Naif Arab University for Security Sciences
Journal of Information Security & Cybercrimes Research
مجلة بحوث أمن المعلومات والجرائم السيبرانية
https://journals.nauss.edu.sa/index.php/JISCR

JISCR

Comparative Study and Analysis on Integrity of Data Files Using Different Tools and Techniques

Kumarshankar Raychaudhuri[1]*, M. George Christopher[2], and Nayeem Abbas Hamdani[3]

[1]Lok Nayak Jayaprakash Narayan National Institute of Criminology & Forensic Science, (LNJN NICFS), New Delhi, India.
[2]State Forensic Science Laboratory (SFSL), Madiwala, Bengaluru, India.
[3]Jammu & Kashmir Police, J&K, India.

Received 09 Apr. 2021; Accepted 27 May. 2021; Available Online 01 Jun. 2021

Abstract

Digital forensic investigation is the scientific process of collection, preservation, examination, analysis, documentation and presentation of digital evidence from digital devices, so that the evidence is in compliance with legal terms and acceptable in a court of law. Integrity of the digital evidence is an indispensable part of the investigation process and should be preserved to maintain the chain of custody. This is done through hashing technique using standardized forensic tools. However, while handling the evidences, lack of knowledge might lead to unintentional alteration of computed hash. This violates the chain of custody and makes the evidence inadmissible in a court of law. In this paper, our objective is to determine the different conditions under which the original hash value of a digital evidence changes. For this, we create different scenarios using sample data files and compute their hash values. A comparative study and analysis are done to determine in which scenario the original hash value of the data file changes. The results of the research will prove useful and essential for Criminal Justice Functionaries in gaining knowledge about various conditions leading to the change in hash value of digital evidence and therefore, avoid its accidental alteration during forensic investigation/examination.

I. Introduction

Digital Forensics, referred to as the application of forensics science for the identification, collection, preservation, examination, analysis, interpretation and documentation of digital evidence and other electronic exhibits while maintaining a strict chain of custody [1,2]. Digital evidence, which forms an integral part of every digital forensic investigation process, is defined as a piece of data that is recorded, stored or transferred through a computer system or similar digital or electronic devices, and can be read, understood and interpreted by a person, computer or similar digital device [1]. Evidence can originate from multiple sources such as seized computer hard-drives and back-up media, ISP records, USB flash drives, e-mail messages, network traffic etc. [3,4]. However, the trustworthiness of this data, source device or both is an important question, which must be looked into carefully by forensic examiners.

According to the principles laid out by the Association of Chief Police Officers (ACPO) [5], "No ac-

Keywords: Integrity, Hash Values, Digital Forensics, Digital Evidence, Evidence Samples, Metadata, OSForensics, WinHex.

Production and hosting by NAUSS

* Corresponding Author: Kumarshankar Raychaudhuri
Email: ksrc089@gmail.com
doi: 10.26735/SYMQ8715

tion taken by Law Enforcement Agencies, persons employed within those agencies or their agents should change the data which may subsequently be relied upon in court." This implies that once a potential digital exhibit is identified at the crime scene, the concerned investigating officer should take necessary steps to ideally hash the exhibit as a part of seizure and collection of digital evidences. According to guidelines of National Institute of Standards and Technologies (NIST) [1] and Handbook of Applied Cryptography [6], the integrity of digital evidence is defined as "the property whereby data has not been altered or modified in an unauthorized manner since the time it was created, transmitted or stored by an authorized source". This implies that the evidence should be handled by authorized personnel and with proper precautions so that there are no inadvertent changes in the original evidence, which might compromise its integrity. The tools used for collection and acquisition of digital evidences employ different hashing algorithms to verify the evidence's integrity.

Digital information is very delicate and fragile and even a minute mistake can prove to be costly. Therefore, not just data tampering but lack of proper knowledge regarding handling of digital evidences might also lead to change in computed hash value [3,7]. The objective of our research is to conduct various experiments to determine various practices (such as modifying file metadata, filename and file extension, file encryption, file compression, file printing, storing the same file in different formats, use of different versions of Windows OS and steganography), that can lead to alteration of computed hash value of digital evidences. To achieve this objective, we have demonstrated different scenarios. In each scenario, samples of data files have been created and a specific activity (as mentioned above) has been performed. Multiple hash calculating tools (using both MD-5 and SHA-1 hash algorithms) have been used to compute the hash values of the data samples (before and after performing the specific activity) in each scenario. The purpose of using multiple tools is to ensure that the result is validated. Finally, a comparative study and analysis of the computed hash values for each sample is done to observe in which scenario, the original hash value changes after performing the activity. The results obtained from this research work can prove useful for criminal justice functionaries and forensic fraternity in learning and avoiding those practices during forensic examination or investigation, which can lead to accidental change of hash value of digital evidence, rendering it inadmissible in the court of law.

The research article is divided into five sections: Section II represents the literature review and background study, which highlights the researches done in this field alongwith a brief introduction to hashing and hash algorithms. Section III includes experimental design, which highlights the methodology and the tools used. This is followed by discussion and analysis of obtained results in Section IV. The research work is concluded in Section V.

II. Literature Review

The integrity of digital exhibits and evidences plays an essential part in the entire process of digital forensic examination [8]. It can ensure that the data present is complete and unaltered from the time of its acquisition till the time it is presented to the court of law. For a digital evidence to be admissible in the court, it must be authentic, complete and reliable [9,10].

Hashing is defined as the technique wherein a fixed-length alphanumeric string is generated from a variable-sized input through hash algorithms [11,7]. The alphanumeric string is known as the hash value or "digest". Hashing plays a significant role in determining the authenticity and reliability of digital evidences [12,8]. Hash functions are collision resistant, which means that the probability of two different inputs having the same hash value is astronomically small and such a result would mean that both the pieces of data are identical [12]. The hashing process forms the backbone of every digital forensic investigation. The hash value of a piece of data might change not just due to its modification but also as a result of unintentional mishandling. This would compromise their authenticity, reliability and integrity.

Hash functions are mathematical algorithms which take individual data or an entire file as an input and produce a fixed length string, called "digest". Some of the popularly used hashing algorithms are as follows:

MD5 (Message Digest 5) – MD5 algorithm, pro-

posed by Ron Rivest, is a widely used cryptographic hash-function with a hash value of 128-bits. The letters "MD" stands for "message digest", while the numerals refer to the version of the algorithm, being from the same hash-function family [7,13,14].

SHA-1 (Secure Hash Algorithm-1) – SHA-1, designed by National Security Agency (NSA), is a cryptographic hash function that takes an input and produces a 160-bit hash value [13]. SHA-1 forms a part of several widely used security applications and protocols, namely TLS, SSL, PGP, SSH and S/MIME [7,14].

SHA-2 (Secure Hash Algorithm-2) – SHA-2 is a cryptographic hash function designed by NSA. It consists of SHA-224, SHA-256, SHA-384, SHA-512 and SHA-512/256. The hash value might range from 224 bits in size to 512 bits in size depending upon the hash function used [15].

In [11], the role of hash value in digital forensics examination has been demonstrated with the help of various cases. The focus of their research work has been on the entire digital drive's hash value and not just a single file. Five scenarios have been demonstrated including addition of file, removal of file, modification in the contents of the file, shifting of contents from one file to another and updating contents of an existing file, on a storage disk. It has been observed that in each scenario, there has been a change in the original hash value of the file. Hence, it has been concluded by the authors that even a small modification in digital evidence can be detected with the help of hash value.

In [16], experiments have shown that the hash value of a hard disk drive changes when it is plugged into the forensic workstation without using a write-blocker. The reason for the change in hash value has been attributed to the fact that extra files got created in the digital exhibit when not plugged using write-blocker. This has been explained in [7], where the author has found that the NTFS file-system plays a significant role in changing the hash value of digital evidence in the absence of write-blocker. On careful analysis of forensic image of the storage device, it has been observed that there are major changes in metadata files namely $MFT, $LogFile and $Tops. This has further, proven that change in hash value of a storage device might not be indicative of the fact that some data files or their contents have been altered. Therefore, the author has suggested that more significance should be given to the hash value of individual data files rather than that of entire exhibit. With this thought, we have conducted this empirical work to find out the different activities that result in modification of the hash value of a data file (digital evidence).

III. Experimental Design

The experiments were conducted by creating 11 different scenarios. In each scenario, different samples of data files (digital evidences) have been created, a specific activity has been performed on the sample and hashing tools have been used to compute the hash value before and after the activity. This section will provide a brief description of the methodology adopted and the different digital forensic tools used.

A. Methodology

The following steps have been used for creating sample data in each scenario:

1. *Scenario A* – Two data files (1 MS-word and 1 notepad file) "Sample_file.docx" and "Sample_file.txt" with text content have been created and hash value computed. Further, some text is added to both files, which are saved, and hash value computed again. The newly added text is deleted, files are saved again, and hash value is computed for the third time.

2. *Scenario B* – MS-word data file "Sample_file.docx" is created and hash value computed. Its name is changed to "Test_file" and extension is changed to ".pptx" from ".docx". Hashing is done again.

3. *Scenario C* – MS-Word data file "Sample_file.docx" is created and stored in compressed form "Sample_file.rar" and the hash value is obtained.

4. *Scenario D* – The properties (metadata) of a

MS-Word data file "Sample_file.docx" such as "Title", "Subject" and "Author Name" are altered and hash values are generated before and after changing of properties.

5. *Scenario E* – The Two different types of data files are created. One file is MS-Word file "Sample_file.docx", while another is an image file, "Sample_file.bmp". Secret data is hidden in the document file by inserting it in the "Comments" section, which is a part of file Properties (metadata), without altering any of the file contents. In the case of image file, the same secret data is hidden by the application of steganography tools, without morphing the image. This technique is referred to as Steganography. The hash values of the files are computed before and after hiding of secret data.

6. *Scenario F* – A MS-word data file "Sample_file.docx" is created in a newer version of the application (MS-Word 2013) in the "docx" format. The same file is opened in an older version of the application (MS-Word 2003) and saved again in "doc" format, without modifying any content. The hash values of both files are computed.

7. *Scenario G* – A PDF (Portable Document Format) version of the MS-word file "Sample_file.docx" is created by the name "Sample_file.pdf" and the hash value is computed before and after creation of PDF.

8. *Scenario H*– A MS-Word data file "Sample_file.docx" is encrypted using password, and the hash values computed before and after encryption.

9. *Scenario I*– Different multimedia files such as image, video and audio files is used as sample data. An image file titled "Test_image.jpg" of size 88KB is stored as the original data file. It is opened using image viewing application and then saved again as "Test_image.png" of size 366KB. Similar operation is performed on audio and video files. The hash values are computed both, for original files and after saving the files in different file formats.

10. *Scenario J*– A MS-Word data file "Sample_file.docx" is printed using MS-Word application, without changing any content of the file. The hash value of the file is computed before and after file printing.

11. *Scenario K*– A MS-Word data file "Sample_file.docx" is created and stored in Windows 7 and hash value computed. The file is transferred and stored in Windows 10. Both the operating systems have same version of MS-Word application. The hash value of the file is again computed after storing in Windows 10.

TABLE I

DESCRIPTION OF EXPERIMENTED SAMPLES AND THEIR RESULTS

Scenario No.	Experiments with Samples	Result
Scenario A (File Content Add and Delete)	Two data files (1 MS-word and 1 notepad file) "Sample_file.docx" and "Sample_file.txt" with text content were created and hash value computed. Then some text was added to both files, which were saved, and hash value was computed. This newly added text was deleted, files were saved again, and hash value of the files was computed.	For the .docx file, the hash value changes each time, whereas for .txt file, the original hash value comes back on reverting to original content
Scenario B (File name and extension change)	MS-word data file "Sample_file.docx" was created. Its name was changed to "Test_file" and extension was changed to ".pptx" from ".docx".	No change in hash value

TABLE I

DESCRIPTION OF EXPERIMENTED SAMPLES AND THEIR RESULTS (*Continued*)

Scenario No.	Experiments with Samples	Result
Scenario C (File Compression)	MS-Word data file "Sample_file.docx" was created and stored in compressed form "Sample_file.rar" and the hash value was obtained.	Change in hash value
Scenario D (File Metadata Change)	The properties (metadata) of a MS-Word data file "Sample_file.docx" such as "Title", "Subject" and "Author Name" were altered and hash values were generated before and after changing of properties.	Change in hash value
Scenario E (Steganography)	Two different types of data files were created. One of them was the MS-Word file "Sample_file.docx", while another was an image file, "Sample_file.bmp". Secret information in the document file was hidden by inserting it in the "Comments" section of the "Details" tab, which is a part of file Properties, without altering any of the file contents. In the case of image file, the same secret information is hidden by the application of steganography tools, without morphing the image. This technique is referred to as Steganography. The hash values of the files were computed before and after hiding information.	Change in Hash value
Scenario F (Using different versions of MS-Word application)	A MS-word data file "Sample_file.docx" was created in a newer version of MS-Word (MS-Word 2013) in the "docx" format. The same file was opened in an older version of the application (MS-Word 2003) and saved again in "doc" format, without modifying any content. The hash values of both files were computed.	Change in Hash value
Scenario G (Creating two different versions DOC & PDF of same document)	A PDF (Portable Document Format) version of a MS-word data file "Sample_file.docx" was created by the name "Sample_file.pdf" and the hash value was computed before and after creation of PDF.	Change in Hash value
Scenario H (File Encryption)	A MS-Word data file "Sample_file.docx" was encrypted using password, and the hash values were computed before and after encryption.	Change in Hash value
Scenario I (Storing Multimedia files in different file formats)	Different types of multimedia files such as image, video and audio files were used as data files. An image file titled "Test_image.jpg" of size 88KB was stored as the original data file. It was opened in an image viewing application and saved again as "Test_image.png" of size 366KB. Similar operation was performed on audio and video files.	Change in Hash value
Scenario J (File Printing)	A MS-Word data file "Sample_file.docx" was printed using MS-Word application, without changing any content of the file. The hash value of the file was computed after printing.	Change in Hash value
Scenario K (Storing file in different versions of Windows OS)	A MS-Word data file "Sample_file.docx" was created using MS-Word application in Windows 7. The hash value was computed, and it was copied into USB flash drive and stored in Windows 10. Both the operating systems have same version of MS-Word application. The hash value of the file was again computed after storing in Windows 10.	No change in Hash value

B. Tools used for the experiment

The hash computing tools used in conducting the experiments, as validated [17], are as follows:

OSForensics– OSForensics is a tool used for extracting forensic evidences from computers. It can be used to compute and verify hash values for individual files (e.g. text, audio, image, video files etc.) and folders using different hashing algorithms such as MD5, SHA-1, SHA-256, and CRC32 [18].

WinHex– WinHex is a hexadecimal editor, used in imaging and analysis of disks and files. It can be used for computing and analyzing the hash value of individual files (e.g. text, audio, image, video files etc.) and folders. Different hashing algorithms such as MD5, SHA-1, SHA-256 etc. are supported by this tool [19].

IV. Discussion and Analysis of Results

For The summarized results obtained from conducting the experiments are shown in Table I. Detailed analysis of the results for individual scenarios was done afterwards.

Analysis for Scenario A - The original MD5 hash value of the file was observed to be "f21171d7ef82b-6b27214a66d34180a79" and SHA-1 hash value was found to be "b289adb6a5d0d361e0dd235c-26893f4846a442f7".The hash value of the .docx file changes on adding and saving new content to the file. On reverting the changes to the original content, a new hash value was generated. Thus, it is observed that different hash values are observed each time. This is because .docx is a word-processing application with various bits of metadata known as Application Metadata (as illustrated in Fig.1), which constitute a part of the file and is hashed along with it [20]. As a result, when any change is made to the document, the application metadata changes, which in turn modifies the hash value, making it completely different. Contrary to this, on performing a similar experiment with a .txt file, we get back the original hash value of the file. This is because applications like Notepad area text editors, having no application metadata stored within the file due to which the hash value of the file solely depends upon the content written inside the file and not on any other properties [21]. This is illustrated in Fig. 2.

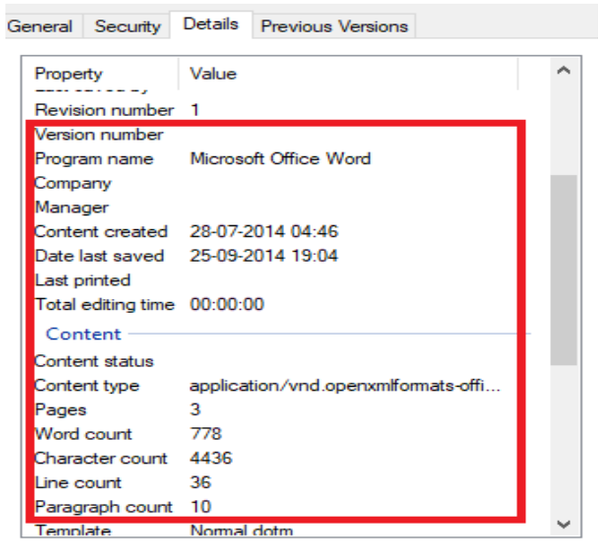

Fig. 1 Application Metadata for a MS-Word file

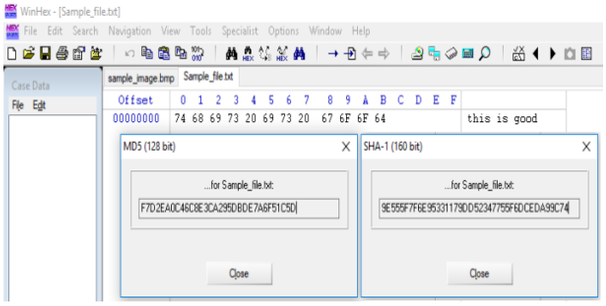

Fig. 2 Change in hash value of MS-word file with the change in Application Metadata (Scenario A)

Analysis for Scenario B - The experiments conducted on the Sample B show no change in the hash value of the data file since the name and extension (file format) are a part of the system metadata of the file and are stored in the Master File Table (MFT), outside the file. These properties can be altered without causing any change to the contents of the file and hence the hash value of the file does not alter [20, 21]. The same result has been found on experimentation with other files such as image files, audio files, PDF etc., as shown in the snapshot in Fig. 3.

Analysis for Scenario C - While performing experiments in Scenario C, there was a change in the hash value after the data file was compressed. On compression, the redundant data within the file is reduced due to which the file size decreases (in

this case the size of the file reduced to 7.35KB from 9.77KB). The lossless compression allows the original data to be reconstructed back once the compressed file is de-compressed again. Hence, due to alteration in the contents, the hash value gets changed [21, 22]. The same result was found on experimentation with other files such as image files, audio files, PDF etc., as illustrated in the snapshot in Fig. 4.

Analysis for Scenario D - Based on the results obtained from the samples, there is a change in the hash value of the original data file, even if the metadata is changed. Every MS-Word document consists of two types of metadata- System Metadata and Application Metadata [20]. As observed during the analysis of Sample A (refer to Fig. 1), Application Metadata resides within the file and does not change unless the contents of the file are changed. On the other hand, System Metadata resides outside the file and can be altered without modifying the contents of the file. In the properties of the file, system metadata consists of the 'Creation/Access/Modified Date', 'Filename', 'Location' etc. while attributes such as 'Title', 'Subject' and "Author's name' constitute Application Metadata and are hashed with the file content. Therefore, altering these fields within the file properties changes the hash value of the file [21], as illustrated in the snapshot in Fig. 5. Therefore, on analyzing the sample in Scenario D, a critical observation is made i.e. there is a change in original hash value even without modifying the file contents, thus resulting in the violation of integrity.

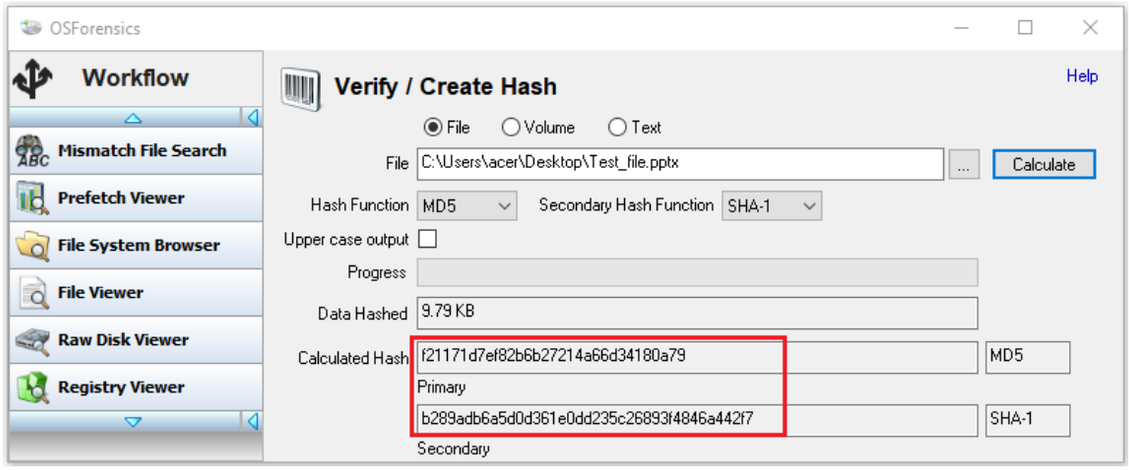

Fig. 3 No change in hash value on modifying file name and extension (Scenario B)

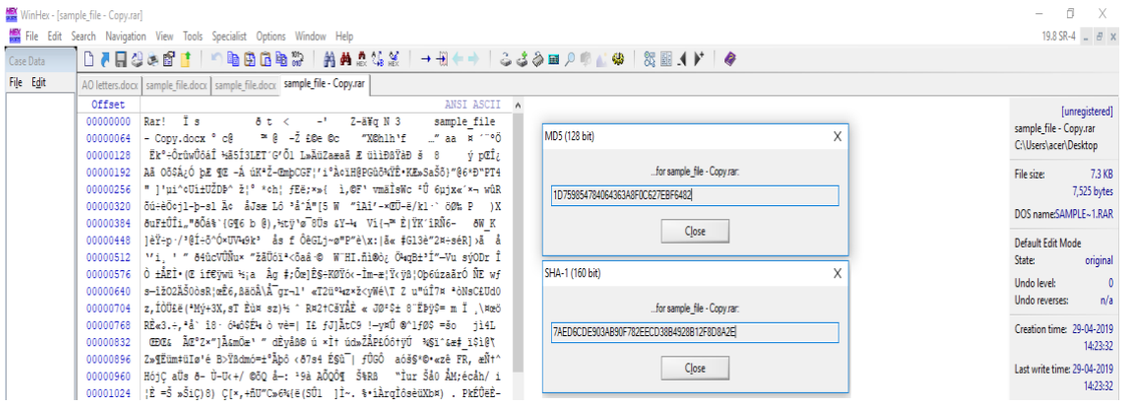

Fig. 4 Change in hash value on compressing the file (Scenario C)

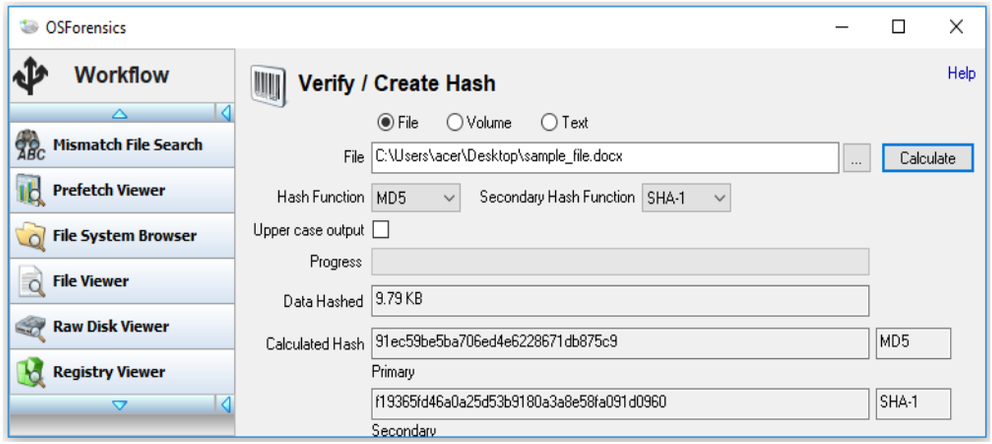

Fig. 5 Change in hash value of .docx file on changing some of its properties (Scenario D)

Analysis for Scenario E - In Scenario E, it was observed that there is a change in the hash values of both, the document as well as the image file when information is hidden in it. The reason for the same is analyzed separately, for document file and image file: In the case of .docx file, the information is hidden in the 'comments' section of a MS-Word document, which falls under the Application Metadata, and hence resides within the file. Therefore, modifying it changes the resultant hash value [20, 21]. On the other hand, in an image file, red, green and blue are the primary component colors, where each colored pixel is represented by eight bits ranging from 0 to 255 (decimal representation) or 00000000 to 11111111 (binary representation). When data or information is hidden inside an image file, the values of these pixels change due to changes in few of the bits. As a result, the data is encoded in the picture creating an imperceptible change in the appearance, which is not visible to the naked eye. However, the hash value of the image does change. Therefore, in both the cases, it is evident that hiding information inside a file transforms its hash value. The same is illustrated in the snapshot in in Fig. 6.

Analysis for Scenario F - Based on the results obtained in Scenario F, it is seen that the hash value changes when the word document is stored in an older '.doc' format. Although there is no addition or deletion to the content in the data file, the integrity is still violated. This is because Microsoft Word keeps a large amount of metadata within the documents. When the same document is opened using a different version of the application, it starts by updating the metadata automatically. Also, the newer version '.docx' acts as a ZIP file, compressing the contents of the file, which reduces the file size. Therefore, change in metadata and file size results in a changed hash value [21].

Analysis for Scenario G - In the experiment of data sample, there is a change in the hash value when the .docx file is converted into its corresponding PDF version. This is because a word document and PDF version might appear the same onscreen; however, they are encoded in entirely different manners. Hence, a change in the file format from (docx) to (pdf) changes the size of the file from 9.77KB to 178.7KB resulting in the change in hash value of the file as well [20, 21]. This is illustrated in the snapshot in Fig. 7.

Analysis for Scenario H - Based on the results achieved, a change in the hash value is observed once a data file is encrypted. The encryption algorithm used by Microsoft Word is AES-128 bit. However, due to encryption, the contents of the evidence file are transformed into an encoded data. This causes a variation in the file size and makes the file look statistically random. Hence, the resulting hash value of the original evidence file changes to form a new one [21, 23], as shown in the snapshot in Fig. 8.

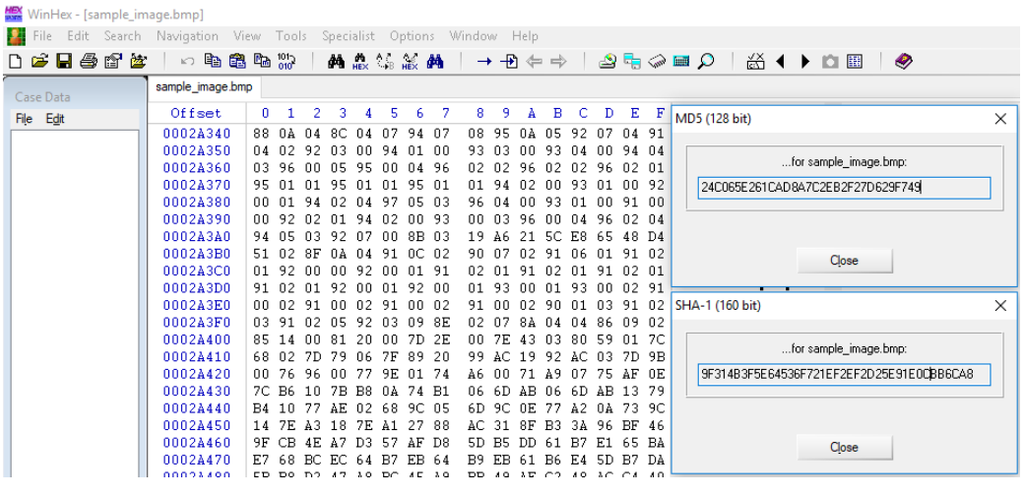

Fig. 6 Hash value of .docx file changes when information is hidden in it (Scenario E)

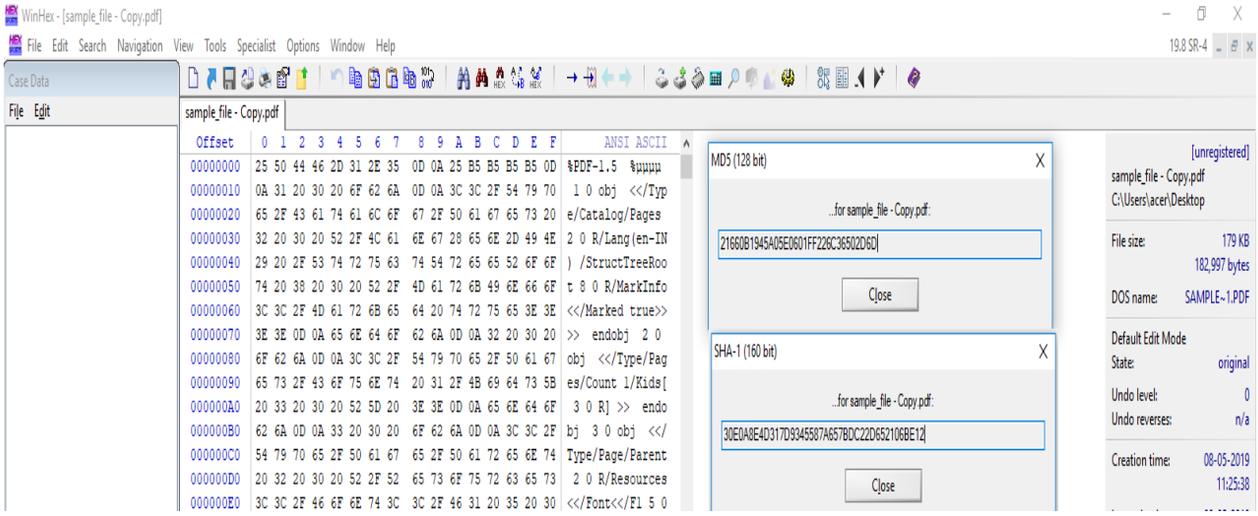

Fig. 7 Hash value of word file changes when converted to corresponding PDF version (Scenario G)

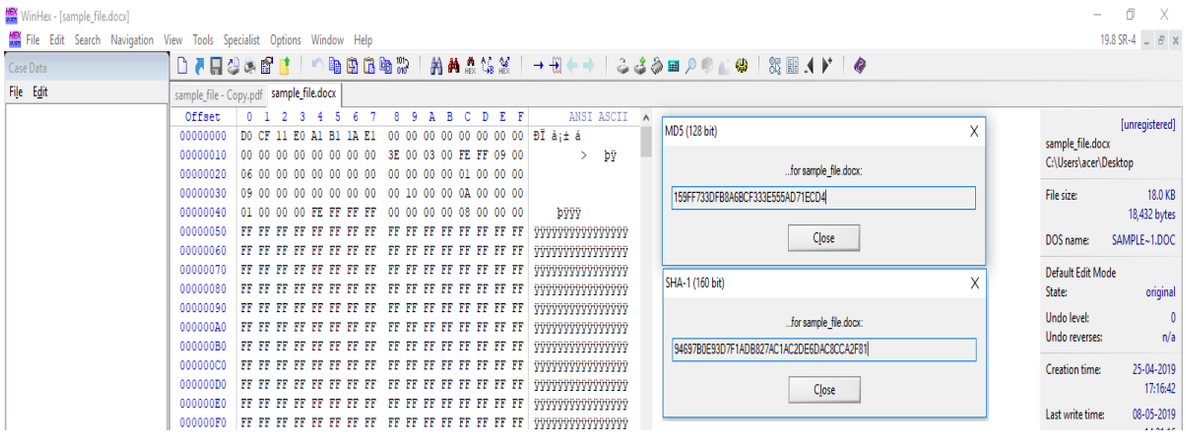

Fig. 8 Hash value of the word file changes on encrypting the file with password (Scenario H)

Analysis for Scenario I – Based on the experiments conducted on the prepared Sample, the hash values of the image files change when they are stored in different file formats (i.e. in .jpg and .png). This is because JPG is a lossy compressed file format, whereas PNG is a lossless compression file format. As a result, the image file in PNG format is much heavier than the same image in JPG format (due to enhancement), which causes the hash value to differ from one another. However, it should be noted that this is different from the case when the extension of the image file is modified from PNG to JPG, wherein there is no change in either the size of the image file or its hash value. Similarly, for audio and video files as well, it has been observed that the hash value changes when the file is converted or stored in another format due to enhancement or variation in the size of the file. This has been illustrated in the snapshot in Fig. 9.

Analysis for Scenario J - While performing the experiment, the hash value changes because when a MS-Word file is printed, the value of the metadata attribute 'Last Printed On' gets modified. Since the 'last printed date' is a part of the Application Metadata, the hash value of the file therefore changes when it is printed. However, this is not true in case of an image, audio, video file or PDF file, which does not contain this property as a part of its metadata [21].

Analysis for Scenario K - Based on the results in Scenario K, there is no change in the hash value of the file. This is because there is no change in the content of the evidence file. Although, the Modification, Accessed and Creation (MAC) date and timestamps of the file vary when copied from one version of an operating system to another, it does not affect the file contents, since MAC date and timestamps belong to system metadata [20]. This keeps the hash value of the file unchanged. Different types of files (for e.g. image, audio and video files) subjected to the same experiment also show unchanged hash value. The same is shown in the snapshot in Fig. 10.

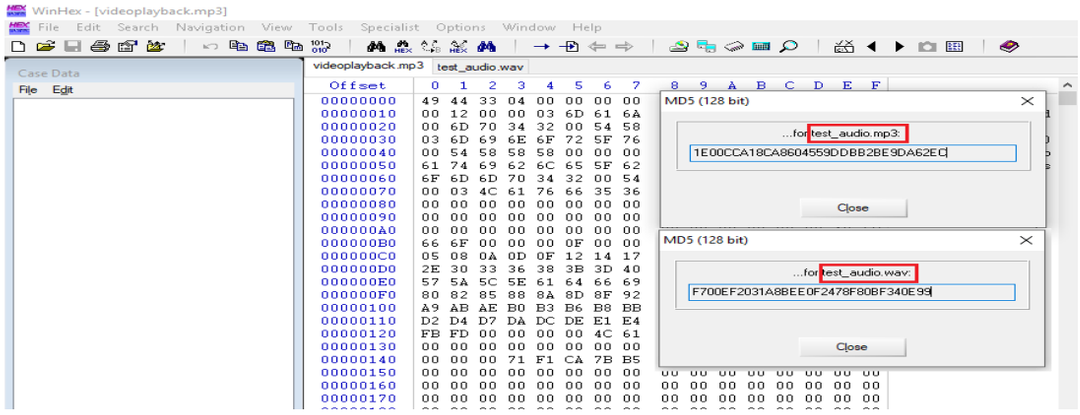

Fig. 9 Change in hash value on storing an audio file in two different file formats (Scenario I)

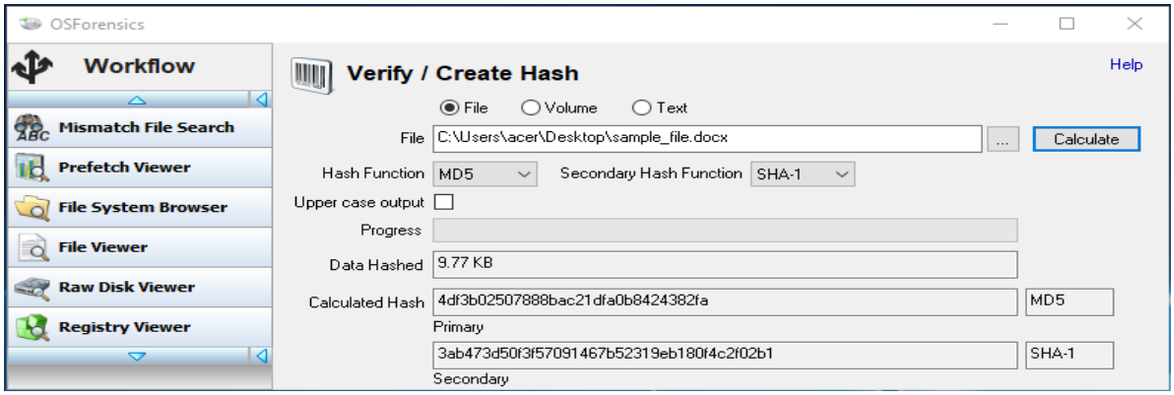

Fig. 10 No change in hash value when the file is moved from version of OS to another (Scenario K)

V. Conclusions

This research work has been carried out with the objective of performing a comparative analysis of the integrity of digital evidence using different tools and techniques. 11 different scenarios including sample data were prepared to compute the hash value before and after performing specific activity in each sample. The objective was to determine whether any of the scenarios show a change in hash value or not and, identify the reason for the change subsequently.

Hence, based on the research work, substantia observations and conclusions have been drawn. The hash value of digital evidence is not just dependent on its content, but other factors such as application metadata, file compression, encoding and encryption also play a major role in changing the hash value. Apart from modifying the contents of data file, steganography, alteration in file properties (which is a part of the application metadata), printing the file and using different versions of the same native application (i.e. MS-Word application used for creating the data file samples), which does not alter the file contents, but modifies the hash value. However, on the other hand it was also concluded that, altering the filename and file extension (which is a part of the system metadata), or storing the data file in different versions of an operating system, did not affect file integrity. Compression is another factor, which affects the file hash. Other than performing file compression, storing the data file in different file formats encodes as well as compresses the data of the file, resulting in new hash value. This applies to both MS-Word file being saved in PDF or multimedia files being stored in different formats. Lastly, encrypting any file by using a password with the intention of securing the document also modifies the file's hash.

From the results of the experiments conducted in the research work, it is concluded that any individual involved with the handling and analysis of data files (digital evidences) should be aware and have the required knowledge about the possible changes that might occur during investigation/examination. Such changes might be unintentional or accidental, however, at the same time can prove to be detrimental to the facts of the case even if the forensic examiner has not altered any contents of the evidence. This can possibly render the evidence inadmissible if the hash value does not match. Therefore, as a part of Standard Operating Procedure and best practices, forensic imaging of the original exhibit/evidence should be done and no work should be conducted on the original evidence should be performed. The results obtained in this empirical study will be useful for the forensic fraternity and law enforcement officers, who are involved in investigation consisting of digital evidences.

References

[1] R. Kissel, "Glossary of Key Information Security Terms," National Institute of Standards and Technology, Gaithersburg, MD, USA, Rep. no. 7298 Rev. 2, NIST Interagency/Internal Report (NISTIR). [Online]Available: https://doi.org/10.6028/NIST.IR.7298r2

[2] T. Grance, S. Chevalier, K. K. Scarfone and Hung Dang, "Guide to Integrating Forensic Technique into Incident Response," National Institute of Standards and Technology, Gaithersburg, MD, USA, Rep. no. 800-86, Special Publication (NIST SP). [Online]Available: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=50875

[3] C. Hosmer, "Proving the Integrity of Digital Evidence with Time," *Int. J. Digit. Evid.* vol. 1, no. 1, pp. 1-7, 2002.

[4] K. Raychaudhuri, "A Comparative Study of Analysis and Extraction of Digital Forensic Evidences from Exhibits using Disk Forensic Tools," *Int. J. Cyber-Secur. Digit. Forensics*, vol. 8, no. 3, pp. 194-205, Sep. 2019, doi: 10.17781/P002608.

[5] J. Williams, *ACPO Good Practice Guide for Digital Evidence*, Version 5.0, (2012). [Online] Available: https://www.digital-detective.net/digital-forensics-documents/ACPO_Good_Practice_Guide_for_Digital_Evidence_v5.pdf

[6] A. Menezes, P. C. Oorschot and S. Vanstone, *Handbook of Applied Cryptography*, Boca Raton, FL, USA: CRC Press, 1997.

[7] K. Raychaudhuri and M. G. Christopher, "An Empirical study to determine the role of file-system in modification of hash value," *Int. J. Cybersecur. Intell. Cybercrime*, vol. 3, no. 1, pp. 24-41, Feb. 2020.

[8] J. Ćosić and M. Bača, "(Im)proving chain of custody and digital evidence integrity with time stamp," In *33rd Int. Conv. MIPRO*, Croatia, 2010, pp. 1226-1230.

[9] J. R. Vacca, *Computer Crime Scene Investigation*, Boston, MA, USA: Charles River Creative, 2002.

[10] J. Richter, N. Kuntze and C. Rudolph, "Security Digital

Evidence," in *2010 Fifth IEEE Int. Workshop Syst. Approaches Digit. Forensic Eng.*, USA, 2010, pp. 119-130, doi: 10.1109/SADFE.2010.31.

[11] K. Kumar, S. Sofai, S.K. Jain and N. Aggarwal, "Significance of Hash Value Generation in Digital Forensic: A Case Study," *Int. J. Eng. Res. Dev.*, vol. 2, no. 5, pp. 64-70, July 2012.

[12] V. Roussev, "Hashing and Data Fingerprinting in Digital Forensics," *IEEE Secur. Priv.*, vol. 7, no. 2, pp. 49-55, March-April 2009, doi: 10.1109/MSP.2009.40.

[13] R. Rivest, *The DS5 Message-Digest Algorithm*, (1992). Accessed: Jan. 20. [Online]Available: https://www.ietf.org/rfc/rfc1321.txt

[14] R. Chaves, et al. "Secure Hashing: SHA-1, SHA-2, and SHA-3," in *Circuits and Systems for Security and Privacy*, F. Sheikh and L. Sousa, Ed, Boca Raton, FL, USA: CRC Press, 2018, p. 382.

[15] *Secure Hash Standard (SHS): Federal Information Processing Standards Publication*, FIPS PUB 180-4, National Institute of Standards and Technology, MD, USA. 2012.

[16] G. C. Kessler, "A Study of Forensic Imaging in the Absence of Write-Blockers," *J. Digit. Forensic Secur. Law*, vol. 9, no. 3, pp. 51-58, 2014, doi: 10.15394/jdfsl.2014.1187

[17] NIST, "Computer Forensics Tools & Techniques Catalog". Accessed: Dec. 21. [Online]Available: https://toolcatalog.nist.gov/

[18] *OSForensics 8.0*, PassMark Software. Accessed: Dec. 20. [Online]Available: https://www.osforensics.com/

[19] *WinHex: Computer Forensics & Data Recovery Software, Hex Editor & Disk Editor*, X-Ways. Accessed: Dec. 15. [Online]Available: https://x-ways.net/winhex/

[20] Craigball, "A Hash of It," Mar. 2012. Accessed: Dec. 10. [Online]Available: https://craigball.net/2012/03/05/

[21] K. A. Schuler, *E-discovery: Creating and Managing an Enterprisewide Program: A Technical Guide to Digital Investigation and Litigation Support*, Burlington, MA, USA: Syngress Media, 2008.

[22] Lossless Compression: An Overview. Accessed: Dec. 8. [Online]Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/data-compression/lossless/index.htm

[23] Dose the MD5 change from encryption?. Accessed: Dec. 5. {online]Available: https://stackoverflow.com/questions/21467727/does-the-md5-change-from-encryption.