Forensic Science International: Digital Investigation 36 (2021) 301111

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

Full Paper

A generalized approach to automotive forensics

Kevin Klaus Gomez Buquerin [a, c, *], Christopher Corbett [b], Hans-Joachim Hof [a, c]

[a] Technical University Ingolstadt, Germany
[b] University of Ulm, Germany
[c] CARISSMA Institute of Electric, Connected, and Secure Mobility (C-ECOS), Germany

ARTICLE INFO

Article history:
Available online 23 March 2021

Keywords:
Automotive
Digital forensics
Diagnostics
Forensic
Cyber security
Embedded
Vehicle

ABSTRACT

In the past years, software became an essential topic in modern vehicles, e.g., with the rise of more and more complex driver assistance systems. The advent of automated driving will drive this trend even further. Today, accident investigation, as well as warranty claim analysis, need to take into consideration an analysis of the rapidly increasing proportion of software and security based implementations as part of modern vehicles, the so-called digital forensics. This paper evaluates the general feasibility of digital forensics on a state-of-the-art vehicle. To do so, we analyzed current digital forensics techniques on a state-of-the-art vehicle to constitute gaps in the automotive forensics process used on in-vehicle systems. We present a general process for automotive forensics to close existing gaps and implemented it on a state-of-the-art vehicle in an in-vehicle device manipulation scenario. The implementation uses the on-board diagnostics interface, the diagnostics over internet protocol, as well as the unified diagnostic services for communication. Our implementation requires automotive Ethernet at the diagnostic interface.

Our research shows future directions for efficient automotive forensic as well as the exemplary feasibility of automotive forensic analysis on state-of-the-art vehicles without the need for additional in-vehicle components such as intrusion detection systems or event data recorders.

1. Introduction

The car industry is currently undergoing a tremendous change, driven by harsh regulations, emerging new business and service models, and new competitors. Technologically, many emerging business and service models depend on the software and/or connectivity between vehicles, infrastructure, and backend services. As shown by Litman (2017), it leads to enhanced complexity of state-of-the-art automotive systems. It further conducts to an increased attack surface as presented by Manadhata and Wing (2011). The past has already seen some serious hacks of vehicles and it is very likely that the future will see even more grave digital attacks on vehicles. Hence, it is only a matter of time that digital forensic investigation of vehicles will be necessary to resolve security incidents.

Digital forensics can be considered a mature science in the field of IT. As state-of-the-art vehicles incorporate more and more computer- and software-based components, it may be possible to transfer some or all digital forensic methodologies from digital forensics in IT to the field of digital forensics of automotive systems. To research this issue, we state the following two questions: "*Is it possible to perform forensic analysis on state-of-the-art vehicles?*", and "*Which gaps can be identified in state-of-the-art vehicles?*".

Our **contributions** are listed below:

- Presentation of stakeholders and scenarios for automotive forensics.
- Classes of data available for automotive forensics investigations as well as their significance for forensic investigations.
- While existing literature mostly addresses automotive forensics investigations with the assumption of additional devices and technologies, we present a general process to perform digital forensics investigations on state-of-the-art automotive systems without such extensions in cars and vehicle connected systems.

This paper is structured as follows: In Chapter 2, we present related work in the field of automotive forensic investigations. Next, we analyze automotive forensics in depth in Chapter 3. This chapter incorporates the definition of automotive forensics, a

* Corresponding author. Technical University Ingolstadt, Germany.
E-mail addresses: extern.kevinklaus.gomezbuquerin@thi.de (K.K. Gomez Buquerin), christopher.corbett@uni-ulm.de (C. Corbett), hof@thi.de (H.-J. Hof).

https://doi.org/10.1016/j.fsidi.2021.301111

presentation of the stakeholder and scenarios, categories and acquisition methods, as well as data types in state-of-the-art vehicles. Chapter 4 presents a general concept for automotive forensics. Details on an exemplary implementation of this concept on a state-of-the-art vehicle are given in Chapter 5. Chapter 6 elaborates the evaluation of the forensic concept based on exemplary implementation and identifies gaps in state-of-the-art vehicles. Chapter 7 concludes the paper and gives an outlook on future work.

2. Related work

In recent publications, researchers presented methods to perform forensic analysis and investigations in the domain of vehicles. Nilsson and Larson (2008) presented requirements for in-vehicle networks to perform forensic analysis. The researchers focused on the Controller Area Network (CAN) bus, a widespread bus technology used in vehicles. The authors presented a qualified attacker model and listed requirements for different areas such as detection, collection of data, and the reconstruction of events.

Kiltz et al. (2009) evaluated several data sources of automotive systems. Their primary focus was conducted on the Background Debug Interface (BDM). By using BDM, they were able to perform embedded forensics investigations on automotive systems.

Hoppe et al. (2012) performed a forensic investigation using a CAN bus data logger. Based on the "*German Bundesamt für Sicherheit in der Informationstechnik (BSI)*" guidelines for IT-forensics, the researchers were able to reconstruct a travel route and resolved a hit-and-run suspect scenario.

Feng et al. (2017) presented a model to collect data from automated vehicles in smart cities. They collected data from the vehicles Electronic Control Module (ECM), smart city data (which includes data hubs, Global Positioning System (GPS), and cellular information), and data collected by the investigator. Afterwards, the information was hashed and encrypted to ensure integrity and authenticity of the collect information. Both are stored in a secure storage system. In an example, they collected diagnostic trouble codes over On Board Diagnostics (OBD). In addition, they collected "digital forensics data" from a Vauxhall Corsa, but do not present details about the collected content itself. Besides diagnostic trouble codes over OBD, the research did not present methods and technologies to acquire data from modern vehicles. The presented framework assumes the presence of vehicle ECM data, smart city data, and data collected from the investigators.

Hossain et al. (2017) presented a full framework to allow forensic investigations in the area of smart and connected vehicles. The authors assumed the presence of multiple components, such as a forensic gateway, installed in vehicles, mobile phones, or other vehicle-to-x components. Information generated on a low level (i.e. Electronic Control Unit (ECU)) was not in focus of this paper. Rather, the different components collected interaction-logs between vehicle-to-x participants in order to allow forensic investigations.

Walter and Walter (2018) performed digital forensic investigations for light-duty vehicles. They acquired data using the OBD connection interfaces and CAN as the underling bus technology. As a result, the authors were able to collect data such as vehicle load, throttle position, and barometric pressure.

Technologies in vehicles drive forward with every model. The research mentioned above all rely on the CAN bus protocol and/or require additional hardware or technologies installed within a vehicle. With several other legacy bus systems inside an in-vehicle network and introduction of automotive Ethernet as shown by Corbett et al. (2016), an analysis of forensic capabilities with a larger scope and no additional hardware in the vehicle is necessary. Additionally, a general and adaptable automotive forensics process has not presented yet. This paper bridges these gaps.

3. Automotive forensics

Automotive forensics is the utilization of digital forensics techniques and methods on automotive-related systems. These systems include in-vehicle components such as ECUs, manufacturer IT systems (backends), consumer electronics (e.g., smartphones), as well as vehicle to infrastructure/vehicle communication systems. Automotive forensics aims at finding an answer to a question stated by a stakeholder about a specific scenario, e.g., an accident. The output of automotive forensics should give information about the 6 W's: *Who*, *why*, *where*, *when*, *what*, and *how*.

Automotive forensics includes the fast acquisition of data, scoping (initial triage), and the time consuming in-depth analysis of in-vehicle components such as embedded forensics, the analysis of embedded devices.

3.1. Stakeholder and scenarios

It should be noted that the main focus of this paper is on security-related scenarios and not solely on accident reconstruction. In this respect, a scenario is a description of a circumstance under which a stakeholder requires specific information, data, and facts about the main cause and the impact of a security incident involving an automotive component, a human being, or any related computer system. A stakeholder in this respect is any party that is obligated to perform monitoring, mitigation, and resolving of any security or safety-related events towards a vehicle and any human being directly or indirectly affected by the event. Hence, it is necessary to derive the requirements for automotive forensics from the needs of all possible stakeholders and scenarios. For this research, brainstorming according to Bryson (2004) was used to come up with relevant stakeholders by a group of professors, PhD students, as well as Original Equipment Manufacturer (OEM) employees. In contrast to other stakeholder identification methods, such as the snow-ball technique,[1] brainstorming is applicable to automotive forensics stakeholders.

The following stakeholders and associated scenarios were identified:

Insurer. Any company or party involved in selling insurance policies covering accidents, service guarantee, and their impact on other parties. Potential scenario examples are:

- Determine any modification of hard- and software components (e.g., tuning) in the vehicle.
- Determine any third party modification or manipulation that could have caused the accident.

Legal Entity. Any party representing a legal authority affected by a security event such as police, legislator, or courts. Potential scenario examples are:

- Determine if the manufacturer and furthermore the released vehicle complies to state-of-the-art laws.
- Determine if the manufacturer covered the necessary technical requirements and state-of-the-art technologies and algorithms to prevent, mitigate, or monitor any security violation.

Manufacturer. An entity that is responsible for the manufacturing and distribution of a vehicle as a product—referred to as OEM. Potential scenario examples are:

[1] This technique would not be applicable for stakeholders such as criminals or government agencies.

- Analyze and reproduce potential security violations and/or misuse of vehicle components.
- Determine if specific types of data are present within their products and fulfilment of regulations are given (e.g., personal information).

Supplier. Any party that is involved in the development, production, and delivery of components or services linked to a vehicle manufactured by OEM. Potential scenario examples are:

- Analyze and prove malfunction, modification, or misuse of equipped components.
- Determine if their intellectual property is protected from forensic analysis.

Customer/Car Owner. Any single entity or group of entities that is the user of a vehicle or service related to the product. Potential scenario examples are:

- Determine liability in the case of malfunction of a vehicle or related service.
- Evidence odd behaviour of the vehicle or a related service.

3.2. Forensic types and acquisition methods

To confirm or reject a given scenario, forensic data acquisition and analysis is needed. We identified two types of forensic analysis that are appropriate for automotive forensics.

Live Forensics. Data is acquired from an running system. This system could be either fully functional (e.g., an up and running vehicle), or it could consist of one or more components interacting in a hardware-in-the-loop simulation as shown by Isermann et al. (1999). One advantage of live forensics is the ability to extract volatile data from the system. However, the danger of live forensics is data loss or corruption of evidence.

Post-Mortem Forensics. In post-mortem forensics, the overall system is shut down to acquire only persisted data. An advantage of this method is the reduced risk of forensic evidence corruption. However, post-mortem forensics does not allow to acquire volatile memory data.

Prerequisite for any forensic analysis is data collection. Hence, we identified the following data acquisition methods relevant to automotive forensics:

Online Acquisition. Forensic data acquisition, where software based techniques are utilized. Examples are log file analysis and acquisition of volatile RAM memory. These methods allow fast acquisition of available data. The amount of collectable information depends on the implementation for each in-vehicle component.

Offline Acquisition. Forensic data acquisition of switched off individual vehicle sub-components. This includes desoldering of logic board components and utilization of established embedded forensic techniques. Due to preparation and disassembly overhead this method is more time consuming.

3.3. Classes of data available for automotive forensics

Data acquisition and advanced data analysis in automotive forensics operates on several classes of data. Our analysis of several in-vehicle components identified five classes of data that are typically suitable for forensic analysis: **firmware**, **communication data**, **user data**, **safety-related data**, and **security-related data** (see Fig. 1 for exemplary entities of these classes). In the following, we elaborate on the importance of these classes of data as well as more details on how data from each class is used in automotive forensic analysis.

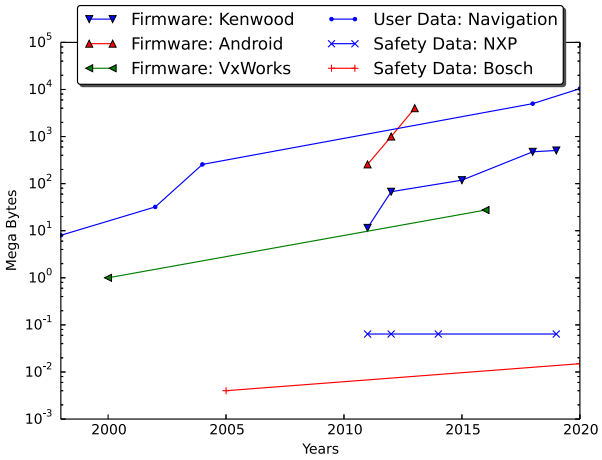

Fig. 1. Growth of size for several examples of data found to be useful for automotive forensics.

Firmware. We define firmware as a monolith of software installed on an ECU. Firmware incorporates the operating system (e.g., Realtime Operating System (RTOS), POSIX (Linux, Android, QNX, VxWorks)), frameworks (AUTOSAR, Adaptive AUTOSAR), device drivers, applications, and other data (e.g., application data, images). An analysis of automotive firmware and associated meta data of Kenwood, publicly available on their website[2]), and of the Android operating system, which is commonly used in infotainment system components, both showed a noticeable increase in firmware size. The size of the Kenwood firmware increased from 11.5 MB in 2001 to 506.3 MB in 2019 (see Fig. 1). The required flash memory for the Asteroid Class from 2011 with 256 MB increased up to 4 GB with the Clarion AX1 in 2013. Another example is the operating system VxWorks provided by Wind River. VxWorks 6 required a minimum of 1 MB Random Access Memory (RAM), whereas the latest version VxWorks 7 utilizes 27,5 MB as shown by Gomez Buquerin (2018). To a certain extend, the growth of firmware images can be linked to a constantly raising number of network protocols, application complexity, and security features.

Significance of firmware data for forensic analysis: An analysis of firmware data is suited to detect ECU modification (e.g., manipulation, feature addition/reduction). Hence, this class of data is important for automotive forensics. Examples are binary (.bin) files publicly available on the manufacturer websites. Embedded forensics techniques can be used to collect firmware from ECUs. By desoldering the flash Read Only Memory (ROM) and dumping the content using the Debian *flashrom* utility.

Communication data. We consider communication data as all data that gets transmitted both inside the vehicle and from the vehicle to any other receiver. Transmission of data may use internal bus technologies and protocols, as well as common Information Technology (IT) protocols. We further distinguish between the following communication scenarios:

- ECU to ECU communication (in-vehicle communication),
- vehicle to IT infrastructure communication (e.g., Backend Communication) (vehicle-to-backend communication), and
- vehicle to traffic-infrastructure communication (e.g., car2car, Power Line Communication (PLC), smart cities) (car-to-x communication)

[2] https://www2.jvckenwood.com/cs/ce/firmware last accessed 24. November 2020.

Vehicle-to-Backend communication can be covered with existing forensic capabilities of the IT domain, hence solely in-vehicle communication and car-to-x communication are in the scope of vehicle forensic analysis.

Similar to the increase in firmware image size of ECUs, the amount of communication data has also increased in the last couple of years. This increase is rooted in additional sensors for driver assistance systems, an increase of on-board entertainment services, and data distribution use cases. Furthermore, the additional use of new protocols (e.g., Scalable service-Oriented MiddlewarE over IP (SOME/IP) and Diagnostic Over Internet Protocol (DOIP)) and the use of multiple proprietary protocol implementations based on Hyper Text Transfer Protocol (HTTP) leads to increased complexity of vehicle communication.

Significance communication data for forensic analysis: Today, the functionality of state-of-the-art vehicles is distributed amongst several ECUs. This implies that network traffic between distributed components is essential to analyze distributed functionalities. Examples are packet captures of communication between the vehicle and official repair shop devices. Here, diagnostic data such as software version number are identifiable. Hardware tools such as OBD dongles and software tools such as Wireshark allow to captures communication data.

User Data. We define user data as any information that is created, modified, or removed through the interaction of any party (e.g., owner, driver, passenger, tenants) interacting with the vehicle, including data transmitted from connected consumer electronics like smartphones or Universal Serial Bus (USB) peripherals to vehicle components (primarily, the infotainment system). Our analysis revealed a significant increase of data sizes over the last two decades in infotainment systems. In 1998, maps of the navigation system manufacturer Garmin required between 8 MB and 16 MB in size, in 2002 between 32 MB and 128 MB were needed to store these maps, and in 2004 as well as 2005 already 256 MB of storage were necessary. The current navigation systems of Tesla and Audi show an increase from 5 GB required storage in 2018 to 10 GB in 2020.

Significance of user data for forensic analysis: Information about interactions with the vehicle and potential persisted malicious/modified files support attack scenario definitions and support the identification of a root cause. Hence, user data is of great importance for automotive forensics. It should be noted that current legislation, in particular the General Data Protection Regulation (GDPR), require specific handling of user data that is out of scope of this paper. Examples of such are call logs or the phone book shared between the vehicle infotainment system and the drivers smartphone. The iVe vehicle forensics tool from MSAB[3] allows to collect such information from infotainment systems.

Safety-related Data. We define safety-related data as data about the safety state of the vehicle and its components. Devices that store information about safety-critical events (e.g., Event Data Recorder (EDR)) are mandatory for vehicles registered in countries like the United States of America, Korea, Japan, Switzerland, and Uruguay. daSilva (2007) showed that these systems acquire and store different types of information, such as accident type, travel speed, seat belt status, airbag deployment, and vehicle motion—to name a few. Usually, the required storage space (e.g., 5 s before and after an accident) is application-specific and cannot be estimated in general. For example, an airbag ECU stores event information within an external Electrically Erasable Programmable Read-Only Memory (EEPROM) and does not require a significant amount of memory. The airbag ECU family MPC56XX released by the semiconductor manufacturer NXP has 64 KB of EEPROM space. This particular ECU remained unchanged in size between 2011 and 2019. Unlike the unchanged storage requirement for EEPROM memory, requirements for ROM - 200 kB in 2005 to 800 kB in 2020 - and RAM - 10 kb in 2005 and 80 kB in 2020 - increased due to requests by the manufacturer Bosch. Böhm et al. (2017) stated that an increased demand for memory in upcoming EDR is inevitable due to new features such as video data recording that requires an estimate of 150 MB of storage for 1 min of uncompressed 30 Frames Per Second (FPS) High Definition (HD) video footage.

Significance of safety-related data for forensic analysis: Safety events can be used as an indicator for manipulation. Hence, it is of great importance for automotive forensics. Examples of such information is data about the break use, the vehicle speed, or which seat belts are used. To collect EDR data, Bosch offers a custom CDR tool.[4]

Security-related Data. We define security-related data as information that is either directly linked to security events (e.g., immobilizers, cryptography material modification, revocations, extended access) or provide implicit information towards security. Unlike the previously stated EDR, ECUs do not have dedicated centralized storage for security events, and therefore no reliable statement about potential memory consumption or event types can be made.

Significance of security-related data for forensic analysis: Security events provide a reliable source of information regarding potential misuse or manipulation of the vehicle or its environment. Examples are information like Diagnostic Trouble Code (DTC) entries, firewall counters,[5] or security monitoring (e.g., Intrusion Detection System (IDS), Intrusion Detection and Prevention System (IDPS)) events. Herold et al. (2016) presented an IDS for SOME/IP. Implementation of such technologies in vehicles would allow collection of security-related data.

4. Automotive forensics process

The previous sections have shown the complexity of modern vehicles, stemming from distributed functionality as well as a mixture of operation system and communication technologies. To address automotive forensics in an efficient way, a generalized automotive forensics process is required that adapts to a wide variety of vehicles and environments. Fig. 2 gives an overview of our automotive forensics process.

Our automotive forensic process is organized in four phases: **forensic readiness (A)**, **data acquisition (B)**, **data analysis (C)**, and **documentation (D)**. Each phase is described in detail in the following.

4.1. Forensic readiness

Forensic readiness evaluates if cost-efficient forensic investigations are feasible. This phase checks if potential log sources, tools, and technologies to exchange information are available. To ensure reproducibility of the forensic analysis, all steps are documented in this phase. In the following, the steps A:1 till A:3 of this phase are described.

A:1. In this step, potential data sources are researched and identified. This step incorporates the use of knowledge from former research, literature, internal documentation, and more.

A:2. In this step, interfaces and communication methods to use

[3] https://www.msab.com/products/ive-vehicle-forensics/last accessed 24. November 2020.

[4] https://boschcdrtool.com/last accessed 24. November 2020.

[5] E.g. number of blocked connections.

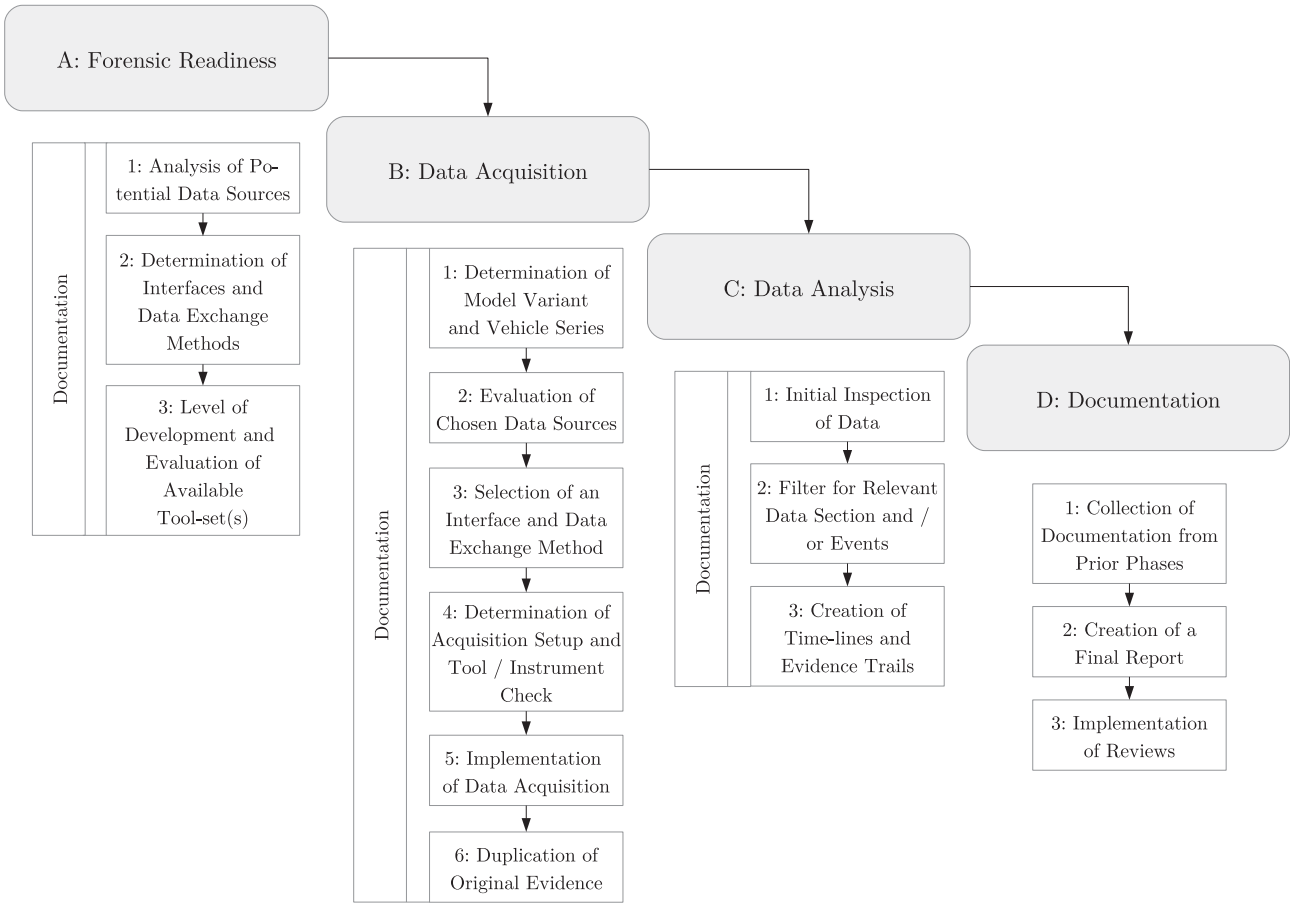

Fig. 2. Automotive forensics process model by Gomez Buquerin (2019).

for automotive forensic are determined. Suitability of interfaces and communication methods depends on the target of the automotive forensics as well as on the intended data classes. Examples include broadcom methods (e.g., Long-Term Evolution (LTE), 3G, Global System for Mobile Communications (GSM)), debug interfaces (e.g., JTAG, BDM), wireless methods (e.g., WiFi, Bluetooth), and diagnostic services (e.g., OBD).

A:3. In this step, the level of development for automotive forensics gets identified and an evaluation of available tools[6] is performed. Court-relevant forensic analysis requires acceptance by the forensic community as shown by Geschonneck (2014). A high level of development of automotive forensics is achieved by using documentation of former analyzes, adapting best-practises, and using accepted tool-sets.

4.2. Data acquisition

In this phase, relevant data is collected by using the interfaces and communication methods identified in phase A. Documentation is performed during all steps of the phase. Positive and negative results are noted.

B:1. In this step, model variant and vehicle series of the target vehicle are determined to identify all components and data sources that are available for forensic analysis.

B:2. In this step, data sources are assessed. The focus of the forensic analysis depends of the stated questions that should be researched. If the current or former position of a car should be listed, the forensic analysis focuses on GPS data. If a leak of personal data from a vehicle should be investigated, the infotainment system and related components are of interest.

B:3. In this step, appropriate interfaces and data exchange methods are selected based on the questions that should be answered. This step differs for each type of acquisition and can reach from embedded forensics techniques, as shown by Park et al. (2012), to using wireless technologies for collection.

B:4. In this step, the data acquisition setup gets installed. It includes a final review of tools, targeted devices, and interface methods. The data acquisition setup gets tested on an automotive system similar to the target system to double-check functionality and correctness.

B:5. In this step, relevant data gets collected from in-vehicle components using the selected data acquisition setup.

B:6. In this step, the assessed data of the target vehicle gets duplicated and original data gets stored in a tamper-proof way. To do so, original data must be duplicated and accumulated in a cryptographic secure way. Later phases will only work on the duplicated data produced by this step.

4.3. Data analysis

In the data analysis phase, the duplicated data is analyzed to answer the stated questions. All analysis steps get thoroughly analyzed to ensure the reproducibility of the analysis results.

C:1. In this step, the analyst gains an understanding of available

[6] The tools are collectable by internet research. Examples are code repositories such as Github.

information in the data. Commonly, it is unclear how much information is contained in the initially collected data and the amount of data to analyze is high.

C:2. In this step, the analyst selects relevant data sections and events by using filters. For example, if ECU software manipulation is in the focus of the analysis, the Unified Diagnostic Services (UDS) data identifier *0xf199* (*ProgrammingDateDataIdentifier*) is of interest.

C:3. In this step, timelines and evidence trails get created. This step is of uttermost importance as logical and reproducible evidence trails must be presented in a potential court hearing. We show an example timeline for analysis with a focus on software manipulation in the following:

1. Last active repair shop code or equipment serial number changed (UDS data identifier *0xf19a* and *0xf198*)
2. Programming data for the firmware on an ECU changed (UDS data identifier *0xf199*)
3. Firmware fingerprint is different to the one captured by the initial flashing of an ECU[7] (UDS data identifier *0xf184*)

4.4. Documentation

Phase D is the last phase of the automotive forensics process. The main focus of this phase is to produce a report for the intended audience and in human-readable form.

D:1. In this step, the analyst collects portions of the final report from the output of tools used in previous phases.

D:2. The goal of this step is a well-structured and comprehensive final report that describes all stated questions, successful implementations, and failures that appeared during the analysis.

D:3. The final step is a thorough review of the final report with a focus on uncertainties and errors. At last, the analyst signs the final report to ensure the integrity of the document for the future.

5. An exemplary implementation of the automotive forensic process on a state-of-the-art vehicle

We implemented the presented automotive forensics process on a state-of-the-art electric vehicle (build in 2018 by an European OEM). The car features lane assistance, a navigation system, an infotainment system, rear and front view cameras, and more.

We selected software or hardware manipulation as scenario. Indicators for manipulations can occur, if the vehicle was tuned. Furthermore, attacks on the ECU firmware can lead to manipulation too. As a result, the state of the vehicular system is not tester properly and the vehicle registration can expire.

As an attacker model we see OBD dongles available on the internet. Those components allow to collect diagnostic information and send requests to different ECUs connect to the diagnostic interface. Manipulation through OBD or other interfaces (e.g., if JTAG is still enabled on an ECU) can lead to issues in the functionality and safety of the vehicle.

We implemented live forensic analysis as well as online data acquisition to avoid physically damaging the vehicle. It is important to note that we did not physically extract any data, i.e. by applying embedded forensics techniques. We were not able to determine user interactions with the systems on a low level information base (e.g., by noticing hardware extraction indicators). Furthermore, no UDS data identifier is present that allows to collect such information.

We first analyzed potential data sources, interfaces, and data exchange methods and determined the level of development as well as evaluated if tool-sets are available. Due to the lack of internal documentation, we used publicly available literature and open-source tools for the first phase of the process. The target vehicle incorporated all data sources presented in Fig. 1. We concluded that WiFi, LTE, 3G, OBD, and Bluetooth interfaces are present.

However, we were not able to determine, if JTAG or other hardware debug interfaces were available because we did not perform hardware forensics techniques. A lot of open-source hardware and software is available for bus systems such as CAN. Regarding the reasoning above, we concluded that forensic readiness is fulfilled.

Next, we executed phase B. In step B:1, we collected the Vehicle Identification Number (VIN). Since this number follows an international standard, we were able to determine information such as general characteristics, vehicle type, and manufacturer of the target vehicle. We further performed a visual inspection of the vehicle (checking the logo of the manufacturer, model designations, and the unique design) and verified the findings from the VIN analysis.

In step B:2, we evaluated the data sources identified in step A:1. We must read ECU data to identify manipulated software or hardware components. This analysis included an analysis of internal and external flash storage as well as EEPROM store programming timestamps, hardware and software identifier, hardware and software fingerprints, and more. This analysis allows determining changes by comparing the results to a known default configuration. We selected the OBD interface to acquire data from ECUs in step B:3. The standardized diagnostic interface allowed us to request information from different ECUs connected to the port.

Within step B:4, we presented the final acquisition setup and performed a tool and instrument check. As shown in Fig. 3, it consists of an analysis device, a connector, and the vehicle.

The analysis device was an Apple MacBook Pro running macOS 10.14. On this system, we installed a Python 2.7 framework implementing the Diagnostic over IP (DoIP) and UDS standards to communicate with the vehicle ECUs using both protocols. Furthermore, the framework fulfils requirements such as message structures and timings, similar to an original workshop tester. Since we had no internal documentation, we build a scanning application that allowed us to determine the logical addresses of installed devices using the target address (TA) field from the UDS protocol. The connector was an OBD-II to Ethernet cable. As presented in Fig. 4, the OBD-II standard introduces pins for multiple protocols such as CAN. Due to this, we were able to build an OBD-II to

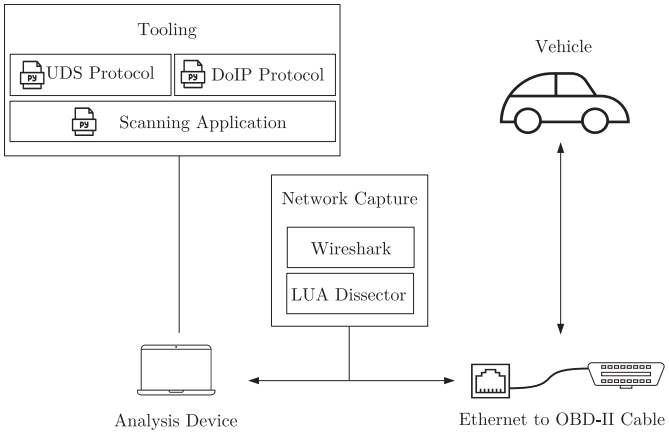

Fig. 3. Data acquisition setup (Gomez Buquerin (2019)).

[7] Assumption is that the original fingerprint is stored.

Ethernet cable by adapting the wiring scheme presented by Matheus and Königseder (2017). To capture traffic between the analysis device and the vehicle, we used Wireshark version 2.6 and a custom LUA dissector to make the output more readable. This setup was tested on a similar vehicle (same model and vehicle series) beforehand. No errors occurred during the tests, so we documented that the tool and instrument check was successful.

In step B:5, we used the scanning application to collect the logical addresses of installed ECUs. The scanning application queries information from all possible TAs (*0x0000* to *0xffff*). If a query was successful, we stored the TA in a JSON file. In total, we were able to identify more than 100 logical addresses. Next, we requested all available UDS data identifiers for each collected TA and monitored the network traffic using Wireshark. After the collection has finished, we stored the network capture as a Packet Capture (PCAP) file.

The last step of phase B was the duplication of the original evidence (step B:6). To ensure the integrity of the collected network capture, we created a duplicate using the command−line interface (CLI) command *cp*. We stored the duplicate on an external hard drive. Besides, we generated a SHA256 hash of the PCAP, stored it in a separate text file (also on the external hard drive), and documented it in the corresponding report for this phase.

We analyzed the collected data in phase C. As in previous phases, we documented all steps. For data analysis, we used the duplicate of the PCAP to ensure the integrity of the originally collected data.

Within the PCAP, we identified more than 3800 packets. Depending on the implementation of the ECU, the PCAP included positive and negative responses, manufacturer-specific identifier, and connection establishment data (e.g., TCP handshake).

We filtered for relevant data sections and events in step C:2. As seen in C:1, we identified events that contain no relevant data for the stated scenario (e.g., software/hardware identifier, software/hardware fingerprints, programming dates, last active tester, or equipment number). Due to this, we applied the Wireshark filter "*uds.service identifier == 0x62*" that only displayed positive responses from the ECUs. This filter decreased the number of displayed packets to 245. Next, we explicitly checked for interesting UDS identifier as presented in Table 1.

In step C:3, we aimed at creating timelines and evidence trails. We did not identify changes in the programming data for specific ECUs. Based on the available data, we concluded that there is no manipulation of the vehicle. As a result, there is no need to construct timelines or evidence trails.

In the final phase of the automotive forensics process (D), we collected all portions of the final report from the output of tools used in previous phases (step D:1). We compiled them into a final report (step D:2). We used separate sections to describe each phase. The final report incorporates all positive and negative test results. Hence, the reproducibility of our forensic investigation is guaranteed. In the last step (D:3), we reviewed the final report and signed it.

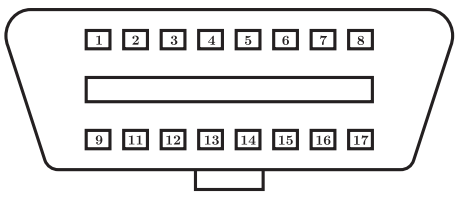

Fig. 4. OBD-II female connector Gomez Buquerin (2019)

1 : Manufacturer Discretion
2 : SAE J1850
3 : Manufacturer Discretion
4 : Chassis Ground
5 : Signal Ground
6 : CAN High
7 : K-Line ISO 9141-2
8 : Manufacturer Discretion
9 : Manufacturer Discretion
10 : SAE J1850
11 : Manufacturer Discretion
12 : Manufacturer Discretion
13 : Manufacturer Discretion
14 : CAN Low
15 : L-Line ISO 9141-2
16 : Battery Power

6. Evaluation and gap analysis

This chapter evaluates the suitability of the automotive forensics concept and shows a gap analysis of the investigation.

6.1. Usability evaluation

Phase A. Steps A:1 to A:3 use publicly available resources. The findings are reproducible by any third party. However, documentation of vehicle components would allow to identify manufacturer-specific data sources, interfaces, data exchange methods, and tools. The same issue exists for phase B and C.

Phase B. We used the VIN for verification of the model variant and vehicle series (B:1). This identifier follows a standard for Europe and Asia. The corresponding federal offices for motor transportation provide matrices to interpret different components of the VIN. As a result, the step is reproducible by any third party.

We chose EEPROM and flash memory as data sources (B:2). Park et al. (2012) had shown that flash memory is suitable for forensic investigations. For EEPROM, Casadei et al. (2006) determined that this technology is suitable for forensic investigations.

The presented tooling is written in Python and based on the DoIP and UDS standards. We selected Python due to the scripting capabilities and fast implementation on different Operating Systems (OSs). However, the tooling is academic code. No dedicated development or testing process was applied.

OBD was used during data acquisition in step B:6. The diagnostic interface is standardised and different adapters (e.g., OBD to Bluetooth, OBD to Wireless LAN (WLAN), OBD to Ethernet, etc.) are available. Hence, any third-party can implement this phase.

In step B:6, the SHA256 hash algorithm was utilized to ensure the integrity of the originally collected data. Preneel (1994) has proven that this algorithm is cryptographic secure.

Phase C. We performed an initial inspection of available data in step C:1 to avoid the complexity problem shown by Raghavan (2013). We were able to get an overview of the collected data and could reduce the number of relevant packets from initially 3800 down to 245.

We implemented filters for specific UDS data identifiers. Due to the lack of internal documentation, manufacturer-specific UDS identifiers were not interpreted. Internal documentation or reverse engineering of in-vehicle components is necessary to address this issue.

6.2. Gap analysis

We were able to identify multiple gaps in the used tool-set and state-of-the-art vehicles.

The Python tool-set offers limited capabilities. Two automotive protocols (UDS and DoIP) are covered. Due to this, only a limited amount of the many in-vehicle technologies can be used in forensic investigations. A solution is an extension with other protocols such as SOME/IP. As a result, our framework is adaptable for more vehicle types from different manufacturers. On the other hand, higher testing, as well as maintenance, is required by keeping the tools up-to-date.

In our research, we had limited knowledge and resources regarding different components. One example is implementations

Table 1
UDS data identifier to identify software or hardware manipulation.

UDS Data Identifier in Hexadecimal	Description
0xf180	bootSoftwareIdentificationDataIdentifier
0xf181	applicationSoftwareIdentificationDataIdentifier
0xf183	bootSoftwareFingerprintDataIdentifier
0xf184	applicationSoftwareFingerprintDataIdentifier
0xf198	repairShopCodeOrTesterSerialNumberDataIdentifier
0xf199	programmingDateDataIdentifier
0xf19a	calibrationRepairShopCodeOrCalibrationEquipmentSerialNumberDataIdentifier

unique to the producers such as UDS manufacturer-specific identifiers. Internal documentation, standardisation, and reverse engineering of in-vehicle components would allow analysts to utilize this information. A positive result is the interpretability of more data for forensic investigations. However, reverse engineering is time-consuming for the amount of in-vehicle devices and various implemented technologies.

We were able to store the collected information in a secure manner. However, state-of-the-art vehicles do not offer tamper-proof storage. As a result, the integrity of in-vehicle data is not guaranteed. The collected data is trustworthy while the data on the in-vehicle components can be manipulated. Dedicated tamper-proof storage is a potential solution, hindering manipulation of data. However, a dedicated tamper-proof storage comes with additional costs.

By using OBD, we were able to collect lots of different sources. However, not all relevant devices are connected to the diagnostic interface. A dedicated security device is a possible solution to this problem. This component has access to all relevant and predefined components to collect data. As a result, fast triage and data collection are feasible. On the other hand, such a device introduces a highly valuable target for attackers. Furthermore, additional costs are a negative point too.

7. Conclusion and future work

This paper focused at the questions if forensic analyses on state-of-the-art vehicles are feasible and which gaps can be identified. To answer those questions, we presented a generic automotive forensics process that allows for reproducibility offers adaptability, and provides generality. It consists of four phases. Forensic readiness to the determine the forensic capabilities of the area of inters, data collection to gather forensically valuable data, data analysis to process and interpret the captured information, and documentation to finalize the forensic investigation in a final report. The process was implemented to analyze potential in-vehicle component manipulation in a state-of-the-art vehicle. By using the OBD interface, data was collected from different ECUs. The analysis captured network traffic between the vehicle and an analysis laptop using Wireshark and a Python framework implementing the UDS and DoIP standard. As a result, we are able showing by example that forensic analysis on modern vehicles is feasible. However, different gaps were identified during our research. The used tool-set offers limited capabilities for vehicles since they implement various sets of different technologies. Furthermore, there is no tamper-proof data storage available in the vehicle. Competent attacker groups could manipulate the information and decrease the forensic value of the data. Besides, no dedicated storage device for security-relevant data is present in state-of-the-art vehicles. Such a component would allow fast forensic acquisitions and secure storage of the data.

This research is a first step into evaluating digital forensics techniques in the area of automotive. We did not offer a complete solution to solve issues in the field of automotive forensics. In future work, we will focus on the expansion of the used tool-set to allow implementing forensic analysis on different technologies. Furthermore, systems will be introduced to solve the identified gaps in modern vehicles. This includes dedicated security storage systems and assisting devices to allow forensics analysis on cars.

References

10 Böhm, K., Nitsche, A., Birke, K., Schweiger, H.-G., 2017. Application of vehicle control units as event data recorders in hybrid and electric vehicles. In: EVU Congress 2017.

Bryson, J.M., mar 2004. What to do when stakeholders matter. Publ. Manag. Rev. 6 (1), 21–53.

Casadei, F., Savoldi, A., Gubian, P., 2006. Forensics and sim cards: an overview. International Journal of Digital Evidence 5 (1), 1–21.

Corbett, C., Schoch, E., Kargl, F., Preussner, F., 2016. Automotive ethernet: security opportunity or challenge? In: Meier, M., Reinhardt, D., Wendzel, S. (Eds.), Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit. Gesellschaft für Informatik e.V., Bonn, pp. 45–54.

daSilva, M., 2007. Analysis of Event Data Recorder Data for Vehicle Safety Improvement.

Feng, X., Dawam, E.S., Amin, S., 2017. A new digital forensics model of smart city automated vehicles. In: 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)., pp. 274–279.

Geschonneck, A., 2014. Computer-Forensik: Computerstraftaten Erkennen, Ermitteln, Aufklären, sixth ed. dpunkt-Verlag.

Gomez Buquerin, K.K., 2018. Security Evaluation for the Real-Time Operating System Vxworks 7 for Avionic Systems.

Gomez Buquerin, K.K., 2019. Analysis of Digital Forensics Capabilities on State-Of-The-Art Vehicles (Master's thesis).

Herold, N., Posselt, S.-A., Hanka, O., Carle, G., 2016. Anomaly Detection for Some/ip Using Complex Event Processing, pp. 1221–1226.

Hoppe, T., Kuhlmann, S., Kiltz, S., Dittmann, J., 2012. IT-forensic automotive investigations on the example of route reconstruction on automotive system and communication data. In: Ortmeier, F., Daniel, P. (Eds.), Computer Safety, Reliability, and Security. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 125–136.

Hossain, M., Hasan, R., Zawoad, S., jun, 2017. Trust-IoV: a trustworthy forensic investigation framework for the internet of vehicles (IoV). In: 2017 IEEE International Congress on Internet of Things (ICIOT). IEEE.

Isermann, R., Schaffnit, J., Sinsel, S., 1999. Hardware-in-the-loop simulation for the design and testing of engine-control systems. Contr. Eng. Pract. 7 (5), 643–653. http://www.sciencedirect.com/science/article/pii/S0967066198002056.

05 Kiltz, S., Hildebrandt, M., Dittmann, J., 2009. Forensische datenarten und -analysen in automotiven systemen. In: DACH Security. Syssec, pp. 141–152.

Litman, T., 2017. Autonomous Vehicle Implementation Predictions. Victoria Transport Policy Institute Victoria, Canada.

Manadhata, P.K., Wing, J.M., May 2011. An attack surface metric. IEEE Trans. Software Eng. 37 (3), 371–386.

Matheus, K., Königseder, T., 2017. Automotive Ethernet, second ed. Cambridge University Press.

Nilsson, D.K., Larson, U.E., 2008. Conducting forensic investigations of cyber attacks on automobile in-vehicle networks. In: Proceedings of the 1st International Conference on Forensic Applications and Techniques in Telecommunications, Information, and Multimedia and Workshop. E-Forensics '08. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), ICST, Brussels, Belgium, Belgium, p. 8, 1–8:6. http://dl.acm.org/citation.cfm?id=1363217.1363228.

Park, J., Chung, H., Lee, S., 2012. Forensic analysis techniques for fragmented flash memory pages in smartphones. Digit. Invest. 9 (2), 109–118. http://www.sciencedirect.com/science/article/pii/S1742287612000643.

Preneel, B., 1994. Cryptographic hash functions. Eur. Trans. Telecommun. 5 (4), 431–448.

Raghavan, S., Mar 2013. Digital forensic research: current state of the art. CSI Transactions on ICT 1 (1), 91–114. https://doi.org/10.1007/s40012-012-0008-7.

Walter, E., Walter, R., nov 2018. Data acquisition from light-duty vehicles using OBD and CAN. SAE International.