# Chapter 4
# Uncovering Limitations of E01 Self-Verifying Files

**Jan Krasniewicz**
*Birmingham City University, UK*

**Sharon A. Cox**
*Birmingham City University, UK*

## ABSTRACT

*In computer forensics, it is important to understand the purpose of evidence file formats to maintain continuity of acquired data from storage devices. Evidence file formats such as E01 contain embedded data such as cyclic redundancy check (CRC) and hash values to allow a program to verify the integrity of the data contained within it. Students in computer forensics courses need to understand the concepts of CRC and hash values as well as their use and limitations in evidence files when verifying acquired data. That is the CRC and hash values in evidence file only verify the acquired data and not the evidence file per se. This important difference in E01 files was highlighted by showing students an anomaly in E01 files where certain bytes can be changed in E01 files without detection by computer forensic software using the embedded CRC and hash values. The benefit to students is that they can see the advantages of self-verification and limitations of what is verified giving the opportunity for a deeper understanding of evidence files and good practice.*

## INTRODUCTION

Teaching good practice in computer forensics is important to understand the correct operation and limitations of computer forensic hardware and software. One task is to demonstrate the self-verification feature of evidence file formats such as the EnCase E01 file format that contains an image of acquired data. The E01 file contains the data plus extra data in the form of hash values and Cyclic Redundancy Check (CRC) values used by computer forensic software to check the data contained within the file has not been tampered with. Students are taught how to carry out this task and verify the file by making a change to the generated file and observing mismatches between hash values and Cyclic Redundancy Check (CRC) values generated when the data was copied and when the file is loaded into computer forensic

DOI: 10.4018/978-1-5225-7492-7.ch004

software. Whilst creating teaching materials for students to carry out this task an anomaly was identified in one of the forensic file formats, the E01 format, commonly used by practitioners. The anomaly allows changes to be made to certain bytes within the file that are not detected by computer forensic software when verified by the associated hash and CRC values. This paper describes the anomaly in the file format, discussed the implications for relying on the self-verification feature of the E01 file format and concludes on methods to make any change to the file contents detectable.

## Background

One of the first tasks before conducting a computer forensic analysis of data is to make a forensically sound copy of the data stored on, for example, a hard disk drive. This task forms the acquisition stage of an investigation. By "forensically sound" it is meant that the copying process does not alter the source data resulting in an exact copy of the data (Casey, 2007). This task involves making a bit-for-bit copy of the data and using a method that assists in determining the integrity of the resulting copy as part of the chain of custody.

It is important to be able to determine that the copy of data has not been changed before it is analysed. It is common practice and recommended by organisations such as the Association of Chief Police Officers (ACPO) and National Institute of Standards and Technology (NIST) to use a mathematical function to calculate a unique value for the data at the time of copying. Examples of mathematical functions used to check the integrity of data are Cyclic Redundancy Check (CRC) and cryptographic hash (Schneier, 1996). These functions are implemented in computer programs to compute a value from a computer file or entire contents of a storage device. The value is recorded so that whenever the digital evidence is analysed the value is recomputed and compared to the original value.

Computer programs have been developed to automate the copying process and calculate the integrity values for the acquired data. These values are stored within the resulting copy of the data. Storing the integrity values within the file allows the copy to be *self-verifying* when analysed with computer forensic software. When the copy is used by a computer forensic software application, such as Guidance Software's EnCase and AccessData's FTK, the application recalculates the unique value and then compares it with the value stored in the file. The program displays a warning message when the original and calculated values are different as this difference indicates the file has changed, the change could be as a result of corruption or it could be more sinister due to a deliberate change by an individual.

This paper considers the integrity values stored in the copy of the data, commonly known as the image file or digital evidence container file (Common Digital Evidence Storage Format Working Group, 2006). The paper describes mathematical functions used to calculate the integrity values and how the property of the function allows data to be validated. The paper then describes how a practical exercise to demonstrate self-verification features to students identified an anomaly where it was possible to change a byte within the file without the self-verification detecting that the copy had been changed. The paper explains why additional integrity values should be calculated based on the entire data, copy and integrity values combined, to further enhance confidence the copy has not been altered after it has been made.

## Hash Functions in Computer Forensics

Hash functions are one approach to solving the problem in computer systems of being able to perform a fast lookup of data in a data store such as RAM or disk drive (Knuth, 1998). The hash function takes