

On the Digital Forensics of Heavy Truck Electronic Control Modules

James Johnson, Jeremy Daily, and Andrew Kongs
University of Tulsa

ABSTRACT

Concepts of forensic soundness as they are currently understood in the field of digital forensics are related to the digital data on heavy vehicle electronic control modules (ECMs). An assessment for forensic soundness addresses: 1) the integrity of the data, 2) the meaning of the data, 3) the processes for detecting or predicting errors, 4) transparency of the operation, and 5) the expertise of the practitioners. The integrity of the data can be verified using cryptographic hash functions. Interpreting and understanding the meaning of the data is based on standards or manufacturer software. Comparison of interpreted ECM data to external reference measurements is reviewed from the current literature. Meaning is also extracted from interpreting hexadecimal data based on the J1939 and J1587 standards. Error detection and mitigation strategies are discussed in the form of sensor simulators to eliminate artificial fault codes. A transparent process for data gathering and handling is discussed.

The needs for improved techniques are motivated through examples of manipulated data and an analysis of potential opportunities that exist to alter the data. As an example, a step by step process of changing the records of a DDEC Reports. XTR file is provided. A detailed examination of resetting the ECM clock is also presented, which motivates a design of a hardware write-blocking device.

Some recommendation to provide more forensically sound records of ECM data is outlined. The strategy records and hashes network traffic to create a verification technique for later use. The data is then used in a replay algorithm so the diagnostic software can recreate the information from the forensic copy of the network traffic. Finally, application of the digital forensics beyond accident reconstruction is noted.

CITATION: Johnson, J., Daily, J., and Kongs, A., "On the Digital Forensics of Heavy Truck Electronic Control Modules," *SAE Int. J. Commer. Veh.* 7(1):2014, doi:10.4271/2014-01-0495.

INTRODUCTION

Each year, many thousands of heavy vehicles are involved in traffic accidents; in 2008, 380,000 trucks were involved in accidents [1]. Litigation connected with these crashes can ensue and, in recent years, this litigation has come to depend more and more heavily on electronic event data recorded on the vehicle's electronic control modules (ECMs). Much like passenger vehicle event data recorders (EDRs), this evidence includes speed records and other event data that can help accident reconstructionists determine what transpired during the event. Since the digital data from these devices may end up as evidence in court it should be forensically sound.

Modern heavy vehicle electronic control modules (ECMs) store incident data that may be valuable to accident reconstructionists in corroborating or clarifying physical evidence. Practices for extracting this information, however, are immature relative to other areas of digital forensics and require great care on the part of investigators in order to properly preserve and present evidence.

Motivation

Like other digital evidence, heavy vehicle digital event data should be held to a standard of forensic soundness, which is a series of principles that ensure that forensic evidence is handled in a secure manner that guards against unintentional, and intentional, alteration or misinterpretation. Currently, evidence is extracted from heavy truck systems using software that was not originally designed to be forensic software, and is not particularly suited to the task. While evidence can be collected using this data in a forensically sound manner, doing so requires diligence on the part of the investigator and mistakes can result in evidence being dismissed. Additionally, the current methods by which data are stored are not particularly resistant to tampering; encryption, when used, is weak and no mechanism for integrity checking is implemented. To date, there has not been any literature that links the concepts developed in investigation practice of personal computers to the digital evidence obtained from heavy duty

ECM's. Some of the solutions presented in this paper could help first responders gather more crash data without the fear of spoiling or losing the data.

Paper Organization

First, the paper will review digital forensics concepts that apply to any kind of digital evidence, including heavy vehicle event data recorder (HVEDR) data. The first concept reviewed is forensic soundness, a term for the legal fitness of evidence. We then give a brief introduction to cryptographic methods for ensuring the integrity of digital evidence. We then discuss the format in which HVEDR data are transmitted and stored. The final item in the background section highlights some potential sources of error in existing evidence extraction processes and current methods by which those risks are mitigated.

Next is a discussion of trustworthiness of data extracted by current methods, including trust levels of vehicle network data and data generated by diagnostic software. Some previously unpublished methods by which data created by current forensic processes may be altered to modify or destroy crucial HVEDR evidence data is included. This highlights areas in which current data may not be trustworthy, and provides a proof-of-concept of threats to evidence integrity.

Finally, we outline recommended practices to bring the trustworthiness of HVEDR data up to currently accepted standards for digital forensic soundness. We also introduce preliminary proof-of-concept implementations of these recommendations, including a cryptographic process for detecting data alteration and a novel method of storing and replaying HVEDR data.

DIGITAL FORENSIC CONCEPTS

A heavy vehicle electronic control module (ECM) is a specialized process control computer that may have data of interest to an investigator. While ECMs may not have keyboards and monitors, they still are computers and have central processing units, memory, storage, and a means of networking or communicating with external devices. As such, many of the principles from computer forensics can be applied to heavy vehicle ECMs.

A Special Report issued by the National Institute for Justice [2] on Electronic Crime Scene Investigation highlights the following basic forensic principles applied to dealing with digital evidence:

- Any process or procedure of collecting, transporting, or storing of digital evidence should not incur any changes to the evidence.
- Only specifically trained experts should examine digital evidence.
- Transparency during the operations of acquisition, transportation, and storage of the evidence should be maintained.

These basic tenets lay the foundation for the idea of forensic soundness. It is important to understand the term “forensically sound” as it relates to digital evidence. Many authors and professional organizations have attempted to rigorously define this concept, including the National Institute for Standards and Technology (NIST) [3], law enforcement entities such as the National Institute for Justice (NIJ) [2], and academic bodies like the International Organization on Computer Evidence [4]. The methods of extracting, analyzing, and presenting digital evidence are forensically sound if they perform the task in a manner such that the results can be used in legal proceedings with a high degree of confidence in their admissibility. The process of extracting, interpreting, and presenting evidence will be referred to in this paper as the “forensic process.”

In addition to the NIJ report, McKemmish [5] enumerates the following components of forensic soundness:

- **Meaning** is a term that denotes confidence in the interpretation of extracted evidence data.
- **Error Detection** denotes processes for detecting or predicting errors in the forensic process.
- **Transparency** means the forensic process is documented, known, and verifiable.
- **Expertise** is required for those investigators examining digital data.

Anyone who designs a forensic process that deals with heavy truck ECMs should consider these requirements and design them into the solution. Another tenet of forensic soundness is the principle of tamper resistance as pointed out by Casey [6]. If it is infeasible to prevent tampering, then the ability to detect tampering should ensure soundness of data. This section of the paper continues with discussions of tamper resistance (encryption and hashing), data meaning (decoding hex data), error detection (in terms of fault elimination), and transparency. It is beyond the scope of this technical paper to discuss the expertise required for handling digital forensic data in general.

Data Integrity

The primary tool for ensuring data integrity and detect tampering in digital forensics is a cryptographic hash algorithm. A cryptographic hash algorithm is an algorithm which generates a fixed-length fingerprint of the input data, in such a way that small changes in the data result in very large changes in the generated hash. This functionality is similar to cyclic redundancy checks (CRCs) used in networks to ensure data integrity.

Because the hash is generally much smaller than the data being hashed, the pigeonhole principle dictates that multiple input data strings must result in the same hash when run through a cryptographic hashing algorithm. A situation in which two inputs have the same hash is known as a hash collision. One important property of cryptographic hash algorithms is that generating collisions is computationally infeasible, meaning that for practical purposes it is impossible (i.e. the computation

time is measured in hundreds of years). This is important to integrity checking because if the algorithm did not have this property, someone could modify the source data in such a way that the change was undetectable by the hashing process. CRC collisions, by contrast, can be generated fairly easily. It is worth noting that the SAE standard for HVEDR data, J2728, recommends a 32-bit CRC for integrity checking [7].

Though an in-depth discussion of cryptographic hashing algorithms is beyond the scope of this paper, the interested reader can find a reasonably complete discussion in [8]. As of this writing, the SHA-256 [9] hashing algorithm is an industry standard choice for cryptographic hashing as a collision generation is still infeasible. Hashing does not encrypt data, so the original data remains unchanged.

A cryptographic hash function is an algorithm that generates a "fingerprint" of a given input. This fingerprint is generated in such a way that if a single bit of the input changes, at least half the bits in the output file are altered [8]. As such, they are commonly used to confirm that data have not been altered. A forensically sound tool should use a hash of the extracted truck evidence data to ensure that it has not been altered. This hash can be calculated and distributed to or stored by a third party immediately after the download takes place to ensure the data is authentic. This practice would eliminate any gaps in time in which someone could manipulate a digital record. Furthermore, hashing immediately at the time of extraction eliminates the need for the download to be witnessed in person or videotaped.

Meaning of the Digital Data from ECMs

If digital data are only represented as binary (or hexadecimal) listings, its meaning is lost. Therefore, it is paramount to an investigation to interpret the digital data in a meaningful way. Within the context of heavy vehicle ECM data, the tools that interpret the data are typically written by the original equipment manufacturers (OEMs) in the form of diagnostic software. Some OEMs provide other data reporting tools, like DDEC Reports and Cummins PowerSpec, to interpret the data, which can be useful for reconstructing crashes. However, these programs do not implement some of the safeguards used to ensure forensic soundness that are found in other areas of digital forensics.

Standards Based Meaning

Most of the research presented in the literature so far has focused on comparing interpreted values to external instrumentation to verify the correctness of those values. Arguably, this is all that is necessary for an expert to establish the meaning of the data. However, the interpretation of the HVEDR record is still left up to the OEM software. There are, however, standards that help interpret digital data into engineering units; the units specified in communications standards such as J1587 [10] provide clues as to the semantics of digital data. Furthermore, standards based tools

can be used to gather standardized data like odometer reading, engine hours, VIN, and component information to supplement other information in the HVDER record.

The SAE J1587 standard [10] defines the data interchange between microcomputer systems in Heavy-Duty vehicle applications. The physical network for J1587 is specified by the J1708 standard [11]. This protocol specification includes standard formats and interpretations for messages regarding the vehicle's operation.

The J1939-71 [12] standard serves a similar purpose to the J1587 standard, defining the J1939 application layer. The standard defines message structure, data format, and engineering units for various parameters. For example, the wheel-based vehicle speed message is defined as a 2-byte value with a resolution of 1/256 km/h per bit. Therefore, it is reasonable to expect that speed data will be stored internally in km/h, the field will be 2 bytes wide, and the 2-byte integer will be the wheel-based vehicle speed multiplied by 256.

The American Trucking Association's Technology & Maintenance Council (TMC) Recommended Practice number 1210 (RP1210) [13] provides a methodology for a personal computer to communicate with J1587 and J1939 compliant networks, as illustrated in Figure 1.

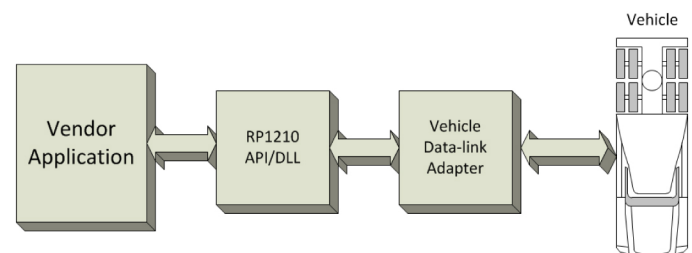


Figure 1. RP1210 information flow chart.

Currently, nearly all OEM-developed applications are capable of using an RP1210 compliant device to communicate with vehicles. As such, any new forensic software or device designed to communicate with these types of networks should support RP1210. Some compatible applications, such as DG Diagnostics, allow for recording of all messages passed between the Vehicle Data-link Adapter and the Vendor Application through the RP1210 API. This RP1210 recording or log file can serve as a rudimentary forensic record of the communications between an ECU and Vendor Application. As discussed earlier, performing a cryptographic hash of the log file after recording can be used to show that the log file has not been altered.

The J2728 standard for HVEDR data [7] specifies data sources that a HVEDR should record. Not only does it specify data points, it also specifies the preferred J1587 or J1939 messages that the EDR should sample to obtain those records. Referring to the J2728 standard for this information and cross-referencing the messages to the relevant vehicle networking standards will allow the investigator to determine which electronic device is responsible for computing the data and

sending the message. The standards can also be used to understand the sensor that the device used to obtain that data. This can enhance the investigator's understanding of the physical phenomena that the EDR data represent. Not all ECMs conform to the J2728 standard, so using standards based meaning should not be done without verification.

The J1939-73 standard for diagnostic data communications in heavy vehicles [14] contains specifications for diagnostic trouble codes used to indicate system or component faults, as well as direct memory access procedures that occur over J1939. The diagnostic specifications are useful for understanding the freeze frame data that may be attributed to a crash. Knowledge of the memory access procedure is also useful for evaluating how commercial tools access evidence data.

The networking protocols (J1708/J1587 and J1939) enumerate and define the various data that are transmitted over the in-vehicle network, which provides clues as to the nature of the data present in the ECM. For example, Road Speed data on a J1708 network only has resolution to the nearest 0.5 mph. The value reported by the ECM may have the same resolution.

Proprietary Meaning

Most new heavy vehicles contain some event data recording capability. These data are accessed from software supplied by the electronic control module manufacturer, as shown in [Table 1](#).

[Table 1. A list of manufacturers and the software used to access the data on the ECM.](#)

ECM Family	Software
Caterpillar	Caterpillar Electronic Technician (CatET)
Cummins	Cummins PowerSpec Cummins Insite
Detroit Diesel	DDEC Reports Detroit Diesel Diagnostic Link (DDDL)
Mack	Mack and Volvo Proprietary
Mercedes	DDEC Reports Detroit Diesel Diagnostic Link (DDDL)
Navistar	ServiceMaxx
Paccar	Cummins PowerSpec Cummins Insite or Paccar Davie
Volvo	Mack and Volvo Proprietary

While it may be tempting to trust the interpretation of the OEM software, research has shown that the interpretation of the data may be at issue. For example, in some Cummins Sudden Deceleration events, speed data is reported in the report at one sample per second; however, after comparing to external reference measurements, it is discovered that the data actually is reported at approximately 0.2 second intervals (5 Hz) [15]. Similarly, Austin and Farrell [16] showed that many Snapshot records from Caterpillar are reported on intervals that are approximately 0.5 seconds apart instead of every second as represented in the OEM software.

Establishing and verifying meaning for all data elements in the digital records is a formidable task. The current body of literature has made strides to evaluate many aspects of these EDR records, but more work remains to be done in this area.

When distilled into some basic elements, the meanings from heavy vehicle digital data representing time history traces (like speed and RPM) can be understood in two dimensions: time and value. Since the ECMs on heavy vehicles are designed to control some process that takes time, the timing of events is often recorded. The other aspect of data is its value. In other words, the question investigators want answered by digital ECM data is, "when did this value occur and what is the accuracy of this value?" As such, there are two dimensions (time and value) to evaluate when looking at establishing the meaning of the data in an ECM record.

Most of the literature regarding heavy vehicle EDR data is dedicated to these timing and value accuracy studies. Recently, Plant et al. [17] studied the timing and synchronization of data from DDEC V control modules. Other notable research on the meaning of EDR data includes Steiner et al. on the Mack V-MAC III [18], Reust on the DDEC IV [19], van Nooten et al. on the Caterpillar and Detroit Diesel Modules [20], Drew et al. on Caterpillar ECM data [21], and Reust et al on DDEC series modules [22]. Messerschmidt et al. [23] explored the effect of power loss for ECM data, which is useful in explaining so-called anomalies in the EDR data.

Daily Engine Usage from DDEC Reports

Most of the aforementioned papers explored speed and brake timing and history. As such, researchers are beginning to focus on some of the other historical data in HVEDRs (e.g. [24]). The premise of the research was to evaluate the timing of the transitions from different states of the vehicle according to the graphs shown in [Figure 17](#) in the [appendix](#). The table entry for this graph from DDEC Reports is shown in line with the text in [Figure 2](#) below and repeated in the [appendix](#) as [Figure 18](#) with a large format.

DDEC® Reports - Daily Engine Usage	
Print Date: 8/21/2013 11:08 AM	Date Range: 01/18/07 To 01/07/00 (EST)
University of Tulsa 800 S. Tucker Dr Tulsa, OK 74104 (918) 631-3056	Vehicle ID: TIB DDEC4 Driver ID: Engine S/N: 06R0495534

Date: 1/18/2007	Total(hh:mm)	09:13	06:00	08:47
Start Time: 00:00:00 EST	Hour(EST)	Drive(min)	Idle(min)	Off(min)
Odometer: 1006109.00 mi	00:00-02:00	0	120	0
Distance: 548.80 mi	02:00-04:00	0	120	0
Fuel: 55.25 gal	04:00-06:00	96	24	0
Fuel Economy: 5.76 mpg	06:00-08:00	104	16	0
Average Speed: 59.54 mph	08:00-10:00	110	10	0
	10:00-12:00	54	66	0
	12:00-14:00	120	0	0
	14:00-16:00	69	0	47
	16:00-18:00	0	0	120
	18:00-20:00	0	0	120
	20:00-22:00	0	0	120
	22:00-24:00	0	0	120

Figure 2. Example data from DDEC Reports regarding daily engine usage.

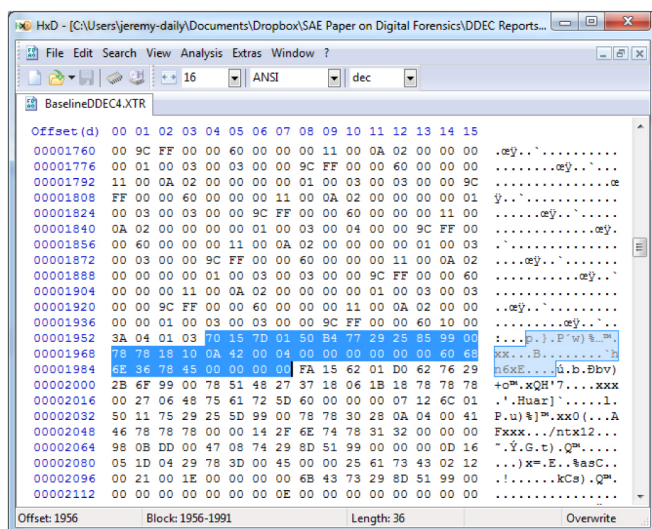


Figure 3. The first record of the Daily Engine Usage Log shown in a hex editor.

Examining the digital record in the XTR file may provide some insights into what numbers are recorded and whether or not the transitions from off to idle to drive correspond to stored data.

After opening the XTR file produced by DDEC Reports in a hex editor, as shown in Figure 3, one can find that the Daily Engine Usage Logs start on byte 1956 in the XTR file and each day takes up 36 bytes. Figure 3 shows the 36 bytes corresponding to the data in Figure 2 highlighted in blue. The interpretation of these hex values is shown in Table 2 in the appendix. The daily engine usage log has a pattern in the XTR file that is repeated 30 times to get the last 30 days.

In a DDEC IV and V (and maybe some others), Timestamps are stored as a 32 bit number (0-44294967295) representing the number of seconds from an epoch corresponding to Jan, 1 1985 at midnight (UTC). Since each machine interpreting these values may be different, the epoch must be calculated before interpreting the number. For example, the time passed from the Windows epoch may be different than the time since the J1587 epoch. This difference is accounted for by subtracting off the PC's epoch from the J1587 epoch. Implementing this with the Python time library [25] is as follows:

```
import time
tZero=time.mktime((1985,1,1,0,0,0,0,1,-
1)) - time.timezone
```

which yields a value of tZero = 473385600 seconds. The time.timezone value includes the PC's time zone into the tZero. For Central Standard Time, time.timezone = 21600 seconds or 6 hours. Interpreting the 32 bit number (Intel format) can be done with Python's struct library [26] as follows:

```
import struct
TSbytes = b'\x50\xB4\x77\x29'
TS = struct.unpack('<L',TSbytes)[0]
```

where b'\x50\xB4\x77\x29' is the byte string corresponding to the time stamp in the XTR file. Adding TS to tZero (the epoch from J1587) gives a value of 1,169,096,400 seconds from the PC's epoch of midnight of Jan 1, 1970 UTC. Converting these seconds into a usable timestamp uses the following Python command:

```
timeString = time.strftime("%A, %d %b
%Y at %H:%M:%S %Z",
time.localtime(TS+tZero))
```

which produces a result of 'Wednesday, 17 Jan 2007 at 23:00:00 Central Standard Time.' This matches the instance from DDEC Reports since the ECM was programmed to work in the Eastern Time Zone.

The odometer reading is stored as a 32 bit unsigned integer representing the number of 10ths of a mile passed. This is consistent with J1587. The next 12 entries are for the idle time and the following 12 bytes are for the drive times as broken down in the number of minutes per 2 hour block. This completes the pattern and these are the only recorded data. The key observation is that the XTR file does not contain time stamps regarding the transition from one state to another. This means the graphic shown in Figure 17 is generated by DDEC Reports on the PC with an algorithm only based on the drive and idle times. The off time, fuel economy and average speed are all calculated parameters in DDEC reports. Therefore determining the exact timing of when a driver transitioned from off to idle to drive requires more evidence than can be provided by DDEC reports. The data also does not account for the number of transitions between drive, idle, and off.

Examining the digital record at the byte level and matching the hex data to the DDEC Reports enables investigators to know what data were recorded and what data are calculated. A deeper examination of the RP1210 log files show that parts of the J1708 network traffic is contained within the XTR file. Therefore, the XTR file is a representation of the communications from the ECM.

The example shown herein is for a Detroit Diesel DDEC IV interpreted by DDEC Reports version 8.02. Other manufacturers interpret and save their data in plain text formats, like html or xml, which can be edited.

Error Detection and Mitigation

Boggess et al. [27] demonstrated that another issue with evidence preservation is the creation of spurious fault codes during bench downloads. Following this work, Plant, Austin, and Smith [28], showed how many data elements remain unchanged when an ECM is removed from a vehicle. Ideally, incident data are extracted through the diagnostic port in the cab of the vehicle. The ECM resides in the vehicle in question, and so the diagnostic trouble codes that it records reflect the actual condition of the vehicle. However, the in-vehicle network may be damaged during the crash, making the diagnostic port unusable for this purpose. This necessitates removal of the ECM from the vehicle and reading it directly, a process commonly known as a bench download. The simplest method of performing a bench download, connecting to the ECM using an engine programming harness and performing the download as usual with a diagnostic tool, can often yield spurious fault codes because the ECM fails to detect various temperature sensors, pressure sensors, and so forth.

Investigators have developed several methods for mitigating this problem. If available, a solution is to use a donor vehicle that is nearly identical to the vehicle involved in the incident. The potential difficulties of obtaining a surrogate vehicle, or difficulty in the case of non-fleet vehicles, are self-evident. Another method involves purchasing all of the components that the ECM expects and connecting them to the ECM. Doing this for each ECM could be prohibitively expensive.

Establishing Transparency and Trust

Regardless of the meaning of the digital data, it is necessary to present data in its final form in such a way that is transparent of its handling to establish trustworthiness. According to the transparency principle of forensic soundness, actions taken by an investigator should be available for later examination. Additionally, any error conditions encountered by the software should be recorded so that the legal weight of the evidence may be accurately considered. Journals are log files generated by forensic software to meet these requirements, and should at least be an option in any forensic solution.

One example of a possible attack on the integrity of current data formats is deleting audit trails in Cummins Insite's EIF files. EIF files are simply XML-formatted plain text files compressed into encrypted ZIP files. While the cryptographic security of ZIP file formats has historically been suspect [29], in this case the encryption strength was irrelevant, as the encryption passphrase was pulled from memory when Insite opened a file. With the passphrase, one can simply decompress the text file, alter the data using a program like notepad, and re-encrypt it.

Baseline of Trust

In order to make a full accounting of potential sources of error in any forensic process, it is important to examine the assumptions that the process requires. Standard methods of

truck evidence extraction assume that the ECM is reporting data faithfully, that the vehicle network is accurately transmitting the communication to and from the ECM. It also assumes the Vehicle Diagnostic Adapter (VDA) is faithfully sending data from the vehicle network to the RP1210 library that is called by the OEM software.

While verifying the correct internal functioning of heavy truck ECMs is beyond the scope of this paper, the fidelity of data transmitted across the various networks (CAN bus, serial bus and USB) can be easily verified. Both protocols make use of error-correcting codes that detect transmission errors with very high reliability; for example, the CAN Cyclic Redundancy Check that is defined in the CAN standard process and detects all errors from 1 to 5 bits in the data frame, and only fails to detect 0.00018% of 6-bit errors in a CAN frame [30]. This represents a level of certainty that would take many years to experimentally verify because the likelihood of having an undetected error is so low.

Assuming the ECM is performing correctly, the programmed parameters (like tire size) are known, and data flowing through the VDA maintains its integrity, then the next level of trust would be the OEM software and the native file formats they produce. These files, however cannot be trusted because they can be altered without detection. Therefore, the baseline of trust for digital forensic purposes is the vehicle network traffic, which can be accessed with a network data logger or with a verbose log file generated by the RP1210 library for the VDA. Details of why native files cannot be trusted are shown with an example from DDEC Reports in the following sections.

Data Manipulation

Currently, truck evidence data are stored in a variety of file formats. These include standard data formats such as HTML or PDF, and proprietary formats used by the various engine maintenance software products. However, these formats are vulnerable to alteration. Through examination of the files and the programs that generate them, it was determined that:

- Cummins PowerSpec Reports are stored natively as HTML files that can be manipulated with a text editor.
- Cummins EIF files are password-protected ZIP archives that contain data formatted using XML, a text-based format for organizing data similar to HTML. The same password is shared between all EIF files so knowing the password allows someone to modify any EIF file by decompressing, modifying, and re-compressing the data. While the cryptosystem used by ZIP files is known to have some flaws [26], one of the authors simply recovered the password by reading it from memory with a debugging program while Insite opened an EIF file.
- DDEC Reports XTR files are a raw record of the data extracted from the ECM via J1708 or J1939. Editing the binary data can be done with a hex editor. Location of the bytes containing the speed data in the Hard Brake Records and Daily Engine Usage Log is presented below.