Chapter 10

FACTORS AFFECTING ONE-WAY HASHING OF CD-R MEDIA

Christopher Marberry and Philip Craiger

Abstract While conducting a validation study of proficiency test media we found that applying the same hash algorithm against a single CD using different forensic applications resulted in different hash values. We formulated a series of experiments to determine the cause of the anomalous hash values. Our results suggest that certain write options cause forensic applications to report different hash values. We examine the possible consequences of these anomalies in legal proceedings and provide best practices for the use of hashing procedures.

Keywords: Cryptographic hash functions, one-way hashing, CD-R media

1. Introduction

Digital forensics professionals frequently use hash algorithms such as MD5 [7] and SHA-1 [4] in their work. For example, they may identify notable files (e.g., malware or child pornography) on media by comparing their known hash values with the hash values of files that exist on the media. Examiners also use hash values to identify and exclude common files, e.g., system files and utilities, thereby reducing the search space in investigations of digital media. But the most important use of hashing is the verification of the integrity of evidentiary media: verifying that a forensic duplicate is a bit-for-bit copy of the original file, and, in particular, verifying that a forensic duplicate has not been altered during the chain of custody. This use of cryptographic hashing is critical in judicial proceedings as it helps preserve a defendant's right to a fair trial.

While conducting a validation study of digital forensics proficiency test media we were surprised to find that several commonly used forensic applications reported different hash values for the same CD. Since a hash value is computed based on the contents of a file, a different hash value should be obtained only if the file's contents were modified. Changes in a file's metadata, e.g,. date and time stamps, location and name, should not affect the computed hash value.

To determine the cause of the anomalous results, we formulated a series of experiments using several variables: CD write options, system hardware, operating system and CD drive. The experiments involved the use of four popular forensic applications to calculate MD5 hash values of several test CDs. Our results suggest that certain write options cause forensic applications to report different hash values. We examine the possible consequences of these anomalies in legal proceedings and provide best practices for the use of hashing procedures.

2. Experimental Method

To reduce the number of factors in our experimental design, we chose variables that were most likely to affect the hash values. These variables were: (i) CD write options, (ii) system hardware, (iii) operating system, and (iv) CD drive.

2.1 CD Write Options

Several dozen write attributes are available for writing a CD. The CD used in our original validation study was created using the `k3b` application in Linux. We used default writing methods to write the CD, which included track-at-once, ISO-9660 + Joliet, and non multi-session.

We employed three common write attributes in our experiments. Each attribute had two options, resulting in eight distinct experimental cells. The three attributes, which are described in more detail below, were: (i) disk-at-once (DAO) versus track-at-once (TAO), (ii) multi-session versus non multi-session, and (iii) ISO 9660 versus ISO 9660 + Joliet.

2.1.1 Disk Write Method. The track-at-once (TAO) disk write method, by default, inserts a two-second pause between tracks during the writing process. It is commonly used to write disks with multiple tracks or disks with audio and data tracks [12]. The disk-at-once (DAO) option writes all the tracks on a CD in one pass, allowing a variable-length pause or no pause between tracks. Unlike TAO, DAO, by default, does not insert gaps between tracks [12]. DAO is commonly used when there is no need to insert gaps between tracks or when a gap that is not two seconds long is needed [12].

2.1.2 Session. The multi-session write option allows multiple sessions to be written on a disk. A session is a container for the individual components that make up the structure of a CD. The session components comprise a lead-in area, the track(s) containing data, and a lead-out area [11]. The lead-in area contains the table of contents for the session, which gives the location of each track in the session (similar to a partition table) [11]. Tracks are the sequential sectors on the disk itself. The lead-out area closes the session on the disk [11]. The non multi-session write option only allows for one session to be opened and closed on a disk. As with a multi-session disk, the session on a non multi-session disk contains a lead-in area, data tracks, and a lead-out area.

2.1.3 File System. The ISO 9660 file system, which was developed for CDs, allows data to be written so that it is accessible by any operating system [3]. The Joliet extension to the ISO 9660 standard allows filenames up to 64 characters [10]. (The ISO 9660 standard restricted filenames to eight characters.)

Table 1. Test systems.

Hardware and Operating System	**Optimal Drive and Firmware**
System 1: Dell Optiplex 260 w/ Windows 2000 SP4	Samsung SC-148C w/ Firmware B104
System 2: Dell Optiplex 620 w/ Windows Server 2003 SP1	NEC ND-3550A w/ Firmware 1.05
System 3: Dell Poweredge 2800 Dual Booting Windows XP SP2/Linux	Samsung SN-324S w/ Firmware U304

2.2 Test Systems

We used three computer systems in our tests, each with a different optical drive in order to determine if different hardware configurations might produce different hash values. We also used different operating systems – Windows 2000, Windows XP, Windows 2003 Server and Redhat Linux Enterprise Workstation 4 – to establish if they had an effect on the hash results. The hardware, operating system and optical drive configurations are presented in Table 1. Since this is not a fully crossed experimental design, it is not possible to separate the hardware configuration and operating system effects.

2.3 Hashing Applications

We selected commonly used forensic applications to hash the test media. For the Windows systems, the applications included Guidance

Software's EnCase 4 and EnCase 5 [5], AccessData's Forensic Toolkit (FTK) [1] and X-Ways Forensics [13]. Specifically, EnCase 4.22a, EnCase 5.05a, FTK Imager 2.2 and X-Ways Forensics 13.0 were installed on each system (see Table 1). For the Redhat Enterprise Linux 4 applications, we used the command line utility `md5sum` 5.2.1, `readcd` 2.01 and `isoinfo` 2.01. We used `md5sum` to produce the MD5 hash for each disk, `readcd` [2] to report the TOC and the last sector used for each disk, and `isoinfo` [2] to report and verify that certain write options were in fact used to create the disk [8].

2.4 CD Test Media

The CD test media used were Imation brand 700MB 52x rated CD-Recordable disks. The test CD disks were created using Nero Burning ROM version 6.6.1.4 [6] on an IBM ThinkPad T43 laptop with a Matshita UJDA765 drive, firmware revision 1.02. We selected Nero because it is a popular CD writing application that is often bundled with OEM computers and retail optical drives [6].

Each disk had the three test components enabled within the tabs of the new compilation menu in Nero. Each disk was set to Data Mode 1, ISO Level 1 for filename length, and ISO 9660 for the character set. Data Mode 1 is a part of the Yellow Book Standard for CD-ROMs; this mode is traditionally used for disks containing non-audio/video data [9]. Data Mode 2 is traditionally used for disks containing audio or video data. Mode 1 utilizes EDC and ECC error correction techniques to ensure data integrity whereas Mode 2 does not [3]. ISO 9660 Level 1 only allows file names with a maximum length of eight characters with a three character extension and a directory depth of eight levels to be written to the disk [10]. The ISO 9660 character set is a subset of the ASCII standard that allows for alpha characters a-z, numbers 0-9 and the underscore "_" [10].

The relaxation options, "Allow path depth of more than 8 directories," "Allow more than 255 characters in path," and "Do not add the ';1' ISO file version extension" were unchecked except for the "Allow more than 64 characters for Joliet names" if an ISO 9660 + Joliet disk was used [6]. The label was the default automatic with a label of "new" and the date's information was also the default [6].

We copied the same executable file to each CD. We expected the forensic applications to report the same hash value for the same CD. Because of the different write options, timestamps, etc., for each of the eight test CDs, it made no sense to compare hash values across CD test conditions as the hash values would be expected to be different.

Table 2. Hash values for Test 1.

Application	**Hash Value**	**Sectors**
EnCase 4	48D3F3AAA43A3AFF516902F0278F849B	1,207
EnCase 5	70E4FA9880726AA8B5BA1E752576CAA9	1,208
FTK	70E4FA9880726AA8B5BA1E752576CAA9	1,208
X-Ways	70E4FA9880726AA8B5BA1E752576CAA9	1,208
`md5sum/readcd`	70E4FA9880726AA8B5BA1E752576CAA9 (System 3 Only)	1,208

3. Experimental Results

This section presents the results of the eight tests. Note that the only valid comparison of hash values is within a particular test. This is because each write option creates different information on a disk, resulting in different hash values for the eight test CDs. Only one table (Table 2) is provided for Tests 1–4 because no variations were observed across tests and systems.

Tests 1–4 (DAO, Multi-Session, ISO 9660)

Systems 1, 2 and 3 reported the same results for the first test disk (Test 1), with the exception of EnCase 4. The EnCase 4 result is anomalous in that it detected and scanned one less sector than the other forensic applications (1,207 sectors instead of 1,208 sectors reported by the other programs). For System 3, `md5sum` reported the same hash value as the Windows applications. `isoinfo` correctly verified the presence of an ISO 9660 disk with no Joliet support. `readcd` reported that the last sector used was 1,208, which correlated with the results for all the Windows applications except EnCase 4.

Results of Tests 1–4

Due to space constraints and consistent results for Tests 1–4, specific results for Tests 2–4 are omitted. The results of Tests 2–4 have the same pattern as those of Test 1, for which EnCase 5, FTK, X-Ways and `md5sum` reported the same hash results (within each test). EnCase 4 exhibited the same behavior for Tests 1–4 in that it reported one less sector than the other applications and, therefore, produced different hash values. The results indicate that no combination of write options, hardware or operating systems had an effect on the hash values produced. The only anomaly was observed for EnCase 4, which undercounted the number of sectors ($n - 1$) and always produced a different hash value. Further study indicated this behavior to be consistent for all CDs using the DAO condition, which always resulted in a different hash value (corresponding to $n-1$ sectors hashed).

Table 3. Hash values for Test 5 (Systems 1 and 2).

Application	Hash Value	Sectors
EnCase 4	A296A352F2C8060B180FFE6F32DE6392	1,207
EnCase 5	A296A352F2C8060B180FFE6F32DE6392	1,207
FTK	44133FEB352D37BC365EC210DF81D7FD	1,208
X-Ways	050EA9954ADA1977CE58E894E73E0221 (1 Read Error)	1,208

Table 4. Hash values for Test 5 (System 3).

Application	Hash Value	Sectors
EnCase 4	A296A352F2C8060B180FFE6F32DE6392 (1 Read Error)	1,207
EnCase 5	7A1366AE9CC3A96FD9BF56B9B91A633B	1,206
FTK	44133FEB352D37BC365EC210DF81D7FD	1,208
X-Ways	2211A026EC7F309517050D55CEEE2954 (2 Read Errors)	1,208
`md5sum/readcd`	(I/O Errors; System 3 Only)	1,208

Test 5 (TAO, Multi-Session, ISO 9660)

Test 5 resulted in discrepancies in reported hash values between systems and applications. Consequently, the results are presented in two tables (Table 3 for Systems 1 and 2, and Table 4 for System 3). EnCase 4 and 5 produce the same results for Systems 1 and 2. The results for EnCase 4 were the same for all three systems, even though a read error was reported for System 3. Note that EnCase 5 reported a different hash and sector count ($n-1$) for System 3. X-Ways encountered a read error and reported the same hash value for Systems 1 and 2, but a different value for System 3. X-Ways also reported two read errors for System 3. Note that `md5sum` reported an I/O error and would not hash the CD. FTK reported the same hash value for all three systems; however, this value was different from the hash values reported by the other applications. `isoinfo` correctly verified the presence of an ISO 9660 disk with no Joliet extensions enabled. `readcd` reported that the last used sector was 1,208, which correlated with the results obtained with FTK and

X-Ways. On the other hand, both versions of EnCase reported 1,207 sectors.

Interestingly, EnCase 4 and 5 reported the same hash value for the test CD with Systems 1 and 2, which is inconsistent with the different results obtained for Tests 1–4. It is not clear why the Test 5 results are consistent for Systems 1 and 2, but inconsistent for System 3. EnCase 5 reported $n - 1$ sectors, while FTK, X-Ways and `readcd` reported n (1,208) sectors on the same disk. It is not clear why EnCase 5 exhibits this behavior.

Of special concern to examiners is the fact that not a single consistent hash value was reported for System 3. In all, five different hash values were reported for the test disk (Test 5). Given the inconsistent results for System 3, it would not be possible to determine which hash value is correct if System 3 was the only one used to validate a CD.

Table 5. Hash values for Test 6 (Systems 1 and 2).

Application	**Hash Value**	**Sectors**
EnCase 4	A0B7E6A28FB17DB7AB1F5C0E1ED414C5	1,211
EnCase 5	A0B7E6A28FB17DB7AB1F5C0E1ED414C5	1,211
FTK	2109A7DBCF1B83D357EA0764100672B1	1,212
X-Ways	0006AEA93E620C864530ADF7FC287A61 (1 Read Error; System 2 Only)	1,212

Table 6. Hash values for Test 6 (System 3).

Application	**Hash Value**	**Sectors**
EnCase 4	A0B7E6A28FB17DB7AB1F5C0E1ED414C5 (1 Read Error)	1,211
EnCase 5	CE37E507FCCFFF857B2BB79F3E57483B	1,210
FTK	2109A7DBCF1B83D357EA0764100672B1	1,212
X-Ways	B703C2E0D42E301ECA71F1C3C1BF6C71 (2 Read Errors)	1,212
`md5sum/readcd`	I/O Error; System 3 Only)	1,212

Test 6 (TAO, Multi-Session, ISO 9660 + Joliet)

The Test 6 results (Tables 5 and 6) had similar discrepancies as those for Test 5. EnCase 4 and 5 reported the same hash values for Systems 1 and 2. EnCase 4 provided consistent hash values for all three systems, but produced a read error for System 1 (similar to Test 5 above). X-

Ways reported read errors for Systems 1 and 2. For System 3, as in the case of Test 5, the forensic applications produced five different hash values. Note that FTK reported the same hash value for all three systems; however, this value was different from the hash values reported by the other applications. Once again, `md5sum` reported an I/O error and did not produce a hash value. `isoinfo` verified an ISO 9660 disk with Joliet extensions enabled. `readcd` reported that the last sector used was sector 1,212, which correlated with the results for FTK and X-Ways as in Test 5.

Table 7. Hash values for Test 7 (Systems 1 and 2).

Application	**Hash Value**	**Sectors**
EnCase 4	897BA35435EE6183B03B7745E4FFCDC0	1,206
EnCase 5	897BA35435EE6183B03B7745E4FFCDC0	1,206
FTK	978F6D133EE22C7C8B692C1A43EFE795	1,207
X-Ways	BAFF87CEF354BF880D4AD8919A25CB6E	1,207

Table 8. Hash values for Test 7 (System 3).

Application	**Hash Value**	**Sectors**
EnCase 4	897BA35435EE6183B03B7745E4FFCDC0 (1 Read Error)	1,206
EnCase 5	986F1E56D89476ABC8F69958C551A42D	1,205
FTK	978F6D133EE22C7C8B692C1A43EFE795	1,207
X-Ways	664C56F4A3F450A8FD1B1D37C526F47A (2 Read Errors)	1,207
`md5sum/readcd`	(I/O Error; System 3 Only)	1,207

Test 7 (TAO, Non Multi-Session, ISO 9660)

The pattern of results for Test 7 (Tables 7 and 8) is similar to those for Tests 5 and 6. EnCase 4 and 5 reported the same hash value for Systems 1 and 2. EnCase 4 reported consistent hash values for all three systems, although it again reported a read error for System 3. X-Ways reported read errors for System 3 only. FTK reported consistent hashes for all three systems, as did EnCase 4. Five different hash values were reported for System 3. Once again, `md5sum` reported an I/O error and did not produce a hash value. `readcd` reported a sector count of 1,207, which correlated with the results for FTK and X-Ways. `isoinfo` verified the presence of an ISO 9660 disk with no Joliet extensions enabled.

Table 9. Hash values for Test 8 (Systems 1 and 2).

Application	**Hash Value**	**Sectors**
EnCase 4	284EF959A864DACF83206C1AA1A0B4CB	1,210
EnCase 5	284EF959A864DACF83206C1AA1A0B4CB	1,210
FTK	7F49A83724130E46974CD24097C01F3A	1,211
X-Ways	41BD02ED23DF42190F06CACC97275D30	1,211

Table 10. Hash values for Test 8 (System 3).

Application	**Hash Value**	**Sectors**
EnCase 4	284EF959A864DACF83206C1AA1A0B4CB (1 Read Error)	1,210
EnCase 5	378D6B62CCB8A81CC5001569AEF1A3D4	1,209
FTK	7F49A83724130E46974CD24097C01F3A	1,211
X-Ways	910962B3A2561FCDB8382B14B9FDDA8B (2 Read Errors)	1,211
`md5sum/readcd`	(I/O Error; System 3 Only)	1,211

Test 8 (TAO, Non Multi-Session, ISO 9660 + Joliet)

The Test 8 results (Tables 9 and 10) are similar to those for Tests 5–7. For Systems 1 and 2, EnCase 4 and 5 produced matching hash values, albeit with one read error for EnCase 4 with System 3. FTK reported consistent hash values for all three systems, but this value was inconsistent with the hash values reported by the other forensic applications. X-Ways produced matching hash values for Systems 1 and 2, but not for System 3. None of the X-Ways hash values matched the values obtained with the other applications. Once again, `md5sum` reported an I/O error and did not produce a hash value. `readcd` reported a sector count of 1,211 that again correlated with the results obtained with FTK and X-Ways. `isoinfo` verified the presence of an ISO 9660 disk with Joliet extensions enabled.

Results of Tests 5–8

The results for Test 5–8 are inconsistent with those obtained for Tests 1–4. The following anomalous patterns are consistently observed for Tests 5–8.

1. EnCase 4 and 5 produced the same hash value for Systems 1 and 2. However, the number of sectors read was one less than reported by the other Windows applications.

2. EnCase 4 reported the same hash value for Systems 1, 2 and 3, and produced a read error for all tests on System 3.
3. EnCase 4 and 5 reported different hash values for System 3.
4. EnCase 5 reported the same hash value for Systems 1 and 2, and a different hash value for System 3.
5. FTK reported the same hash value for Systems 1, 2 and 3. However, the hash value was different from the values reported by the other applications.
6. X-Ways reported the same hash value for Systems 1 and 2.
7. X-Ways produced read errors for all tests on Systems 1, 2 and 3.
8. `md5sum` failed to produce a hash value, always reporting an I/O error.
9. For Systems 1 and 2, the sector count pattern was always $n-1$ for both EnCase 4 and 5, and n sectors for FTK and X-Ways.
10. For System 3, the sector count pattern was always $n-1$ for EnCase 4, $n-2$ sectors for EnCase 5, and n sectors for FTK and X-Ways.

In light of these results it is clear that at least some write options and hardware/operating system combinations affected the hash values (especially when the TAO write option was used). In Tests 5–8 for System 3, there were never less than five distinct hash values although two applications (FTK and X-Ways) always reported the same sector count. This is more than likely related to the read errors that X-Ways encountered with most TAO disks used in our experiments. The results should also be cause for concern for examiners who use the forensic applications to hash evidentiary CDs with the aforementioned write options.

It appears that that the write method used (TAO versus DAO) is the primary factor in producing the anomalous results. The results of Tests 1–4, which used DAO, produced the same hash value regardless of session type or file system used. The anomalous results obtained in Tests 5–8 suggest that the TAO write method affects the computed hash values.

4. Results for a Bad Drive

During our initial testing we found that System 2 configured with a HL-DT-ST GWA4164B drive (firmware version D108) was "bad." The drive was very erratic at reading disks during Tests 1–7; by Test 8, the drive would not read the disk at all. We confirmed that the drive was bad by comparing the hash values obtained in the eight tests with the results obtained when a new drive was installed in System 2; also, we examined the hash values obtained in the eight tests for Systems 1 and 3. We addressed the issue by replacing the bad drive with an NEC ND-3550A drive (firmware version 1.05) to create a new System 2, which was used for Tests 1–8 described above.

Table 11. Test 1 with Bad Drive (System 2).

Application	**Hash Value**
EnCase4	25E52C25C5841A7415F65301121DF986
EnCase5	A00EF2AD9822461DAC328C743D45638C
FTK	70E4FA9880726AA8B5BA1E752576CAA9
X-Ways	A00EF2AD9822461DAC328C743D45638C

Table 12. Test 1 with Good Drive (System 2).

Application	**Hash Value**
EnCase4	48D3F3AAA43A3AFF516902F0278F849B
EnCase5	70E4FA9880726AA8B5BA1E752576CAA9
FTK	70E4FA9880726AA8B5BA1E752576CAA9
X-Ways	70E4FA9880726AA8B5BA1E752576CAA9

Table 11 helps illustrate the effects of a bad drive on the reported hash values (Test 1 with System 2). There are some interesting consistencies in the reported hash values for the bad drive (Table 11). For example, EnCase 5 and X-Ways reported the same hash value; however, this value is inconsistent with the other hash values. Another interesting observation is that the hash value reported by FTK was the same for the bad drive (Table 11) and the good drive (Table 12).

The sector counts reported by the bad drive for all the disks in the experiment had the same patterns. EnCase 4 reported *n-3* sectors, EnCase 5 *n-2* sectors, FTK *n* sectors, and X-Ways *n-2* sectors. The sector count reported by FTK correlates with FTK's results for all three good drives, which might explain why they all produce the same hash values.

5. Discussion

Our experimental results demonstrate clear trends in the factors affecting the values computed for CDs, especially between the DAO and TAO writing methods. CDs written with the DAO option produced consistent hash values. On the other hand, CDs written under the TAO option produced anomalous results, including inconsistent hash values and sector counts across forensic applications, read errors for certain forensic applications, and inconsistent hash values across hardware configurations and operating systems.

The results obtained in the case of the “bad drive” underscore the importance of verifying that an optical drive or media reader – especially one used in a digital forensics investigation – reports accurate data. This is challenging because there is usually no means of calibrating such a device after it leaves the manufacturing facility. Consequently, we recommend that examiners triangulate hash results across other drives and/or systems. If the time and/or systems are unavailable to triangulate the results, then a comparison of the hash value computed by the untested system with the hash value of a known verified and “triangulated” disk could be used to show that the results reported by a particular drive are accurate.

6. Conclusions

Cryptographic hashing is crucial to verifying the integrity of digital evidence. Given the results obtained for the TAO disk writing option, the potential exists that the integrity of digital evidence could be challenged on the grounds that the hash value calculated by a defense expert does not match the value presented by the prosecution. Such a challenge, if successful, would almost certainly affect the outcome of the case. However, the fact that different hardware/software combinations produce different hash values for an item of digital evidence does not mean that cryptographic hashing cannot be relied on to verify the integrity of evidence. As long the entire hashing process can be duplicated and the results shown to match, there should be no problems in using hash values to verify the integrity of the evidence. Therefore, it is crucial that examiners maintain detailed documentation about the specific hardware and operating system configurations, optical drives and firmware revisions, and the forensic applications used to produce hash values of evidentiary items.

References

[1] AccessData, Forensic Toolkit (www.accessdata.com).

[2] CD Record (cdrecord.berlios.de/old/private/cdrecord.html).

[3] G. D’Haese, The ISO 9660 file system (users.pandora.be/it3.consultants.bvba/handouts/ISO9960.html), 1995.

[4] D. Eastlake and P. Jones, US Secure Hash Algorithm 1 (SHA1) (www.ietf.org/rfc/rfc3174.txt), 2001.

[5] Guidance Software, EnCase (www.guidancesoftware.com).

[6] Nero AG, Nero (www.nero.com).

[7] R. Rivest, The MD5 message-digest algorithm (www.ietf.org/rfc/rfc 1321.txt), 1992.

[8] M. Shannon, Linux forensics (www.agilerm.net/linux1.html), 2004.

[9] Sony Electronics, What are CD-ROM Mode-1, Mode-2 and XA? (sony.storagesupport.com/cgi-bin/sonysupport.cgi/M1QyNUjPDA 4dzatpCN5uJ=xjZSMlgW60/faq/view/413).

[10] Wikipedia, ISO 9660 (en.wikipedia.org/wiki/ISO_9660).

[11] Wikipedia, Optical disc authoring (en.wikipedia.org/wiki/Optical _disc_authoring).

[12] Wikipedia, Optical disc recording modes (en.wikipedia.org/wiki /Disc_At_Once).

[13] X-Ways Software Technology AG, X-Ways Forensics (www.x-ways .net).