Chapter 21

AUTOMOBILE EVENT DATA RECORDER FORENSICS

Nathan Singleton, Jeremy Daily and Gavin Manes

Abstract Automobile event data recorders (EDRs) provide vital information for reconstructing traffic crashes. This paper examines the primary issues related to evidence recovery from EDRs and its use in crash reconstruction. Recommendations related to the use of EDR data in court proceedings are also presented.

Keywords: Automobile event data recorders, evidence extraction

1. Introduction

Vehicle collisions cause significant personal injuries and financial losses on a daily basis. Several techniques and tools have been developed for traffic crash reconstruction, which involves the scientific interpretation of physical evidence to determine the events that precipitated a crash [3]. Event data recorders (EDRs) in passenger vehicles provide detailed data about vehicular operation and state. EDR data, which varies according to the make, model and year of vehicles, can augment the physical evidence used in crash investigations.

Methods for retrieving data stored in EDRs are generally proprietary in nature. The Bosch crash data retrieval (CDR) system is used for automobiles from General Motors, Ford, Chrysler and partner companies. Data contained in EDRs of other vehicles is usually recovered using manufacturer-specific hexadecimal translation tools (HTTs).

This paper focuses on the recovery of digital evidence from EDRs. A case study involving a 2001 Chevrolet 1500 pickup is used to clarify recovery techniques and digital forensic practices.

2. Automobile Event Data Recorders

Modern automobile event data recorders (EDRs) record pre-crash vehicle performance data and system status, accelerations during a crash, safety restraint system data, driver control inputs and post-crash information such as automatic crash notification. The development of EDRs can be traced back to 1990, when General Motors introduced the diagnostic energy reserve module (DERM) to record data about airbag systems. Evidence from DERMs has been used in litigation related to the design and operation of airbag systems [8].

The next generation of EDRs, called sensing and diagnostic modules (SDMs), were introduced in 1994. These modules were designed to perform three main functions in the following prioritized order: (i) deploy airbags in the event of a crash, (ii) perform airbag system diagnostics, and (iii) monitor and record system and event data during an "event." An "event," in this context, is a sudden change in vehicle acceleration that initiates an algorithm in the airbag module. Depending on the decision logic, the event may or may not cause the airbags to deploy.

Early SDMs also recorded system status data related to seat belt use and accelerations during a crash. In 1999, SDMs began to record pre-crash data such as vehicle speed, engine rpm, brake light switch status, throttle position, warning indications and seat belt use. The data is measured external to the SDM and is transferred to the module via a vehicle system bus.

In 2006, the U.S. National Highway Traffic Safety Administration (NHTSA) estimated that about 64% of new passenger vehicles were equipped with EDRs [9]. EDR use is rising due its voluntary inclusion in vehicles by manufacturers. Indeed, it is rare for automobiles manufactured in 2008 not to have some form of EDR. There is no standard location for positioning an EDR; however, it is usually located inside the vehicle cabin, near the centerline of the vehicle or under/in one of the front seats. Physical removal of an EDR typically requires the disassembly of a vehicle's interior.

In a typical EDR, vehicle system data and crash information are continuously stored in a volatile data buffer during normal operation. Depending on the module and the type of event, the volatile data may be flashed to an EEPROM. In the event of an airbag deployment in a General Motors vehicle, this data is permanently written to the EEPROM (and the module has to be replaced). However, if the airbag is not deployed, EEPROM data is cleared after the SDM is turned on 250 times. These characteristics vary for modules from different manufacturers; interested readers are referred to [1] for additional information about SDMs used in General Motors automobiles.

Driven by the need to ensure the accuracy, reliability and privacy of automobile event data, the Society of Automotive Engineers (SAE) and the Institute for Electrical and Electronics Engineers (IEEE) joined with NHSTA to form working groups to address policy issues and standardization [6]. Interested readers are referred to the NHTSA website [7] for information about these working groups and their activities.

Historically, the primary concern has been the reliability of automobile event data as it pertains to supporting physical evidence in crash investigations [10]. Consequently, the majority of studies related to EDRs have focused on using data after it has been recovered and decoded [2, 4, 11]. However, it is just as important to ensure that event data used in legal proceedings accurately reflects the data captured by the EDR. This paper is motivated by the need to develop sound forensic techniques for evidence recovery from EDRs.

3. Crash Data Retrieval System

General Motors initiated the development of the crash data retrieval (CDR) system; this system is now also licensed to Ford and Chrysler. At this time, a CDR can only download data from EDRs in select General Motors automobiles manufactured after 1994, Ford vehicles built in 2001 or later, and Chrysler automobiles from 2004 onwards.

3.1 System Connections

The system connections for data recovery are presented in Figure 1. A standard nine-pin RS-232 cable is used to connect a CDR interface module to a computer. However, a special 15-wire cable is required to connect the interface module to the EDR. The interface module end has a modified 15-pin serial connector with only the pins required to make the EDR connection, typically two for power and one for the signal. The other end of the cable has a specialized connector that mates to the EDR or directly connects to the OBD-II or DLC diagnostic ports of an automobile.

EDR connections may be established in two ways. In the field, the primary method is to connect through the OBD-II or DLC diagnostic ports located under the driver side dashboard. However, this requires the EDR to have electrical power. The second method requires direct access to the EDR; this is used performed when the electrical system is non-functional.

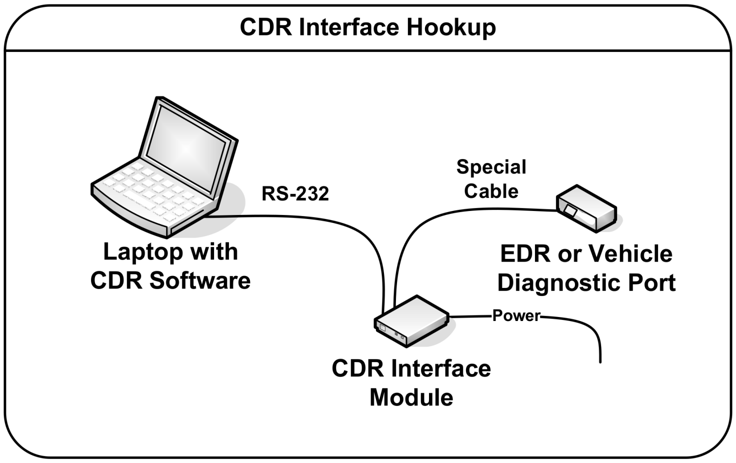

Figure 1. Data recovery system connections.

Having established the CDR-EDR connections, the CDR interface module is connected to a computer with CDR software via an RS-232 cable. Power is applied to the EDR through a lead attached to the CDR interface module (Figure 1).

3.2 Data Recovery

After the CDR software (version 2.8) is initialized, it communicates with the serial communications port using the standard `41 54 0D` (AT) command. The software then checks for the presence of the CDR interface module by sending the data: `D3 56 00 D7`. Upon receiving a satisfactory response, the software opens an existing file or starts a new case depending on the investigator's selection.

When a new case is started, the investigator is required to enter case-specific information such as the vehicle identification number (VIN), investigator name, case number, investigation date, crash date and comments (e.g., accident location and details). The investigator may save the report as a file; the default file name is the VIN.

After the case-related information has been entered, the CDR software sends an initial polling signal, `53 56 47 10`, which retrieves EDR variant identifier code from the EDR register. For example, a 2001 GMC Sierra 1500 EDR returns the code `88 59 91 17 00 00 77`; the `91 17` sequence corresponds to the specific SDMG2000 EDR, which is a GM product (G2000 refers to the version). On the other hand, a 2001 Oldsmobile Alero EDR returns the sequence `88 59 08 23 00 00 F4`, where the `08 23` corresponds to an SDMG2001 EDR. The EDR variant identifier code specifies the type of EDR installed, which dictates the cabling and CDR setup requirements and the specific commands that can be used.

Next, the CDR sends a dump command corresponding to the EDR model. For example, the SDMG2000 dump command is `47 56 01 62` while the SDMG2001 dump command is `47 56 06 5D`. The dump command downloads the non-volatile memory from the EDR to the CDR interface module; the sequence `D0 56 47 93` is sent to the CDR when the process is completed. At this point, the recovered data is stored in volatile memory on the CDR interface module. The CDR then sends a command to transfer approximately half of the downloaded data for processing. In the case of the SDMG2000 EDR, this command is `EF 5A 01 1F 00 80 00 17`. The `EF 5A 01 1F` portion is the download command; `00` marks the starting location and `80` is the ending location. This command returns the following hexadecimal data:

```
EF D6 01 91 17 00 00 A7 18 41 53 30 33 34 30 4B 46 33 42 39 32
00 15 76 31 80 A3 A5 A4 F8 AC 00 03 A4 34 80 83 81 85 70 FF 00
FA FA FA FA FA FA FA FA FA FA FA FA FA FA FF 02 00 00 00 FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF 81
```

Data at the beginning of the file (`EF D6 01`) and at the end of file (`81`) is not downloaded by the CDR; however, it is appended to the data in the CDR interface module. The code `91 17` following the beginning of the file marker corresponds to the EDR type. The rest of the data is downloaded with sequence `EF 5A 01 1F 80 5E 00 B9`. Note that `80` and `5E` mark the starting and ending locations, respectively.

The CDR interface module then transmits the following data:

```
EF B4 01 FF FF FF FF FF FF 80 00 00 FF 80 FE FF BF FF FF FF FF
FF FF FF FF FF FF 7C 04 03 01 01 02 00 00 00 00 00 00 00 00 FF
FF FF FF FF 0A 10 00 61 70 70 6E 6C 6A 00 80 00 00 73 73 73 73
00 20 20 20 20 20 00 F8 25 FE 00 00 00 04 00 FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF 98
```

The sequences `EF B4 01` and `98` are the beginning of file and end of file markers, respectively; the two markers are appended to the data by the CDR interface module. The entire retrieval process, which begins with the CDR sending the EDR a dump command, is repeated two more times for a total of three passes.

3.3 Retrieved Data

After the CDR software completes the retrieval process, it analyzes the EDR data and generates a report. The report is placed in a temporary file, which is deleted unless it is explicitly saved prior to exiting the program. A saved report is stored in a proprietary format (`*.CDR`) or as a `*.pdf` file. In addition to the data in the report, the `*.CDR` file contains formatting data and error checking data (hash value and field size counts).

Analysis of the CDR file reveals that the data is stored in a simple file format. A common element in the hex data is the sequence `0D 0A`, which is used as a delimiter to separate fields and as the carriage return/line feed. Additionally, hex `20` is used as the space character and to fill fixed-size fields.

Three sources of data are contained in the dump: user-entered data, CDR-supplied data and hash values. The user-entered data includes the VIN, investigator's name, case number, comments, etc. The user-entered fields, "Investigator," "Case Number," "Investigation Date" and "Crash Date" have a maximum of 64 characters. The "Comments" field can have a variable amount of data; preceding the actual data is a data-size marker of two bytes that indicates its length.

The CDR-supplied data is also variable in size and incorporates a data-size marker at the beginning of each field. This is most likely due to the fact that the "Interface Used to Collect Data" field has carriage returns inserted in the data, which have the same hex code as delimiters. When the size of the field is calculated, all carriage returns/line feeds in the entered data are counted as two bytes and ignored as field delimiters.

Two hash values are included in the hex dump. The hash value that appears toward the middle of the hex is used to ensure the data has not been altered. The last hex code value corresponds to the Reporting Program Verification Number and the Collecting Program Verification Number, both of which are displayed in the CDR report.

The first 17 bytes of data recovered from a 2001 GMC Sierra 1500 contain the VIN. Once the program is running, this is the first information requested as user input by the CDR software:

```
32 47 54 45 43 31 39 54 35 31 31 32 34 34 39 38 39
 2  G  T  E  C  1  9  T  5  1  1  2  4  4  9  8  9
```

Following the VIN is the delimiter `0D 0A`, which is used to separate fields. The next field is inserted by the CDR software and contains information about the EDR type and model. Note that hex code `20` is used to fill the field:

```
53 44 4D 47 32 30 30 30 20 20 20 20 20 20 20 20
```

```
 S  D  M  G  2  0  0  0

39 31 31 37 20 20 20 20 20 20 20 20 20 20 20 20

 9  1  1  7
```

This data is followed by the investigators's name and other user-entered data such as the investigation date. The actual EDR data appears later and is preceded by a size field in big endian. Note that `A7` below computes to 167 bytes, which is the number of bytes stored on the EDR including the initial padding of six sets of zeros. The reason for the padding is unknown.

```
00 00 00 00 00 00 91 17 00 00 A7 18 41 53 30 33 34 30 4B 46 33
42 39 32 00 15 76 31 80 A4 A6 A5 F8 AD 00 03 A4 34 80 84 81 85
70 FF 00 FA FA FA FA FA FA FA FA FA FA FA FA FA FA FF 02 00 00
00 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF 80 00 00 FF 80 FE FF
BF FF FF FF FF FF FF FF FF FF FF 7C 04 03 01 01 02 00 00 00 00
00 00 00 00 FF FF FF FF FF 0A 10 00 61 70 70 6E 6C 6A 00 80 00
00 73 73 73 73 00 20 20 20 20 20 00 F8 25 FE 00 00 00 04 00 FF
FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

The file have two hash values that are used to verify the integrity of the data. The exact hash functions are unknown. Based on experimentation, including trial-and-error attempts at determining the hash functions used by manipulating the data, it appears that the data is rehashed every time the CDR software is asked to open the document. The new hash value is then compared with the old value; if the hash comparison fails, an error is reported and the program exits.

Although this file format appears to be simple, no public description of the format exists and there is no method to review many of these fields using the CDR software. Furthermore, the human-readable reports do not contain hash values or any file verification data that a digital forensic examiner would come to expect.

4. Digital Forensic Issues

EDR evidence must be introduced in court by an expert witness. The expert must provide testimony that relates to knowledge or experience beyond that possessed by lay persons. Furthermore, the individual must have specialized knowledge, skill, training or education regarding the subject matter of the testimony, which must be based on reliable scientific, technical or other specialized information.

EDR data introduced in court must pass the well-known Daubert and Frye tests of scientific evidence. The first criminal case to introduce

EDR data as evidence was Colorado v. Cain in 2002; since then, numerous criminal and civil cases have employed EDR evidence [5]. Due to the increased use of EDR data, many jurisdictions have create statutes regarding EDRs. This section examines some of the major issues pertaining to EDR data and data collection that may impact the quality of the recovered evidence.

4.1 Missing and Uninterpreted Data

The CDR report includes downloaded hex data with the presumed register numbers from where the data was pulled. The registers for the 2001 GMC Sierra 1500 system contain six bytes of data, but those for other automobile EDRs have fewer bytes. Also, certain segments of these registers are missing. The system documentation does not explain this anomaly.

The CDR report also displays uninterpreted data, possibly proprietary information such as deployment thresholds. However, when the data transfer is monitored with a sniffer, discrepancies appear between different passes. The method used by the system to select and insert data is unknown, making it difficult to verify the repeatability of extraction process and the results.

Despite these somewhat disturbing findings, the information provided in a CDR report appears to be complete. Without additional techniques and translation tools, it must be assumed that all data that is available is being translated. However, it would be very useful to have an alternative HTT to compare the results.

4.2 Data Collection Discrepancies

Discrepancies were observed in the bytes obtained during the three data dump passes. The number of discrepancies increased when multiple downloads were performed in short order (each download takes three to five minutes). The changed bytes were found in the first portion of the download from the CDR interface module.

Analysis revealed that the values were random and not due to a clock or counter. For example, the CDR interface module sent the same data during the first two of six runs. However, there were discrepancies when comparing the three passes in each run. The remaining four runs also contained discrepancies. These discrepancies were discovered by comparing the hex values in the final CDR report to those collected by a sniffer.

Additionally, the `*.CDR` file does not contain the original data collected during the three passes. This implies that the data read from the EDR is transformed by the CDR software (possibly by performing certain calculations on the data) before being displayed in the CDR report.

4.3 Unexplained Methods

The final data in a CDR report is processed by an algorithm before it is displayed. In fact, the following statement is provided at the beginning of a CDR report:

> Once the crash data is downloaded, the CDR tool mathematically adjusts the recorded algorithm forward velocity data to generate an adjusted algorithm forward velocity change that may more closely approximate the forward velocity change the sensing system experienced during the recorded portion of the event. The adjustment takes place within the downloading tool and does not affect the crash data, which remains stored in the SDM. The SDM Adjusted Algorithm Forward Velocity Change may not closely approximate what the sensing system experienced in all types of events.

It is important to note that the description of the algorithm is not provided by the manufacturer. Also, terms such as "more closely approximate" are undefined. Furthermore, the original data stored in the SDM is not displayed in a human-readable format and it may not be possible to verify the data.

4.4 Evidence Identifiers

Initially, it was believed that the CDR software employed VIN data ("World Manufacturer Identifier," "Vehicle Attributes," "Model Year") to determine the type and version of the EDR being read. However, we discovered that any data may be entered as long as the "World Manufacturer Identifier" corresponds to a manufacturer supported by the CDR and follows the VIN formatting requirements. Thus, it is possible to spoof a CDR system in an attempt to download data from a different module than intended.

4.5 Unwiped Media

There are indications that the CDR interface module memory is not wiped between downloads. We attempted to verify this fact by experimentation. Power was applied to the CDR interface module and the data in CDR memory was requested prior to performing a download of the EDR.

When the following data was sent by the CDR interface module to the SDMG2000 EDR (see Section 3.2):

```
EF 5A 01 1F 00 80 00 17
```

the following block of data was returned:

```
EF D6 01 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 10 11
12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25 26
27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35 36 37 38 39 3A 3B
3C 3D 3E 3F 40 41 42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50
51 52 53 54 55 56 57 58 59 5A 5B 5C 5D 5E 5F 60 61 62 63 64 65
66 67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76 77 78 79 7A
7B 7C 7D 7E 7F 7A
```

When the following data was sent by the CDR interface module:

```
EF 5A 01 1F 80 5E 00 B9
```

the following block of data was returned:

```
EF B4 01 80 81 82 83 84 85 86 87 88 89 8A 8B 8C 8D 8E 8F 90 91
92 93 94 95 96 97 98 99 9A 9B 9C 9D 9E 9F A0 A1 A2 A3 A4 A5 A6
A7 A8 A9 AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9 BA BB
BC BD BE BF C0 C1 C2 C3 C4 C5 C6 C7 C8 C9 CA CB CC CD CE CF D0
D1 D2 D3 D4 D5 D6 D7 D8 D9 DA DB DC DD 49
```

This suggests that the memory in the CDR interface module is not reset and unknown data is available for download as authentic data.

5. Recommendations

EDRs provide vital data for reconstructing traffic crashes. However, during a crash reconstruction, it is important to also consider the physical evidence that is always present instead of relying solely on digital evidence.

Digital forensic professionals should be cognizant of the following recommendations related to EDRs and EDR data:

- Maintain a detailed record of the chain of custody of an EDR. This specifically includes documenting all physical extractions from the module because there are no unique identification marks on the EDR that tie it to the vehicle.

- Preserve and maintain the original human-readable report as this document becomes the evidence used in the legal context. Note that the original data in the EDR device could be erased, overwritten or corrupted. For example, an SDM does not write its memory to a permanent record for a non-deployment event and the memory is erased after the unit is turned on a certain number of times.

- Understand when and how the data is erased, overwritten or corrupted. For example, some powertrain control modules on Ford vehicles retain 25 seconds of data (including vehicular speed) in a circular buffer. However, the data is not "locked" in the event of a crash and the reapplication of power to the module causes the data to be overwriten. Note that modules from different manufacturers vary considerably in terms of their operational characteristics.

- Learn how current vehicular technology is used to generate data for the EDR. Many EDRs rely on external sensors and advanced vehicular systems for data. For example, the speed sensor actually measures the rotation of the driveshaft, not the true speed. This means that the speed of a vehicle that is sliding sideways on ice is not reported correctly in the EDR.

6. Conclusions

EDR data is extremely valuable in reconstructing the physical events leading to automoble crashes. However, the case study involving the extraction of data from a 2001 Chevrolet 1500 pickup EDR using the Bosch CDR system has revealed several problems that may impact evidentiary quality. The problems include missing data and uninterpreted data, data collection discrepancies, unexplained methods, and issues with evidence identifiers and unwiped media. Investigators should be aware of these problems and should use sound forensic procedures and tools to ensure that EDR evidence is not excluded in legal proceedings.

References

[1] A. Chidester, J. Hinch, T. Mercer and K. Schultz, Recording automotive crash event data, *Proceedings of the International Symposium on Transportation Recorders*, 1999.

[2] J. Correia, K. Iliadis, E. McCarron and M. Smolej, Utilizing data from automotive event data recorders, *Proceedings of the Twelfth Canadian Multidisciplinary Road Safety Conference*, 2001.

[3] J. Daily, N. Shigemura and J. Daily, *Fundamentals of Traffic Crash Reconstruction*, Institute of Police Technology and Management, University of North Florida, Jacksonville, Florida, 2006.

[4] R. Fay, R. Robinette, J. Scott and D. Deering, Using event data recorders in collision reconstruction, *Proceedings of the Society of Automotive Engineers World Congress and Exhibition*, SAE Technical Paper Series 2002-01-0535, Society of Automotive Engineers, Warrendale, Pennsylvania, 2002.

[5] Harris Technical Services, EDR Case Law, Port St. Lucie, Florida (harristechnical.com/cdr5.htm).

[6] IEEE Vehicular Technology Society, IEEE Project 1616: Draft Standard Motor Vehicle Event Data Recorders, Piscataway, New Jersey (grouper.ieee.org/groups/1616/home.htm).

[7] National Highway Traffic Safety Administration, Event Data Recorder (EDR) Applications of Highway and Traffic Safety, U.S. Department of Transportation, Washington, DC (www-nrd.nhtsa .dot.gov/edr-site).

[8] U.S. Court of Appeals (Sixth Circuit), Harris v. General Motors Corporation, *Federal Reporter Third Series*, vol. 201, pp. 800–805, 2000.

[9] U.S. Government, Event Data Recorders, Department of Transportation, National Highway Traffic Safety Administration, *Federal Register*, vol. 71(166), pp. 50998–51048, 2006.

[10] S. van Nooten and J. Hrycay, The application and reliability of commercial vehicle event data recorders for accident investigation and analysis, *Proceedings of the Society of Automotive Engineers World Congress and Exhibition*, SAE Technical Paper Series 2005-01-1177, Society of Automotive Engineers, Warrendale, Pennsylvania, 2005.

[11] C. Wilkinson, J. Lawrence, B. Heinrichs and D. King, The accuracy and sensitivity of 2003 and 2004 General Motors event data recorders in low-speed barrier and vehicle collisions, *Proceedings of the Society of Automotive Engineers World Congress and Exhibition*, SAE Technical Paper Series 2005-01-1190, Society of Automotive Engineers, Warrendale, Pennsylvania, 2005.