00000010: F0 14 | 00000000 | 00010100 |             |
| 00000010: EB B7 | 11101011 | 11101111 |             |
| 00000010: 0D 0F | 00001101 | 00001111 |             |
| 00000010: EC 2C | 11101110 | 00101000 |             |
| 00000010: 2A 7A | 01101010 | 01101000 |             |
| 00000010: 2C 30 | 00101100 | 00100000 |             |
| 00000010: 7A 5A | 01101010 | 01101010 |             |
| 00000010: 60 64 | 01100000 | 01100100 |             |
| 00000010: DD 61 | 11011110 | 01100001 |             |
| 00000010: 91 89 | 10010001 | 10001101 |             |
| 00000010: BC A0 | 10101000 | 10101000 |             |
| 00000010: BC 04 | 10111110 | 00000100 |             |

© 2016 ADFSL
Although beyond the scope of this paper, a pattern emerges when looking at the bytes bit-by-bit. The following table shows the values of the 128-byte "difference" block when the two files are Exclusively-ORed (XOR) together; a 0 indicates bits that are the same in the two blocks and a 1 indicates bits that are flipped:

| Offset | Hex | Binary       |
|--------|-----|--------------|
| 00C0   | 00001100 00000000 00000000 00000010     |
| 00D0   | 10111100 00000000 00000000 00011010     |
| 00E0   | 00001100 00000000 00000000 00000010     |
| 00F0   | 10111100 00000000 00000000 00011000     |
| 00C4   | 00000000 00000000 00000000 00000000     |
| 00D4   | 00100000 00000000 00000000 00010000     |
| 00E4   | 11100000 00000000 00000000 00000000     |
| 00F4   | 10110000 00000000 00000000 00010000     |
| 00C8   | 01110100 00000000 00000000 00011100     |
| 00D8   | 00100100 00000000 00000000 00011100     |
| 00E8   | 01110100 00000000 00000000 00011100     |
| 00F8   | 00000000 00000000 00000000 00011100     |
| 00CC   | 00111100 00000000 00000000 00000100     |
| 00DC   | 11101100 00000000 00000000 00010100     |
| 00EC   | 00101100 00000000 00000000 00000100     |
| 00FC   | 10111100 00000000 00000000 00010000     |

The table above only shows the portion of the block from offset 0x00C0-00FF; the block from offset 0x0100-0x013F exhibits the same pattern.

In summary, 62 bytes of the 128-byte block (48.4%) are different, including 92 of the 256 nibbles (35.9%) and 150 of the 1,024 bits (14.6%).

Appendix 2: FTK Imager report for Test #S1

Created By AccessData® FTK® Imager 3.1.3.2

Case Information:
Acquired using: ADI3.1.3.2
Case Number: SHA Test
Evidence Number: S1
Unique Description:
Examiner: GCK
Notes: hash1.pdf

Information for C:\Users\gck\Documents\SHA_test\TestS1:

Physical Evidentiary Item (Source) Information:
[Device Info]
Source Type: Physical
[Drive Geometry]
Cylinders: 3
Tracks per Cylinder: 255
Sectors per Track: 63
Bytes per Sector: 512
Sector Count: 62,719
[Physical Drive Information]
Drive Model: SanDisk Cruzer Mini USB Device
Drive Serial Number: 20051941901913139434
Drive Interface Type: USB
Removable drive: True
Source data size: 30 MB
Sector count: 62719
[Computed Hashes]
MD5 checksum: 62960d3b87b42763f817665e11560fb7
SHA1 checksum: 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18

Appendix 3: FTK Imager report (partial) for Test #S2

Created By AccessData® FTK® Imager 3.1.3.2
Case Number: SHA Test
Evidence Number: S2
Examiner: GCK
Notes: hash1.pdf (overwrite)

Information for C:\Users\gck\Documents\SHA_test\TestS2:
[Computed Hashes]
MD5 checksum: 62960d3b87b42763f817665e11560fb7
SHA1 checksum: 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18

Appendix 4: FTK Imager report (partial) for Test #S3

Created By AccessData® FTK® Imager 3.1.3.2
Case Information:
Case Number: SHA Test
Evidence Number: S3
Examiner: GCK
Notes: hash2.pdf overwrite

Information for C:\Users\gck\Documents\SHA_test\TestS3:
[Computed Hashes]
MD5 checksum: 5704f9b18354cc804c08b3836e87d43f
SHA1 checksum: a00b80e17de1677d34d21c6e53ff9e0603eadbe6
Image Verification Results:
Verification started:  Fri Feb 24 21:53:00 2017
Verification finished: Fri Feb 24 21:53:01 2017
MD5 checksum:    5704f9b18354cc804c08b3836e87d43f : verified
SHA1 checksum:   a00b80e17de1677d34d21c6e53ff9e0603eadbe6 : verified

Appendix 5: FTK Imager report (partial) for Test #S4

Created By AccessData® FTK® Imager 3.1.3.2

Case Information:
Case Number: SHA Test
Evidence Number: S4
Examiner: GCK
Notes: hash1.pdf overwrite

--------------------------------------------------------------
Information for C:\Users\gck\Documents\SHA_test\TestS4:

[Computed Hashes]
MD5 checksum: 62960d3b87b42763f817665e11560fb7
SHA1 checksum: 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18

Image Information:
Acquisition started:   Fri Feb 24 22:01:21 2017
Acquisition finished:  Fri Feb 24 22:01:26 2017
Segment list:
C:\Users\gck\Documents\SHA_test\TestS4.E01

Image Verification Results:
Verification started:  Fri Feb 24 22:01:26 2017
Verification finished: Fri Feb 24 22:01:26 2017
MD5 checksum: 62960d3b87b42763f817665e11560fb7 : verified
SHA1 checksum: 0a7c8c48793c0742ae37b9d5b4877ef7700b9b18 : verified