Reliability Engineering and System Safety 96 (2011) 11–25

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

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

Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures

Tobias Hoppe *, Stefan Kiltz, Jana Dittmann

Otto-von-Guericke University of Magdeburg, ITI Research Group on Multimedia and Security, Universitaetsplatz 2, 39106 Magdeburg, Germany

ARTICLE INFO

Available online 17 July 2010

Keywords:
Automotive IT
Practical attack scenarios
Interplay between security and safety
Countermeasures
Automotive intrusion detection
Automotive IT-forensics

ABSTRACT

The IT security of automotive systems is an evolving area of research. To analyse the current situation and the potentially growing tendency of arising threats we performed several practical tests on recent automotive technology. With a focus on automotive systems based on CAN bus technology, this article summarises the results of four selected tests performed on the control systems for the window lift, warning light and airbag control system as well as the central gateway. These results are supplemented in this article by a classification of these four attack scenarios using the established CERT taxonomy and an analysis of underlying security vulnerabilities, and especially, potential safety implications.

With respect to the results of these tests, in this article we further discuss two selected countermeasures to address basic weaknesses exploited in our tests. These are adaptations of intrusion detection (discussing three exemplary detection patterns) and IT-forensic measures (proposing proactive measures based on a forensic model). This article discusses both looking at the four attack scenarios introduced before, covering their capabilities and restrictions. While these *reactive* approaches are short-term measures, which could already be added to today's automotive IT architecture, long-term concepts also are shortly introduced, which are mainly *preventive* but will require a major redesign. Beneath a short overview on respective research approaches, we discuss their individual requirements, potential and restrictions.

1. Introduction and motivation

The complexity of current automobiles is constantly increasing. Modern cars contain a variety of electronic control units (ECUs) that are connected to each other via different kinds of bus systems in order to reduce the amount of cables needed.

But this growing complexity and added functionality might increasingly attract attackers to misuse these systems for their individual purposes, which has already been speculated about by IT security researchers like Eugene Kaspersky [1]. Another factor is the trend of increase in information exchange between automotive systems and the outside world: For example, Barisani and Bianco [2] demonstrated a technique to inject forged traffic information into navigation systems using the wireless protocols Radio Data System (RDS) and Traffic Message Channel (TMC). Further, future technologies like car-to-car (C2C) [3] or car-to-infrastructure (C2I) communication are already being discussed to implement several new automotive applications.

Looking at these trends and the interplay between potential IT security related attacks and the high safety risks in this domain of fast-moving computing systems, automotive IT security is an important emerging area of research: Unlike within typical home PC systems, a successful security violation on an automotive IT system might not only cause nuisance and disclosure of sensitive data but also directly endanger the safety of its human users (drivers, occupants, etc.) and environment [4].

In this article we illustrate that today already the IT security of current automotive systems has to be addressed more forcefully. We demonstrate this by summarising results of four attack scenarios S1–S4 targeting the components electric window lift (S1), warning lights (S2), airbag control system (S3) and gateway ECU (S4). These attack scenarios, which we implemented practically for previous publications using current automotive hardware based on the controller area network (CAN) bus system [5], are systematically analysed in this article using the CERT taxonomy [6]. We further identify basic security weaknesses exploited in these tests and the potential implications to the safety to discuss exemplary countermeasures for the future. Suggestions for holistic approaches for long-term solutions are already under research. However, these have special requirements with respect to the underlying hardware, general system design, etc., so a complete implementation cannot be

* Corresponding author.
E-mail addresses: tobias.hoppe@iti.cs.uni-magdeburg.de (T. Hoppe), stefan.kiltz@iti.cs.uni-magdeburg.de (S. Kiltz), jana.dittmann@iti.cs.uni-magdeburg.de (J. Dittmann).

0951-8320/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ress.2010.06.026

expected to be available in the next few years. In this article we therefore focus on short-term countermeasures, which could also be added to the existing automotive IT technology to address the basic weaknesses identified so far. This includes intrusion detection technology as well as proactive IT-forensic measures designed to support a data analysis in order to reconstruct incidents inside the automotive IT system. Both concepts might help to achieve a reasonable security compromise until and beyond such a major redesign.

The article is structured as follows: in the following Section 2 we shortly present the state-of-the-art of automotive IT security measures, starting with existing applications. In Section 3 we describe our practical tests investigating attacks in four scenarios S1–S4 on exemplary automotive components. These tests, which have been partly extended for this publication, are used to identify violated security aspects and sensitise for potential impacts to safety and comfort. In Section 4 we discuss potential countermeasures (some of which have been demonstrated practically already) as well as their potential and restrictions. This includes intrusion detection techniques and proactive IT-forensic measures, which are discussed for the application in the automotive domain. After an outlook on future, more holistic solutions in Section 5, Section 6 concludes this article with a summary and an outlook.

2. State-of-the-art

This section discusses the state-of-the-art of IT security in today's automotive IT systems, referring to exemplarily selected applications and measures. For this article, this serves as a basis stating how the current state-of-the-art in automotive IT security can be assessed and what aspects remain as a task of research. Whilst car manufacturers have improved the safety of their automobiles a lot during the past decades, adequate holistic concepts for IT security are not available yet. However, since IT security related incidents can also affect the safety of automotive IT systems, this interplay between safety and security clearly motivates automotive IT security as a research topic with increasing relevance and importance.

2.1. Exemplary preventive measures

As state-of-the-art, IT security mechanisms based on encryption or digital signatures can already be found in today's cars (see [7,8]) but only in a very local scope protecting single components or functionalities.

Modern anti-theft systems like the central locking system or the immobiliser usually employ cryptographic protocols. One example is the keyless entry, which typically uses a cryptographic challenge-response to protect against replay-attacks: the car generates a random value (challenge), which has to be processed by the key remote control using its secret key. After passing back the correct result, the car doors will be opened. Even if an attacker records the entire communication between the car and the key remote during this process, a replay of these logs does not allow him to enter the car in the absence of the authentic driver. However, such systems have to be designed carefully. Recently, a successful side channel attack on the proprietary system "Keeloq" has been presented in [9]. It yields a manufacturer specific master key allowing an attacker to access every car implementing this algorithm after sniffing two messages from a distance of up to 300 ft.

Other potential attack targets that car manufacturers are trying to protect are the contents of memory chips, especially of rewritable flash memory holding updateable programme code and configuration data. One motivation is the protection of their intellectual property represented by this data. Other threats are posed by common attacker types like car tuners who frequently modify programme code or configuration data to achieve a higher power output (or, increasingly, also less fuel consumption/eco tuning). Since such unauthorised manipulations also affect issues like safety and liability, the integrity of flash updates has to be ensured, as well. In the context of the HIS ("Herstellerinitiative Software") group in Germany [10], several car manufacturers joined and developed a common specification for secure flashing, which employs digital signatures as cryptographic protection mechanism.

Although these examples for sound IT security approaches can already be found in current cars of many manufacturers, they are only covering a very local scope. They are not conceived to provide a holistic protection for the entire system. This is demonstrated in Section 3 by presenting results from practical tests we performed in the context of prior research. Supplemented by an analysis of underlying security weaknesses and potential safety threats, the results illustrate that automotive IT systems currently lack holistic concepts to hinder such dedicated attacks.

2.2. Exemplary reactive measures

While also systems from the desktop IT domain can never be taken for 100% secure, established measures are available there to at least detect malicious incidents at runtime or in retrospect, e.g. by Intrusion Detection Systems (IDS) or IT-forensic investigations. However, both concepts are not fairly established in the automotive domain yet. Instead, at least several safety-focused self-diagnosis mechanisms are put in place. They are designed to guide car service technicians in the process of isolating component failures, i.e. to support the maintenance of the car. For that, a reserved portion of memory contained in modern ECUs is reserved to store diagnostic trouble codes (DTC). Alongside the component that suffered a temporary or permanent component failure (e.g. a sensor or an actuator), some ECUs also record environment conditions during the failure (e.g. engine temperature and rpm). Beyond the maintenance scope, this can also be used to reconstruct events, for instance, to support a car crash investigation.

However, these safety-centred diagnostics systems do not address dedicated attacks. They could partially be useful to find safety-related implications of uncovered security-related incidents but do not meet the requirement of recorded evidence being proven authentic and unaltered. Using today's mechanisms, some data within the scope of the self-diagnosis property of modern ECUs can be retrieved, which in turn can be useful in a root/cause data analysis also after IT security related incidents. To investigate malicious activity inside an automotive IT infrastructure, however, other mechanisms apart from the self-diagnosis property of ECUs have to be implemented.

As a main finding, the current state-of-the-art in automotive security only contains local security concepts and large parts of the entire system are not protected against any malicious activities.

3. Four exemplary automotive IT security threats to discuss attack potential

In previous work, several practical test setups have been created to analyse and demonstrate IT security threats of current automotive technology. For this article, four significant test setups (denoted as attack scenarios S1–S4) have been selected to

perform a broader analysis of potential safety implications and to define and evaluate first countermeasures. In this section we summarise the basic principles and results of these tests. While some aspects of these tests have been described in more detail in previous publications, some of them have been slightly extended as a new contribution for this publication, also extended with a systematic analysis of the incident, violated security aspects and potential implications in the interplay with safety.

Each of the selected tests could be demonstrated on real, recent automotive hardware. They could be verified in two different technical setups, both containing a wiring harness and different electronic control units (ECUs) from a recent model (built in 2004 and 2005, respectively) of a big international car producer. Cars of both series have a similar technical architecture and use the CAN bus for the communication between the separate devices. Supported by different bus interfaces, a PC system can be used to interact with or investigate the automotive system. Fig. 1 illustrates the basic concept of the technical setup.

In the following four subsections, the practical examples will be presented and analysed in the following structure. First, the implementation of the attack is explained and the results of the attack are presented. Afterwards, a potential security incident that could have employed the described attacking technique is discussed by using the CERT taxonomy [6]. Table 1 shows its main structure. Incidents are classified by identifying the kind of *attackers*, the *tools* they used as well as the *vulnerabilities* they exploited. Elementary *actions* and their *targets* are identified together with the *unauthorised results*. Also the *objectives* of the attacker are examined in order to understand the underlying motivation. As the horizontal bars indicate, the actual *attack* and *events* occurring in its context are subordinate phases of the entire *incident*.

Following the CERT classification, each attack scenario is analysed with respect to the violated security aspects and potential implications, especially in the interplay with safety.

In this article we use the five central security aspects known from IT security (confidentiality, integrity, availability, authenticity and non-repudiation) to identify underlying weaknesses and to analyse potential countermeasures. In the context of this section, the violation of security aspects in the four attack scenarios S1–S4 has been evaluated with a focus on the digital information communicated via the car's internal bus networks (see Table 2).

3.1. Scenario S1: analyses on the electric window lift

The first potential attack target we investigated was the electric window lift system. For early practical tests performed on this scenario a simulation environment was used. This was a simplified car environment being part of CANoe, an established development and simulation software from Vector Informatik [11] widely used in the automotive industry.

In this test, a few lines of malicious code have been added to an arbitrary ECU attached to the simulated comfort CAN subnetwork. Once a predefined condition is met (in this case when the car's speed exceeds 200 km/h) the code replays the CAN message containing the flag for opening the driver window. Although the real console still sends its messages in the same frequency indicating that no button is currently pushed, the simulated window opens and will not close again until the end of the attack. Even if the driver has a fast reaction and pushes the "close" button while the window is still opening, it blocks and remains stuck. More details about this test can be found in [12] (as well as in [4] and [13]).

Meanwhile, the completion of the aforementioned physical test setups allowed us to demonstrate similar results on a real window lifter (being part of the door control modules in our practical test setup, see Fig. 2) during a student project. After identifying the CAN messages relevant for triggering the window

Table 1
Main structure of the CERT taxonomy [6].

Incident						
	Attack					
			Event			
Attackers	**Tool**	**Vulnerability**	**Action**	**Target**	**Unauthorised result**	**Objectives**

Table 2
Security aspects mapped to information within automotive bus systems.

Confidentiality	Is communicated information read by unauthorised nodes?
Integrity	Is there communicated information which is semantically incorrect (and potentially processed by unaware nodes)?
Availability	Is information, which is to be communicated, accessible by all requiring nodes? Is an implemented service applicable?
Authenticity	Has the communicated information been created by an authorised node (i.e. stems from the expected sender)?
Non-repudiation	Can a node prove that it did or did not communicate certain information?

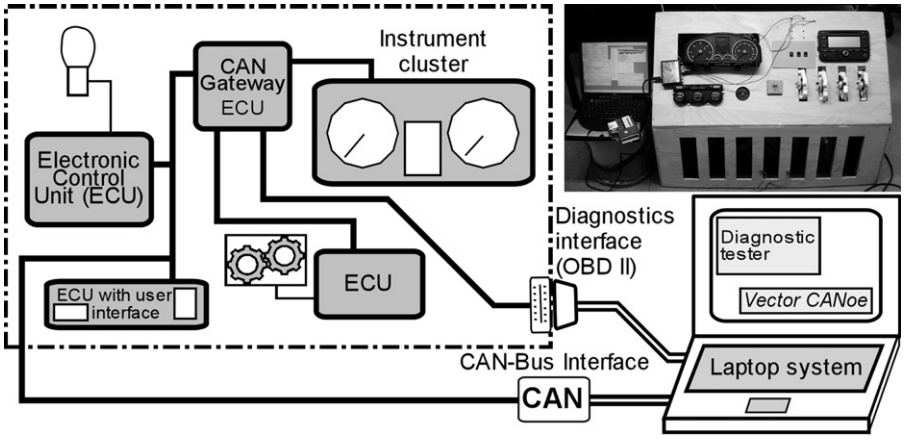

Fig. 1. Illustration of the practical test environment of automotive hardware.

lifts, an attack strategy similar to the simulated attack has been conceived: every time a CAN message is observed on the comfort CAN subnetwork containing a flag set to open the window, a new copy is generated onto this bus specifying an opposite (close) or cleared (no action) flag.

This attack scenario constitutes a Denial of Service (DoS) attack on the functionality of the window lift system.

3.1.1. CERT classification

Opening a window once the car exceeds a speed of 200 km/h might be motivated by the pure challenge or thrill of "hacking" (*objectives*). The respective *attacker* might be a hacker who intends to prove his skills. As a *tool* he uses malicious code. The *vulnerability* that enables this attack lies within the design of the CAN bus (e.g. that received messages cannot be authenticated). The malicious code performs "Read" and "Spoof" *actions* to observe the current speed and spoof the window commands. The *target* is the left door ECU. As *unauthorised result* of the attack, the window opens and the system blocks until the end of the attack, which constitutes a Denial of Service (DoS) attack onto this system. This exemplary CERT classification of scenario S1 is illustrated in Table 3.

In an additional scenario not respected in Table 3, the attacker might inject malicious code in a device having access to the comfort CAN subnetwork to open the windows in absence of the driver (e.g. during the night). In this case, the attacker might be a thief motivated by financial gain as his main objective.

3.1.2. Implications to comfort, security and safety

The implications of a successful implementation of this scenario can affect each of the domains comfort, security and safety.

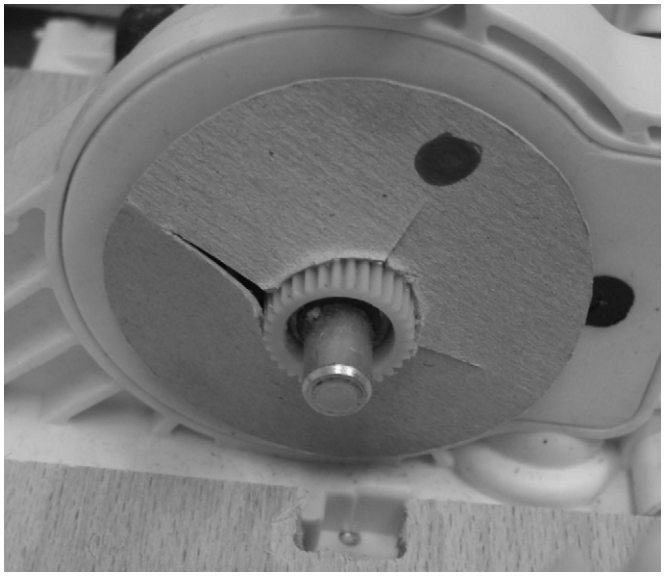

Fig. 2. Electric window lift in the laboratory test setup.

The *comfort* can be reduced because of potential inconveniences caused by the stuck window.

The following *security* aspects are violated by the attack: The authenticity of control commands is violated, since these are not sent by the authentic console. Also the integrity of the switch status transmitted this way is affected, because it does not correspond to the actual button state. As an overall implication, the availability of the entire window lift system is violated by the DoS attack: The windows do not respond correctly to authentic requests until the attack finishes. Also the non-repudiation is affected, since the authentic console cannot prove that the close-commands have not been sent by itself.

In the scenario discussed, also implications to the *safety* might arise. A relevant factor could be that the opening window, occurring all of a sudden and at high speed, affrights the unsuspecting driver. In such a case, he might overreact (e.g. hectically trying to close it again) and lose control over the car (e.g. by not focusing on the traffic ahead).

3.2. Scenario S2: analyses on the warning lights

As a second target, the warning lights (the indicators) have been analysed. Amongst others, the anti-theft system triggers them once an intrusion into a parked and secured car is detected. A common scenario is the unauthorised opening of a door. Triggered by a corresponding event from the door contact sensor, the door control module reports this event to the comfort system ECU, which also contains some of the anti-theft system functionality. Now, an alarm is generated for a few minutes by sending alternating command telegrams to the vehicle electronics ECU to set or unset the warning lights (and, frequently, the alarm horn as well).

This scenario served as another test case. In our evaluation we found that every component with access to the comfort CAN subnetwork (this might be an original ECU after the injection of malicious code or a device attached additionally like a developer circuit board) can heavily interfere with this process by immediately sending an "off" command once an "on" command (sent by the comfort system ECU) is observed. Even though the "on" commands do not get removed from the comfort CAN subnetwork, in our tests [13] this attack proved to be quite powerful: The indicator bulbs (see Fig. 3) stay completely dark most of the time, while (apparently due to timing reasons) sometimes only a short, weak glowing appears (though this is not expected to be noticeable through orange glass covers).

3.2.1. CERT classification

A relevant *attacker* for this scenario might be a thief intending to steal the car or items from its interior (i.e. financial gain as *objectives*) and to remain undiscovered while doing so. As his *tool* he uses malicious code that he attaches to the comfort CAN subnetwork (in some cars respective bus wires can also be found in exposed locations like the exterior mirrors). The underlying *vulnerability* is rooted, as in the previous scenario, in the design of the CAN bus (as in scenario S1). Again, the malicious code performs "Read" and "Spoof" *actions* to observe command messages to set the warning lights, immediately spoofing an

Table 3
CERT classification of scenario S1.

Attackers	Tool	Vulnerability	Action	Target	Unauthorised result	Objectives
Hackers	Malicious code (attached elec. circuit)	Design (CAN)	Read	Control unit (left door ECU)	Blocking of window system (DoS)	Challenge/status/thrill
			Create/Spoof		Violation of security aspects (see below)	

unset message. The *target* is the vehicle electronics ECU that processes these commands. As *unauthorised result* of the attack, the indicator bulbs stay off, which also represents a Denial of Service (DoS) attack onto this system. This exemplary CERT classification of scenario S2 is illustrated in Table 4.

3.2.2. Implications to comfort, security and safety

As a result of such an incident, an unnoticed theft of the car or valuable items from the interior might arise. The implications of a successful implementation of this scenario can mainly affect the security and safety domain.

The following *security* aspects are violated by the attack: The authenticity and integrity of control commands is violated, since these are not sent by the authentic ECU implementing the anti-theft functionality and the content of the spoofed commands to unset the light does not comply with the intended state (set). As an overall implication, the availability of the indicators is violated by the DoS attack since they do not respond correctly to any actuation command. Also the non-repudiation is affected, since the authentic ECU cannot prove that it did not generate the forged unset-commands.

While for this attack target *comfort* implications are hardly relevant, the employed malicious code could also affect the *safety*, e.g. if it activates while the car broke down and hinders it to send a warning to other road users.

Fig. 3. Suppressed indicator bulb.

3.3. Scenario S3: analyses on the airbag control system

Another automotive component which we checked for security vulnerabilities was the airbag control system. In this attack, which is described in more detail in [14], the airbag control system can be removed from the system, but the loss of functionality is masked by some bogus, injected logic. Obviously, this attack can endanger the car's occupants (by the loss of a safety system). Though, in principle, this might also be an intended aim of the attack, more probable motivations would by monetary interests: After crash events, airbags frequently have to be substituted with expensive, new parts. As more and more police and press reports state, the theft of airbag systems is already quite common—and thieves might try to avoid the loss being detected. Also re-sellers of used cars might try to hide a defect of the airbag system in order to save repair costs and achieve higher sales prices.

Within the attack examined, the attacker tries to suppress several signs of the non-functional system, which might sooner or later raise suspicion. One example clearly visible to the driver is the airbag warning lamp within the instrument cluster, which indicates a failure (or absence) of the airbag control system (see left part of Fig. 4). Another sign would be the failure of a communication with the "defective" system using the diagnostics protocol, which might be performed in the car service station by connecting to the car's diagnostics interface.

In [14] we managed to emulate the behaviour of a fully functional airbag control module within a diagnostics session by any device with access to the powertrain CAN subnetwork (where the removed system also was attached to). In practice this might be another original device after some software manipulation or an additionally attached cheap circuit board; in our tests we used a PC system attached to the powertrain network via the CAN bus interface. After recording the reactions to diagnostic queries during a regular diagnostics session, these replies could successfully be replayed in the absence of the airbag control module. The diagnostics software reports the presence of the device (including its name, part no., etc.) and attests the absence of any error conditions.

Since this technique only covers the diagnostics protocol so far, it does not yet also lead to an expiration of the airbag warning light within the instrument cluster, which is triggered by the CAN gateway ECU. To monitor the presence of each other, ECUs generally do not use the diagnostics protocol. Instead, they monitor other status messages usually transmitted by the respective device—in this case by the airbag control module. After identifying the relevant CAN message that the gateway ECU expects from the airbag control module, this could be addressed by also emulating the periodic communication of the airbag control system (next to the diagnostics protocol already covered) [15]. By replaying the respective message

Fig. 4. Suppression of the airbag warning lamp.

in its original frequency onto the powertrain CAN subnetwork, the malicious device can also pretend the presence of the airbag system among the other ECUs.

Since the identified message also contains a bit flag to set and unset the airbag lamp in the instrument cluster, the malicious device also has full control over this indicator. Consequently, it can disable the warning lamp during runtime (see right part of Fig. 4) and could also emulate a successful startup check (which is usually expected in form of a short blinking signal after the ignition).

3.3.1. CERT classification

As explained below, the *attacker* might be a re-seller of the car trying to reduce repair costs and to increase the sales price, therefore having financial gain as *objectives*. As a *tool* he uses malicious code (e.g. in form of an electronic circuit board), exploiting the same design *vulnerability* as above. The performed *actions* include reading of diagnostic request and spoofing the expected reply (which has been copied from an authentic session beforehand). The periodic status message (also copied from a fully operational system beforehand) is spoofed as well. Relevant *targets* are the instrument cluster and a diagnostic tester. An exemplary *unauthorised result* is the theft of resources, which is the functionality of the airbag system. This exemplary CERT classification of scenario S3 is illustrated in Table 5.

3.3.2. Implications to comfort, security and safety

While not reducing comfort (the driver will not notice any lack of functionality in regular operation), this attack scenario violates security aspects and can have severe safety implications.

The following *security* aspects are violated by the attack: The authenticity of the periodic status messages and of diagnostic reply is violated, since they are not generated by the airbag ECU but by a bogus device. Also the integrity of the periodic status information and the diagnostic data is concerned, since they do not correspond to the true system state. As an overall implication, the availability of the entire airbag system is violated. While the occupants do not notice this flaw during normal usage, the airbags will not be available in emergency cases.

Consequently, clear and severe implications to the *safety* are evident. Injuries in case of accidents can be expected to be a lot more severe in the absence of a functional airbag system. So this attack scenario would be an exemplary target for proactive forensic measures described in Section 4.2 of this publication.

3.4. Scenario S4: analyses on the gateway ECU

Another security evaluation was performed on the central gateway ECU, which interconnects several internal and one external CAN subnetworks (as indicated in Fig. 1), while the external CAN is connected to the on-board diagnostics (OBD) interface. Using this interface, which is freely accessible from the car's interior, this is the first of our security analyses which does not even require hooking up to any internal bus wires, directly.

The gateway ECU implements basic filtering functions with respect to the internal communication of the car. This means, that some messages are to be forwarded by the gateway ECU into a different subnetwork for further processing. Others are only needed within their original subnetwork and therefore are not forwarded anywhere. This way, the gateway ECU implements a kind of network isolation, which saves bandwidth and especially reduces the impact of harmful messages that are accidentally or intentionally sent. Additionally, the internal communication is not accessible from the open diagnostics interface during normal operation.

By sniffing a regular diagnostics session and analysing the basic context of the respective protocol, an implementation flaw in the implementation of the gateway ECU could be identified and exploited. The analysis revealed that the CAN message-IDs to be used during the diagnostics session are negotiated between the testing and the tested device within the initialisation of the session. The gateway ECU unlocks these IDs for forwarding between the respective subnetworks for the duration of the session. However, in the present implementation, the gateway ECU does not check if the negotiated CAN IDs violate the range reserved for such sessions. By sending manipulated initialisation requests, the gateway ECU can be induced to also forward arbitrary internal CAN messages to the outside. This is caused by the fact that the gateway ECU cannot tell the original, internal messages apart from the messages belonging to the active session, if all of these use the same message type. As the internal, tested ECU replies using a reserved CAN ID, this attack also bears the potential to indirectly write into the respective internal network, since the other ECUs might misinterpret these ambiguous messages as well. Fig. 5 shows a screenshot of the prototypical attack demonstrator during an active sniffing attack.

The technical implementation of this attack has been described in more detail in [16] with a focus on the frequent black-box perspective of automotive attackers as well as on increase in privacy threats to automotive IT systems.

3.4.1. CERT classification

Eavesdropping internal information while minimising any intrusive actions (i.e. by simply attaching a self-contained plug-in device as a *tool* at the freely accessible OBD interface) might be performed by a spy as *attacker*. In such a scenario, the *unauthorised result* is the disclosure of information. This might be personal data like person related or person relatable information (e.g. personal driving habits, frequent routes, etc.) or dialled phone numbers from the infotainment subnetwork. He might act out of financial gain as *objectives*, e.g. for extortion purposes or to sell such information to others. In this scenario, the *vulnerability* is an implementation flaw within the gateway ECU, which does not prevent the usage of reserved CAN IDs for diagnostic connections. As *actions*, the tool creates a forged diagnostics request, bypasses the implemented network isolation and reads out arbitrary internal communication. Consequently, the internal bus systems and the transmitted information are *targets* of the attack. This exemplary CERT classification of scenario S4 is illustrated in Table 6.

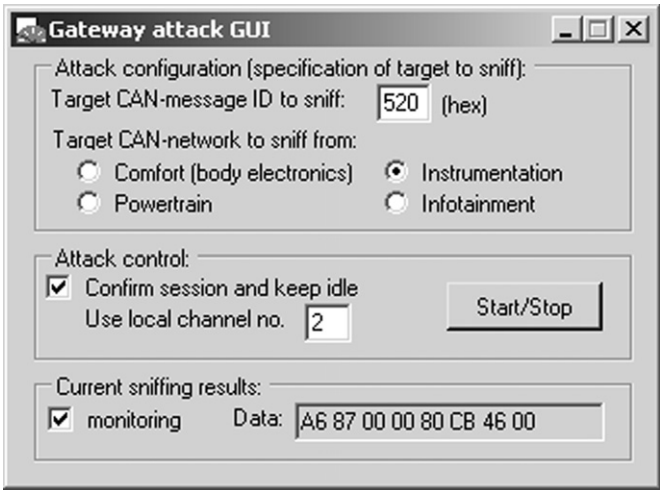

Fig. 5. Graphical user interface (GUI) of the prototypical attack demonstrator.

3.4.2. Implications to comfort, security and safety

The implications of a successful implementation of this scenario affect security and safety.

With respect to the *security* aspects, especially the confidentiality of eavesdropped internal messages is violated. In case of personal or person relatable information, also the privacy can be affected. Looking at the diagnostics request injected, also the authenticity of these requests is violated, since they are not sent by an authentic diagnostic tester but by a forged device. (If the described technique is used to indirectly inject arbitrary messages into the internal subnetworks – which is not within the scope of this scenario – also the integrity and authenticity of internal communication would be affected).

When employing the described attack to eavesdrop internal communication, also implications to the safety could arise (at least as a potential side-effect). Since the used internal ECU replies to the incoming request with at least one message containing the forged, reserved CAN ID, other devices within the target network might misinterpret its content (just like the gateway ECU does with the authentic messages of that type). If safety-relevant applications process such a misinterpreted message, its occurrence may lead to a safety relevant system malfunction.

3.5. Safety and security implications—summary and analysis of underlying problems in scenarios S1–S4

Main examples for violated security aspects that have already been mentioned during the analyses of S1–S4 are summarised in Table 7. Besides the identified examples for potential safety implications, also examples for further kinds of implications are listed.

Based on these exemplary vulnerabilities exploited in the scenarios S1–S4, in the following we identify basic weaknesses in today's automotive systems that make such attacks possible.

While this is done with a focus on the violated security aspects, this also addresses potential safety implications (like those summarised in Table 7), which can arise from successful violations of IT security. This analysis of current weaknesses in automotive IT security also serves as a basis to discuss potential countermeasures for future systems in Section 4.

In the first three practical scenarios S1–S3 we accessed the car's IT infrastructure from within its internal bus systems. In the scope of this article, we do not focus on the question, what technique a potential attacker might have chosen to get into this position. As already mentioned earlier, he might simply have placed some additional circuit board onto the bus wires, like we did with the CAN bus adapter we used (on most current cars adequate, exposed positions containing wires of the corresponding buses can be found). The example presented in scenario S4 proved that attacks can also be established without directly gaining access to an internal (bus) system. The diagnostics interfaces, which are usually freely accessible from the car's interior, are increasingly also made available via wireless protocols like Bluetooth, for example. But also with respect to scenarios like S1–S3, an attacker could reduce the required amount of physical access and equipment by injecting malicious code into an existing device, e.g. by exploiting unsecured diagnostics interfaces, manipulated update discs for media systems distributed by social engineering or exploiting potential weaknesses of wireless communication systems (like future C2C/C2I systems).

The exemplary attacking strategies that we utilised in scenarios S1–S3 primarily exploited drawbacks of the CAN bus protocol frequently employed in today's automobiles (see CERT analysis in Tables 3–5). For this reason we concentrate on discussing exemplary requirements for a secure automotive bus communication, using the CAN bus as example.

Though the CAN bus does provide measures to ensure aspects like the integrity of the transmitted information from the functional safety perspective (protection against unintended transmission errors by cyclic redundancy checks (CRC)), the existing measures do not meet the requirements from the IT

Table 4
CERT classification of scenario S2.

Attackers	Tool	Vulnerability	Action	Target	Unauthorised result	Objectives
Thieves	Malicious Code (attached elec. circuit)	Design (CAN)	Read Create/ Spoof	Control Unit (vehicle electronics ECU)	Blocking of warning light system (DoS) Violation of security aspects (see below)	Financial gain

Table 5
CERT classification of scenario S3.

Attackers	Tool	Vulnerability	Action	Target	Unauthorised result	Objectives
Re-Seller	Malicious code (attached elec. circuit)	Design (CAN)	Read Create/spoof copy	Control unit (instrument cluster) Diagnostic tester	Theft of resources (airbag functionality) Violation of security aspects (see below)	Financial gain

Table 6
CERT classification of scenario S4.

Attackers	Tool	Vulnerability	Action	Target	Unauthorised result	Objectives
Spies	Malicious code (plug at OBD interface)	Implementation (Gateway ECU)	Read Create/spoof bypass	Bus system, internal data	Disclosure of information Violation of security aspects (see below)	Financial gain (e.g. for extortion)

Table 7
Summary of violated security aspects and potential implications (to safety, comfort, etc.).

Scenario, targeted component and aim	Violated *security* aspects	Potential *safety* threats	Further implications
S1 (electric window lift): unauthorised actuation and DoS	Authenticity (control messages) Integrity (switch status) Availability (window lift) Non-repudiation (source ECU)	Affrighted driver loses control over the car	Comfort (inconveniences due to a stuck window that cannot be closed)
S2 (warning light system): DoS	Authenticity (control messages) Integrity (actuation command) Availability (indicators) Non-repudiation (source ECU)	Potential accidents due to unavailable warning lights	Unnoticed theft of the car or valuable items from the interior
S3 (airbag control system): faking a healthy system state	Authenticity and integrity (reported system status) Availability (defective or removed airbags)	Missing protection by defective airbags in emergency cases, probably higher severity of injuries	Potential warranty claims by victims towards sellers, etc. Loss of road worthiness certificate (in case of disclosure)
S4 (gateway ECU): eavesdropping internal information	Confidentiality (internal messages) Authenticity (incoming diagnostic request)	Potential malfunction of safety-relevant systems due to misinterpreted CAN messages	Loss of person related or person relatable information (e.g. personal driving habits, frequent routes, etc.)

security perspective. For example, a CRC checksum is not sufficient for detecting falsified contents of a CAN message, which has intentionally been generated by an attacker—just because he would also re-adjust the CRC information accordingly.

When looking at the IT security aspects listed at the beginning of Section 3, as of this writing no sufficient assurance measures are provided at the CAN bus level yet.

3.5.1. Confidentiality/privacy

A message sent onto a CAN bus is at least received by all other ECUs connected to that bus system. Based on the type identifier (ID) of the message, each ECU decides if or if not to use it. If a gateway ECU is amongst these nodes and transmits the message into another subnetwork, even more nodes are affected. Due to this broadcast character of CAN, each of the receiving nodes can principally read up to 8 bytes transported with each message. However, in some applications the transmitted information might be regarded as confidential; by collecting information from CAN bus systems, an attacker could, for example, be empowered to conclude privacy-relevant information (e.g. driving behaviour) of the current (or during diagnostic sessions even about previous) drivers. In scenario S4, the sniffing of internal communication was even made possible from the restricted diagnostics port. However, if the exploited filtering flaw of the gateway ECU would be patched, the direct access to the internal bus systems would still be quite easy. Approaches like encryption or anonymisation could reduce such threats. Looking at direct implications to safety, violations of the security aspect confidentiality can be considered as less relevant.

3.5.2. Integrity

Relevant violations of the integrity of communicated information could be found in the scenarios S1–S3. The violated integrity of incoming CAN messages cannot be detected by the receiving nodes, if the contained checksum is correct. With reference to the example given at the beginning of this section, a checksum is not a sufficient measure to ensure integrity from the IT security perspective. Appropriate measures known from desktop IT would be cryptographic hash functions, message authentication codes (MAC) or digital signatures, which cannot be "re-adjusted" by an attacker without the knowledge of a secret (private) key. Since forged information processed by safety-relevant systems can have a huge impact to the behaviour of the car, violations of integrity can be considered as highly relevant with regards to safety.

3.5.3. Authenticity

The CAN bus protocol provides no authenticity measures; CAN bus messages do not even contain a sender or receiver address. If a node is not configured to be a regular receiver of the respective type of message (with respect to its ID), the message and its contents are ignored. The usual sender of each message type is implicitly known, but a node has no possibility to verify this assumption. As our practical tests show, malicious nodes can easily spoof messages usually sent by others. Receiving devices cannot detect that these originate from a non-authentic source, rely on the forged contents and consequently perform unauthorised actions. In future automotive networks this could be addressed, e.g. by MACs or digital signatures. Especially if a safety-relevant system processes information generated by unauthorised nodes, violations of authenticity can also be considered as having a high potential impact to safety.

3.5.4. Availability

Using techniques like repeatedly sending unauthorised error flags or high-priority messages, a malicious node can easily overload an entire CAN (sub-) network. During such a DoS attack, none of the other devices in this network would be available from the bus level. As it has been illustrated by the targeted injection of forged messages in the scenarios S1 and S2, DoS effects can also be achieved without overloading the bus or target system with message flooding. So ensuring availability in the face of DoS-attacks is a difficult problem in general. The specification of the oncoming FlexRay bus system [17] considers the option of disconnecting malfunctioning devices or branches from the network by node-local or central "bus guardians". However, this also seems to be more of a safety measure against unintended malfunctions than to address security viewpoints. If safety relevant systems like the airbag control system in scenario S3 are not available, potential safety risks can be considered as highly critical.

3.5.5. Non-repudiation

After an incident like the spoofing attacks in our practical tests it is hard for the affected devices to deliver proof of their innocence (i.e. that they did really receive such a malicious command or, correspondingly, that they did not send such a message). In the absence of mechanisms for the four aspects above, this is even more difficult to ensure.

After analysing the four attack scenarios, the main findings of this section can be summarised: Due to missing security measures especially at the bus system level, malicious interactions with automotive systems are practically possible and bear potential, partly severe, safety risks.

4. Discussion of short-term countermeasures to address the demonstrated threats, their potential and restrictions

As a consequence of the results presented in the previous section the internal communication of a car will have to be *secured* more in future. This is an essential demand in order to also reduce *safety* implications caused by IT security related incidents. In this section, two exemplary countermeasures are being discussed that could help to increase the IT security of future automobiles by addressing these problems like the basic weaknesses exploited in our practical tests. Some aspects of the discussed approaches have already been examined in our practical test environment.

As mentioned before, a holistic approach obviously would be the best choice. But ensuring a maximum number of the IT security aspects introduced before would require an expensive, major redesign. An outlook on such future measures is given in Section 5, where an overview on some current efforts in automotive IT security research is presented.

While such extensive solutions are expected to be inevitable in the long term, simpler and cheaper solutions might be a way to address the most urgent weaknesses in the near future. Therefore, this section focuses on discussing concepts that might help to address basic weaknesses that allowed our practical tests to succeed. The missing authenticity measures in CAN communication are a main example. To address the problem of increase in attack potential in the automotive domain (as presented in the practical scenarios S1–S4), in this section we discuss two different approaches. These are motivated by the disciplines of Intrusion Detection (Section 4.1) and IT-forensics (Section 4.2), which are today primarily used within the desktop IT domain. The adaptation of these strategies to automotive IT is a novel research topic. Based on previous work (see [18] and [19]), in this section we apply these concepts to the four attack scenarios S1–S4.

4.1. Intrusion detection techniques

Often when a given system has no effective means to prevent some kind of attacks initially, it should at least be tried to automatically detect them. In the desktop IT domain such components are usually called Intrusion Detection Systems (IDS) [20]. Once an incident has been discovered by such a system (having discovered suspicious activity patterns in the network activity or at some end system), it might generate warnings or trigger reactions to limit the consequences of the attack (in that case such systems are often also called Intrusion Response or Intrusion Prevention Systems (IPS)).

A potential application of intrusion detection approaches to automotive systems could be useful as well: in an emergency case when an attack is detected, which has not been thwarted by other existing measures, a warning could be generated to the driver and advise him to perform an appropriate reaction (e.g. stop the car at the next safe position). Automatic, autonomous reactions of an automotive IPS could also be discussed as a further option. However, due to the high safety risks in an automotive environment and the ever-present risk of potential false positive classifications or the choice of inappropriate reactions, such an extension would have to be developed with great care.

4.1.1. Three exemplary detection patterns

With reference to the practically investigated attack scenarios S1–S4, we already identified patterns that could be applied to detect such attacks. Three of these we shortly introduce in this section (while this approach is discussed in more detail within [18]) and illustrate their application with respect to the scenarios from Section 3. Their individual advantages and disadvantages are discussed below in Section 4.1.3.

Pattern 1: increased message frequency

Often CAN messages of a given ID are broadcasted by a single sending device and in a constant frequency. In scenario S1 this applies to the state of the window switches as well as to the message triggering the warning lights in scenario S2. As we demonstrated in the tests, another (malicious) device with access to the respective (sub-) network can simply add contradicting messages of the same type to the bus communication in order to achieve unauthorised actions by the receivers. However, since removing existing messages from the bus is harder to achieve, this often results in a notably higher occurrence rate and frequently changing semantic contents of messages having the respective ID. Such features can serve as a simple detection pattern for this kind of attack, indicating *authenticity* and *integrity* violations as evident in scenarios S1 and S2.

Pattern 2: obvious misuse of message-IDs

In each of the scenarios S1–S4 unauthorised messages have been put on the bus by a device distinct from the original sender. Since the receiving nodes have no proof of the *authenticity* of the message (i.e. if it really has been sent by the original sender), this attack proved to be very effective. However, these injected messages also arrive at the original sending ECU, if this is still attached to the bus (scenarios S1 and S2). Currently, from the perspective of an attacker, there is no serious problem, because that device is not expected to evaluate this type of messages (since usually these are only sent by the device itself). Consequently, with little effort some IDS functionality could be added to any ECU looking for suspicious incoming messages like such ones using its exclusive message ID. For instance, the console ECU in scenario S1 could look for foreign messages containing the window lift controls and raise an alarm in such an exception. This pattern could also be applied to gateway ECUs: given, a gateway is configured to pass messages of type m_a from a subnetwork n_a to another subnetwork n_b, using the (maybe differing) ID m_b. If in this setup a malicious message with the ID m_b is injected to the target network n_b (which would not be visible to the originally sender, who is only responsible to detect forged messages of type m_a in the source network n_a), the gateway would be able to detect this incident (unauthorised use of its exclusive ID m_b within n_b), accordingly.

Pattern 3: low-level communication characteristics

In addition to the techniques chosen in the previous patterns, the last pattern discussed in this subsection uses a substantially different approach to detect forged messages that have been injected into a CAN network from an arbitrary bus location. While the previous patterns only regard information available from the data link layer (OSI level 2, see [21]), we assume that for this purpose also information from the physical layer (OSI layer 1) could be useful. To put a CAN message onto the bus, every ECU has to pass it to some CAN controller that generates the corresponding electrical signal at the bus wires. These controllers are available from different manufacturers (partly as CPU integrated circuitry). While all of them are supposed to fulfil the CAN specifications in the end, it might be possible to identify features characteristic for each individual chip when looking more closely at the electrical signal generated. Such features might be voltage amplitudes and their stability, the shape of the clock edges, propagation delays, signal attenuation due to wire

lengths, etc. While still being within valid intervals or above/below acceptable thresholds, these low-level communication characteristics could be analysed by a special detection unit to identify the authentic device that has sent the current message. Such a system could provide useful additional information allowing the verification of the *authenticity* of sending nodes within CAN networks (without the need of any change to the existing bus specifications). For example in scenario S3, the instrument cluster, if equipped with such functionality, could detect a change in the signal properties of messages sent by the airbag control ECU and, this way, detect the event of unauthorised replacement.

4.1.2. Exemplary practical implementation of detection Pattern 1

In [18] we already demonstrated the automotive application of intrusion detection approaches practically: The above-mentioned "Pattern 1" has been implemented into a prototypical automotive IDS component and successfully tested it within our setup for the attack introduced in scenario S2. It proved to be appropriate to detect the exemplary attack onto the warning light system. As Fig. 6 shows, a traffic light symbol that permanently indicates a healthy system status (the one in the left part of the figure having the lower, green light set) indicates an alert in case a suspicious incident is detected (the one in the right part of the figure having the upper, red light set).

Further concepts of how to communicate security related information to the occupants (especially the driver) have been discussed in [18]. There, a conceptual model is introduced that proposes to use existing multimedia interfaces for this purpose, which are already present in most modern cars. The model furthermore respects the frequently changing environmental conditions within the automotive domain (e.g. changing light or noise levels) to choose an optimised way of communication with respect to the current situation.

4.1.3. Restrictions of Patterns 1–3 and potential for future research

Being part of further research, the automotive application of Intrusion Detection still bears many open questions. To provide some exemplary aspects, this section discusses individual advantages and drawbacks/restrictions of the three detection patterns introduced in Section 4.1.1. For this purpose, the estimated effort and costs for development and implementation are considered as well as the expected effectiveness and performance.

Effort and costs for development and implementation

As exemplary factors we consider the complexity of the respective detection algorithm (development costs), the required amount of changes in the entire system (implementation costs) and additionally required resources (especially costs for hardware).

Patterns 1 and 2 have the advantage that the detection algorithms are very simple to develop and implement: basically only the appearance of certain incoming messages has to be noticed and/or counted. An additional advantage of Pattern 1 is that it could be implemented on a single, central device (with access to the respective bus systems, e.g. at a central gateway), which reduces implementation costs. This is a minor downside of Pattern 2, which requires a distributed (and therefore more expensive) implementation on different devices.

The related effort and costs are the major downside of Pattern 3: the evaluation of respective signal properties can be expected to be complex in development. Furthermore, an implementation would probably require additional (expensive) hardware support. However, as a small advantage a central implementation (e.g. at the gateway ECU) would be sufficient.

Effectiveness and performance

While an implementation of Patterns 1 and 2 could be implemented in a relatively easy and cost-effective way, both have some drawbacks in terms of their effectiveness (i.e. both do not detect some kind of attacks). Pattern 1 is only applicable to messages transmitted cyclically. It cannot be applied to message types that only appear occasionally (i.e. which are only sent event-based, e.g. when a button has been triggered). In comparison, Pattern 2 has the advantage to be operational in a much more versatile manner: each defined CAN message has an authorised sender, which could implement this pattern. However, Patterns 1 and 2 can obviously only be used to detect an incident, as long as the authentic sender is still present and fully functional (i.e. non-manipulated). For example, Pattern 2 is not appropriate to detect the forged messages from attack scenario S3 because of the removed original airbag control unit.

The main advantage of Pattern 3 is its potential to detect a broad spectrum of added or swapped devices. This way it could compensate these restrictions of Patterns 1 and 2. However, if malicious messages are sent by the same device (i.e. the attacker

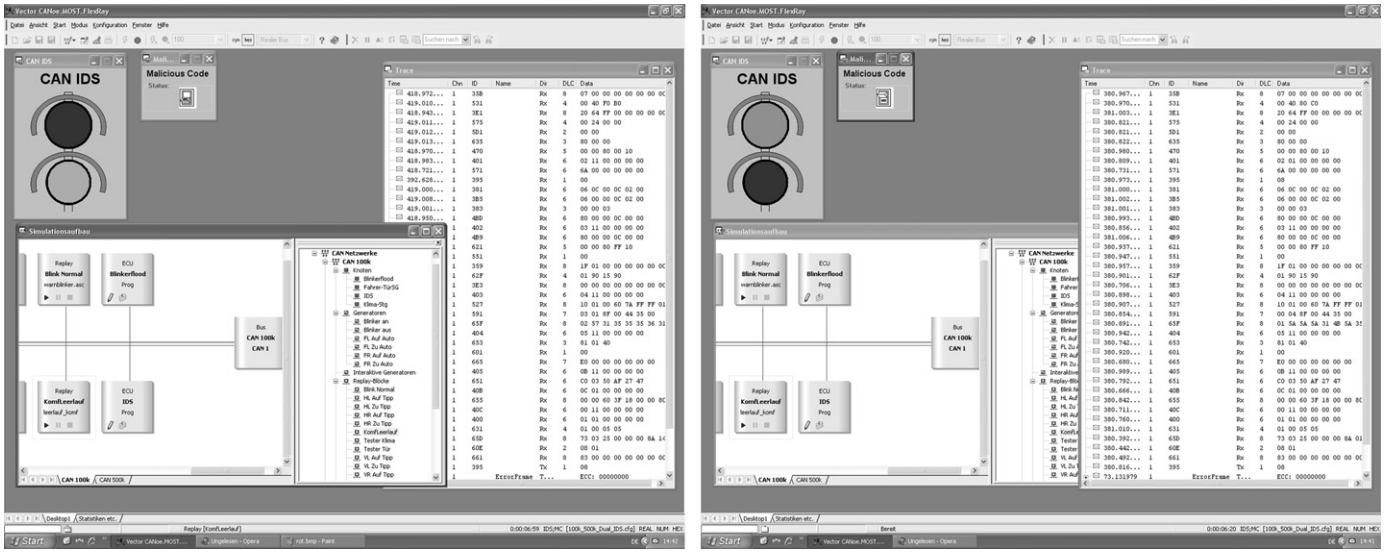

Fig. 6. Prototypical automotive IDS component.

managed to modify the original sending ECU directly, e.g. by injecting malicious code), their low-level characteristics do not differ. Another expected problem, which can be expected, might be that different ECUs can use the same CAN controllers (same manufacturer, same product line, etc.). Amongst these, the differences can be expected to be very subtile. So an interesting point of future research would be finding appropriate features with an adequate resolution also for these cases. A further challenge is that such a system can be expected to require a training phase in the final car to learn about the allowed environment. Also the problem of a legitimate swap of an ECU (e.g. due to component failure) would have to be addressed.

Further examples for general challenges in automotive intrusion detection functionality are how exactly to react after a detected incident (e.g. in terms of warning the occupants [18] or potential autonomous reaction) and how anomaly patterns or signatures could be maintained and updated efficiently (e.g. using future C2C/C2I technology).

4.2. Proactive forensics support

Assuming that IT security related attacks will increase in future, also post-incident inquiries on automotive systems can be expected to become more and more common (driven by police, insurance companies, etc.) and forensic investigations need to be conducted.

To support a forensic investigation the existing self-monitoring and diagnosis system applied in modern ECUs would have to be extended accordingly. Not only safety related events (more or less random component failures, blackouts and other malfunctions) would have to be logged but also additional information especially relevant for security related inquiries. This might contain information about flash operations (updated device, timestamp, source, etc.), systems being connected from the outside, power downtimes and many more.

In the following, a forensic model of the forensic process is described in Section 4.2.1 as well as two different approaches of establishing a proactive forensic capability (Section 4.2.2). Afterwards, the exemplary scenarios presented in Section 3 are addressed out of the perspective of IT-forensics by applying the introduced forensic model in Section 4.2.3. Finally, Section 4.2.4 discusses the restrictions and the potential for further research.

4.2.1. A forensic process model and its automotive application

In order to ensure that all relevant data are acquired, examined and analysed in a methodological way, a forensic model can be used, which covers the forensic process in a holistic way. Several of those models exist (e.g. see [22]); for the remainder of this article the model described in [23] was chosen because it does address not only *phases of the forensic process* but also *forensic data types*. In addition it also includes *classes of forensic methods*, which describe classes of software that has forensic capabilities (e.g. operating system, file system, applications, etc.). Using this model allows to identify forensically relevant data together with an appropiate acquisition and investigation mechanism whilst being independent of a particular software package. This classification therefore supports an adaption process to evolving forensic hard-/and software solution whilst retaining a focus on the forensically relevant data inside IT systems. In [19] it was adapted to the automotive domain. Classes of forensic methods address mainly the firmware of ECUs. The *phases of the forensic process* and the *forensic data types* will be introduced in the following in more detail.

The Phases of the forensic process

The grouping of investigation steps that belong together logically, according to [23], leads to six phases: strategic preparation, operational preparation, data gathering, data investigation, data analysis and final documentation. Those are shortly introduced in the following. All phases have in common that each step has to be documented extensively in terms of reasoning (e.g. why has a particular step been chosen), what forensic tools (e.g. HW/SW) were used and what individual results have been achieved.

An important aspect within a forensic investigation is the phase of *strategic preparation*. In this phase, all the **proactive measures** need to be put in place because they have the potential to substantially support or even enable a following forensic investigation. Since this (in [23] newly introduced) phase is so vital for the forensic process, all suggestions in the following sections and their application to the scenarios S1–S4 focus on this phase. In the automotive application, the manufacturer of the ECU should put mechanisms in place prior to a suspected incident, which are able to greatly support investigations (e.g. using extensive logging). This is somewhat analogous to the self-diagnosis but needs to ensure authenticity and integrity of the collected and stored data.

After an incident but before the initiation of the investigation, the phase of *operational preparation* has to take place. During that, all the options, tools and potential data sources have to be considered, as well as potential influences from some measures (e.g. when removing the power supply or cutting the bus wires some or all volatile data are lost). Also the provision of wiring schemes, means to access the MCU component inside the ECU and other related documentation, has to happen before any measures for data gathering are put into place.

In the *data gathering phase* all potential data sources containing potential evidence for an IT security related incident have to be collected. The results of the data gathering process using debug interfaces are memory dumps of the respective ECUs. Also the bus traffic should be recorded using bus sniffer modules if the vehicle still has its power supply and wiring (partially) functioning. To prove that content has not been altered, the usage of accepted cryptographic hashing mechanisms during the data gathering phase is necessary.

This phase is followed by the *data investigation phase*, in which the collected data are transformed into data that can be interpreted by the human investigator. In the case of the collected memory dumps this entails giving portions of the data a semantic meaning (e.g. locating and extracting diagnostic trouble codes (DTC)). The implementation of the self-diagnosis routines inside ECUs differs a lot between different manufacturers and different models. A lot of reverse engineering is necessary to be able to interpret the DTCs. One way to gain information as to how an ECU stores DTCs is to have a spare ECU and to attach and remove sensory input, comparing the resulting memory dumps. This way we found out that some DTCs are using a geographic encoding, which means if a component failure occurs, the location in the memory dump represents the type of DTC (which might indicate a defective component, e.g. temperature sensor). The numerical content at that location represents the number of occurrences and whether the component failure is permanent or transient. Not only the DTCs but also the programme containing the control logic in the ECU has to be investigated. This is useful to detect potential malicious or unauthorised alterations to this control logic.

After that, in the forensic process the *data analysis phase* follows. During that, the interpreted data are correlated and reduced. The correlation combines evidence from different ECUs and puts them into a timeline and a causal context. Often a lot of the data are unconnected to the incident that triggered the investigation (e.g. events that happened earlier). Therefore a reduction of data to that belonging to the investigation is necessary.

In the final *documentation phase*, the findings of the investigation have to be distilled into a report. Depending on the target audience, the technical content inside this report may vary, e.g. the report to an expert commission investigating the cause of an accident will include much more technical data than the report for a judge in a civil case.

The forensic data types

As stated in [23], eight different data types containing forensically relevant data types could be identified in the previous work. The modelling of the forensic data types stored and manipulated within a computer system represents a layered approach similar to the widely known OSI/ISO network layer model [21] with respect to describing layers of abstraction starting from low-level data characteristics successfully adapted to the automotive domain (see also [19]). The eight forensic data types are

- *Hardware data DT_1*. Hardware data are data in a system, which is not or only in a limited way influenced by the operating system and application. Examples are serial numbers of hardware devices, data contained in ROM, etc.
- *Raw data DT_2*. Raw data are a sequence of bits (or data streams) of components of the system not (yet) classified. In principle they can contain data of all the other data types. Examples of raw data are images of all types such as RAM dumps, firmware contained in Flash memory storage and also the raw data stream transported over the CAN bus.
- *Details about data DT_3*. Details about data constitute metadata added to user data. These can be stored within user data or externally. Details about data can be persistent or volatile. Examples are mileage counter readings or time information in DTCs. Volatile details about data are, for instance, the number of payload bytes in bus datagrammes.
- *Examples are Configuration data DT_4*. Configuration data are data that can be changed by the operating system or applications, which modify the behaviour of the system but not its behaviour with regards to communication. An example is the correction factor for the vehicle speed in relation to the wheel perimeter.
- *Communication protocol data DT_5*. Communication protocol data contain data, which modify the behaviour of the system with regards to communication. That includes, amongst network configuration data, also inter-process communication. An example would be the arbitration mask and therefore the priority of a CAN bus message.
- *Process data DT_6*. Process data are data about a running process. That includes, for example, the status of the process, the owner of the process, its priority, memory usage or the related application. Since multi-process software is not common in today's ECUs, they are very rare in automotive systems.
- *Session data DT_7*. Session data constitute data collected by a system during a session, regardless of whether the session was initiated by a user, an application or the operating system. A prominent example of session data is a DTC.
- *User data DT_8*. User data are contents edited or consumed by the user. This includes, for example, media data such as pictures, texts, audio or video data, for example used in infotainment systems.

These forensic data types can be used to determine, which data should be routinely collected or when certain predefined circumstances are met. The selection is a vital part of proactive forensic measures and has to take part in the phase of strategic preparation.

4.2.2. Centralised vs. distributed approaches to proactive measures in support of IT-forensics

Somewhat analogous to two approaches in the layout of Intrusion Detection Systems, also the proactive forensics support can be designed *centralised* (i.e. network-based) or *distributed* (i.e. host-based).

In the first proposal of a *centralised* system a dedicated supplemental unit would have to be installed into an existing automotive system and collect data, for instance, from the CAN bus network. It records a portion of network data from all the different bus systems inside a car (e.g. the powertrain bus or the instrumentation bus), if pre-selected conditions potentially constituting an IT security related incident are met. Because of the problems caused by the missing authenticity checks in today's bus systems, such a unit could ensure integrity and authenticity of the collected data only once it enters the unit.

In a *distributed* approach, each ECU's functionality would have to be extended with proactive forensic support, which goes a lot further than today's self-monitoring and diagnosis capabilities. It would require a lot more storage space for the recorded data and a substantial increase in processing power (e.g. for cryptographic means to ensure authenticity and integrity). The big advantage would be that additional sensory input only available at the particular ECU could be collected and IT security related incidents inside the ECU could potentially be recognised. Also if all ECUs had this forensic capability as described above, they would gather the same data as the centralised approach *plus* the local data that never leaves the ECU itself. Additionally, this approach would require less input from today's unreliable bus systems (with regards to authenticity and integrity of the data). However, it would mean a substantial and expensive redesign of every ECU's self-monitoring and diagnostic capabilities.

4.2.3. Proactive forensic measures—exemplary application to the attack scenarios S1–S4

With respect to the attack scenarios S1 and S2, the centralised approach would be sufficient to investigate these attacks since the relevant messages appearing at the internal CAN bus systems (addressing the electric window lift system and the vehicle electronics system, respectively) are suited as potential evidence. The forensic data types that are of primary concern are raw data DT_2 (i.e. the data stream that constitutes the affected bus datagrammes) and the communication protocol data DT_5 (i.e. the type of CAN bus messages). Session data DT_7 are not of concern since the event is unlikely to have triggered a DTC entry by the corresponding ECU.

The attack at the airbag control system in scenario S3 is very hard to prove using the centralised approach solely based on evidence from the bus systems (however, a detailed comparison of recorded diagnostics sessions with a reference-measure from a non-tampered system could reveal indications of the emulation). For the investigation of S3, the implementation of a distributed approach suits better, because this way the replacement of the original airbag control system by a bogus device can be expected to be easily provable. The main reason is that also its local forensic capabilities would have to be included in the emulation. Depending on the implementation, this can be expected to require secret information like cryptographic keys for the provision for data authenticity and integrity. The forensic data types that are of primary concern in S3 are raw data DT_2 (i.e. the data stream that constitutes the bus datagramme) and configuration data DT_4 (i.e. a disabled passenger seat airbag) as well as communication protocol data DT_5 (i.e. the type of CAN bus message) and session data DT_7 (i.e. status messages transmitted). The hardware data DT_1 are of no use because the relevant data

(i.e. part number, software revision, etc.) are likely to be spoofed to represent valid data.

In scenario S4 a CAN message spoofed towards the gateway ECU is used to gain unauthorised read access to internal CAN bus data via the diagnosis interface. Again, a centralised approach would be sufficient to investigate S4 incidents, because the malformed diagnostics requests could be revealed that are forwarded by the gateway into the internal bus networks. Also the unauthorised result of internal messages appearing at the outward diagnostics bus could be revealed if this is included in the monitored bus systems. The forensic data types that are of primary concern are raw data DT_2 (i.e. the data stream that constitutes the bus datagramme) and the communication protocol data DT_5 (i.e. the type of CAN bus message). Session data DT_7 are not of concern since the event is unlikely to have triggered a DTC entry by the corresponding ECU.

4.2.4. Discussion of restrictions and potential for future research

When restricting proactive forensic measures to already employed automotive technology by adding further components, the security aspects vital for a forensic investigation, namely integrity and authenticity, can only be assured after the data enters the added components. The main problems with today's automotive bus systems, like missing sender-/receiver authentication and lack of cryptographic means of ensuring data integrity, will substantially limit the potential of a forensic investigation into IT security relevant results.

A combination of both a centralised and a distributed approach (although being the most expensive) promises the best results as both local events and events transmitted over the subbus system could be taken into account (again, analogous to the combination of intrusion detection approaches).

In future it needs to further researched, which data and to what extent are forensically relevant and therefore what amount of storage capacity is necessary to hold that data. Also this information should be standardised, eradicating the need for reverse engineering especially during the data investigation phase. Furthermore, to employ practicable real-world solutions in future, these systems would need a secure basis, to especially provide integrity and authenticity for the communication and storage of data.

As main findings of this section, automotive adaptations of intrusion detection technology and proactive IT-forensic measures could already be considered as an extension of today's automotive IT technology, i.e. being short term measures with a reactive character. Regarding the prevention of such incidents the capabilities of more holistic concepts for a secure technical fundament in tomorrow's automotive IT are discussed in Section 5.

5. Holistic concepts for automotive IT security as long-term solutions, their potential and restrictions

In the long run, holistic security concepts for automotive systems are inevitable. Research about an appropriate basis for the implementation of such security measures has just started in the last few years (e.g. [24,8]). In this section, examples for relevant requirements are discussed and a short overview on selected, current research trends is given (Section 5.1). Reflections of the potential of such holistic security measures are made in Section 5.2, before this section closes with a discussion of remaining challenges and restrictions (Section 5.3).

5.1. Exemplary requirements and research trends

Looking at the special requirements of automotive systems and their role in everyday life yields a few important examples individual to this domain: Unlike home or office computer systems, cars are a kind of target frequently being physically exposed to different kinds of attackers (even the owner can be interpreted as an attacker if he tries to 'tune' or unlock some features in his home garage). This means, beneath a protection against software-based attacks like prevailing in desktop IT, the design of a holistic security concept for automotive IT systems should also put special focuses on hardware-related attacks. Another important factor is economy, i.e. the high cost restraints car manufacturers have to face. The components to establish a holistic automotive security platform have to be as cheap as possible.

Especially to guarantee aspects like authenticity or integrity, many current IT security measures are based on asymmetric cryptography, which is known to be computationally very expensive. To reduce computation and therefore hardware costs, alternative asymmetric algorithms like elliptic curve cryptography (ECC) are currently discussed [7], which are more efficient (compared to RSA, for example). An additional measure to address this is implementing these consuming algorithms in hardware.

To provide trustworthy computing platforms in the desktop IT domain, several international companies joined the Trusted Computing Group (TCG) [25]. The so-called Trusted Platform Modules (TPMs) developed by the TCG can already be found in many computers sold today and first security-related applications increasingly use the features of these hardware components. The potential of the underlying Trusted Computing (TC) technology for the protection of automotive IT systems is currently being researched (for example, see [26]). Due to the special requirements for the automotive domain (see above), current TPMs have been identified as inappropriate for the automotive application. Since current TPMs are separate chips being connected via bus systems, they are vulnerable to hardware attacks and are not suited for the automotive application with users not being trustworthy. Instead, one-chip solutions are being discussed combining CPU and TPM in a single, secured chip. To be as cost efficient as possible, it might only contain the least subset of TC functionality necessary for the automotive application.

5.2. The potential for future automotive IT applications

Once such a secure hardware basis will be available in future, the automotive applications will also need to use these newly provided functions in order to really tap the potential this new security basis offers. Consequently, within the entire automotive system the TC functionality would have to be employed to secure critical assets. So we expect a major redesign of automotive components and networks to be necessary in that stage. Looking at the results of our four practical attack scenarios in Section 3, the following example illustrates this: a car manufacturer might decide to utilise such an automotive Trusted Computing basis only in a small subset of devices, e.g. for securing software updates (flashing via OBD or update media) and tamper-protection for the anti-theft system as well as for selected sensitive information like the mileage counter. Consequently, this will not cover attacks from the bus level, if the communication between the ECUs (even if they internally use TC) will still use unsecured communication channels, like automotive bus systems established today. At least an additional security layer would be required on top that utilises the functions provided by the TC basis.

So holistic IT-based measures for automotive IT security need holistic utilisation in future, complex automotive IT systems (including their environment).

Applied to the relevant systems from the four attack scenarios S1–S4 (Section 3), a secure bus communication framework could be used for the prevention of respective attacks. For example, forged messages allegedly originating from the window switch console (S1) or the anti-theft system (S2) could be exposed, e.g. due to invalid digital certificates. An authentication system on component-level would be suitable to reveal a replaced airbag control unit in scenario S3. Assuming that a future gateway ECU implementation would be affected by a similar implementation flaw as in scenario S4, sniffed internal communication would be useless for the attacker, if properly encrypted. This way, an extensive utilisation of future secure bases for automotive IT security can help to effectively reduce the spectrum of potential attacks on automotive IT.

5.3. Remaining challenges and restrictions

However, some open questions remain, e.g. as how to keep the deployed crypto algorithms up to date to face the continuous improvements in cryptanalysis. Currently, the life cycle of cryptographic algorithms is significantly lower than the typical life time of current cars (which might easily be on the road for around 20 years). Hardware implementations of cryptographic algorithms (as discussed) are performing better and are cheaper than software implementations. On the other hand they are harder to maintain. Field-Programmable Gate Array (FPGA) chips might be a compromise to address this.

Consequently, also in tomorrow's automotive IT systems having such a holistic basis for security *protection*, flaws still allowing unauthorised access should be expected to arise at least temporarily (i.e. until a security-fix will have been deployed). The reactive concepts of adapted IDS and proactive forensics support could therefore also extend future automotive IT systems with such a holistic security basis (i.e. they could be used beyond an extension of today's technology that does not yet provide such a basis). Implemented on such a secure basis, they could themselves employ TC functionality to protect themselves against targeted manipulations (e.g. facing the intrusion detection functionality or of stored forensic evidence).

Besides the fact that every future automotive security solution will only be a compromise between the achievable security level and the resulting costs, it is important to be aware that even a technically perfect IT security solution (if actually possible) could not be expected to provide a full protection against all kinds of intended attacks. Especially respecting the human factor is a challenging task and a major restriction in many security concepts as already known from the desktop IT domain: Users tend to ignore warning messages and click them away if they bother them too frequently (e.g. whilst surfing through the web). Others enter sensitive information into forged phishing web pages because an authentic looking email advised them to do so.

To demonstrate the relevance of this fact also or especially within the automotive domain we close this section with a practical example for such "Social Engineering" attacks in the automotive domain. We prepared a multimedia disc containing MP3 music. An attacker might give or send this disc to his victim as a 'kind' gift, knowing that the victim might listen to it at his next car ride. The multimedia system, which is also part of our automotive test environment, plays the music and, for comfort reasons, always shows artist and title information about the current track (read from tag information contained) on its display using a large font. After a few regular songs, a specially prepared section might have been inserted by the attacker. In our tests we have split one song into short fragments and specified a seriously looking warning message as track information on every second fragment, while letting the entries in the other fragments (nearly) blank. When the player reaches this location during playback, it starts to display a flashing warning message (see Fig. 7).

Fig. 7. Exemplary low-tech attack on multimedia system interfaces.

This attack might even get extended by mixing a horrific warning signal into the sound material. Frightened by this situation, the driver might not realise the simplicity of this hoax and be seduced to follow such a malicious advice, and e.g. stop the car immediately—while the system actually still operates as designed.

Obviously, this attack does not need to break any technical security mechanisms in order to be effective. Beneath a secure technical platform, for a sound design of an automotive system in its entirety also non-technical aspects need to be addressed—like a very careful design of the user interfaces. For example, passing metadata of entertainment media (like MP3 tags) to the instrument cluster (which seems not to be supported in our test setup) would be even more critical. Where such arbitrary information is to be displayed, the designers should take great care to always emphasise the context of information being displayed. Although it consumes a bit more valuable display area, leading "artist:" or "title:" strings in the same font size, which are displayed by default, might be an appropriate measure to address this.

As main finding of this section, holistic concepts for automotive IT security will be inevitable in the long run. Future automotive systems will have to utilise these secure bases in a holistic way, to achieve an effective protection. However, due to various restrictions (having technical as well as non-technical reasons) even systems fulfilling these requirements might not to be considered as 100% secure. Therefore, also the extension with reactive security measures as discussed in Section 4 might be an appropriate choice.

6. Summary and outlook

With the focus on practical CAN based attacks on automotive IT systems, in this article we motivated the development of more efficient automotive IT security measures in the future. Based on the description of four practically implemented attack scenarios S1–S4, individual classifications of these incidents have been performed using the established CERT taxonomy and relevant examples for violations of the five main security aspects have been stressed out. Also, a special focus has been put to safety threats, which can potentially arise as implications of the security-based incidents depicted by these four scenarios. Based on the results of these tests, basic examples for the underlying security weaknesses in today's automotive communication networks have been identified. In consequence, future countermeasures have been discussed.

With respect to future, holistic concepts for automotive IT security an overview of main concepts has been given that are currently under research. We shortly introduced some examples for such holistic approaches that are currently developed by automotive IT security researchers and exemplarily discussed their advantages, potential and restrictions.

In this publication we focused on additional measures that could also be added to today's automotive IT systems, i.e. as short-term solutions addressing the most basic weaknesses that made our test results possible. We presented two exemplary approaches for such mechanisms and discussed their individual advantages, potential and restrictions. The first is the automotive application of intrusion detection technology, which is already well established in the desktop IT domain. A first, adapted implementation has been presented, which has already been tested exemplarily in practice on current automotive IT. The second concept is the adaptation of the IT-forensic process to the automotive domain with a special focus on proactive measures (i.e. prior suspected incidents). Both concepts have been discussed with respect to the four attack scenarios S1–S4.

However, both solutions from the previous paragraph are designed as reactive measures and do not directly prevent any kind of automotive, IT security related attack. Consequently, in the long run, holistic preventive solutions will be inevitable to increase the overall system security. They will serve as secure basis for further automotive IT security services, e.g. based on C2C communication. Also, the discussed reactive concepts intrusion detection and proactive forensics support, being promising extensions of future automotive IT systems as well, could profit from the additional security offered by such a holistic, secure basis.

As an important challenge for future research appropriate maintenance measures are needed covering the entire life cycle of modern cars. For preventive concepts this includes update facilities of cryptographic algorithms (usually having a significantly shorter life cycle). Also reactive measures like adapted intrusion detection concepts need appropriate maintenance concepts, e.g. to cover newly arising attack techniques (e.g. in the form of anomaly pattern updates or additional attack signatures).

Acknowledgments

The work described in this article has been supported in part by the European Commission through the EFRE Programme "Competence in Mobility" (COMO) under Contract No. C(2007)5254. The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The work in this article concerning IT-forensics has been supported in part by the Federal Office for Information Security (BSI).

References

[1] Eugene Kaspersky, Viruses coming aboard? Viruslist.com 〈http://www.viruslist.com/en/weblog?discuss=158190454&return=1〉; 2008 [accessed 12.07.10].
[2] Andrea Barisani, Daniele Bianco. Unusual car navigation tricks: injecting RDS-TMC traffic information signals. In: CanSecWest, Vancouver, 2007.
[3] Car-2-Car Communication Consortium, 〈http://www.car-2-car.org/〉; 2008.
[4] Andreas Lang, Jana Dittmann, Stefan Kiltz, Tobias Hoppe, Future perspectives: the car and its IP-address—a potential safety and security risk assessment. In: Computer safety, reliability, and security, Proceedings of the 26th international conference SAFECOMP 2007, Nuremberg, Germany, September 2007, vol. 4680. Springer LNCS. pp. 40–53. isbn 978-3-540-75100-7.
[5] BOSCH CAN, 〈http://www.can.bosch.com/〉; 2010.
[6] Howard, John D Longstaff, Thomas A, A common language for computer security incidents (SAND98-8667)/Sandia National Laboratories; 1998 (isbn 0-201-63346-9).
[7] Marko Wolf, André Weimerskirch, Thomas Wollinger, State of the art: embedding security in vehicles. EURASIP Journal on Embedded Systems 2007; (2007): 16 pages. Article ID 74706, doi:10.1155/2007/74706.
[8] Marco Wolf, Security engineering for vehicular IT systems—improving trustworthiness and dependability of automotive IT applications. View-eg+Teubner; 2009. isbn 978-3-834-80795-3.
[9] Press release of Ruhr-Universität Bochum. Remote keyless entry system for cars and buildings is hacked 〈http://www.crypto.rub.de/imperia/md/content/projects/keeloq/keeloq_en.pdf〉; 31 May 31 2008.
[10] HIS: Herstellerinitiative Software, 〈http://www.automotive-his.de/〉; 2010.
[11] Vector Informatik, 〈http://www.vector-informatik.com/〉; 2010.
[12] Tobias Hoppe, Jana Dittmann, Sniffing/replay attacks on CAN buses: a simulated attack on the electric window lift classified using an adapted CERT taxonomy. In: 2nd Workshop on embedded systems security (WESS'2007), a workshop of the IEEE/ACM EMSOFT'2007 and the Embedded Systems Week October 4; 2007.
[13] Tobias Hoppe, Stefan Kiltz, Andreas Lang, Jana Dittmann, Exemplary automotive attack scenarios: trojan horses for electronic throttle control system (ETC) and replay attacks on the power window system. In: Automotive Security—VDI-Berichte 2016, 23. VDI/VW Gemeinschaftstagung Automotive Security, Wolfsburg, Germany, 27–28 November 2007. VDI-Verlag; 2007. pp. 165–183. isbn 978-3-18-092016-0.
[14] Tobias Hoppe, Jana Dittmann, Vortäuschen von Komponentenfunktionalität im Automobil: Safety- und Komfort-Implikationen durch Security-Verletzungen am Beispiel des Airbags. In: Sicherheit 2008 "Sicherheit—Schutz und Zuverlässigkeit" Saarbrücken, Germany, April 2008; 2008. pp. 341–353. isbn 978-3-88579-222-2.
[15] Hoppe Tobias, Kiltz Stefan, Dittmann Jana. Security threats to automotive CAN networks—practical examples and selected short-term countermeasures. In: Harrison Michael D, Sujan Mark-Alexander, editors. Computer safety, reliability, and security, Proceedings of the 27th international conference SAFECOMP 2008, Newcastle, UK, September 2008, vol. 5219. Springer LNCS; 2008. pp. 235–48.
[16] Hoppe Tobias, Kiltz Stefan, Dittmann Jana. Automotive IT-Security as a challenge: basic attacks from the black box perspective on the example of privacy threats. In: Buth Bettina, Rabe Gerd, Seyfarth Till, editors. Computer safety, reliability, and security, Proceedings of the 28th international conference SAFECOMP 2009, Hamburg, Germany, September 2009, vol. 5775. Springer LNCS; 2009. pp. 145–58.
[17] FlexRay, The communication system for advanced automotive control applications. 〈http://www.flexray.com/〉; 2010.
[18] Hoppe Tobias, Kiltz Stefan, Dittmann Jana. Applying intrusion detection to automotive IT—early insights and remaining challenges. Journal of Information Assurance and Security (JIAS) 2009;4(6):226–35.
[19] Kiltz Stefan, Hildebrandt Mario, Dittmann Jana. Forensische Datenarten und—analysen in automotiven Systemen. In: Horster (Hrsg) Patrick, editor. DACH Security 2009; 2009.
[20] Stakhanova N, Basu S, Wong J. A taxonomy of intrusion response systems. International Journal of Information and Computer Security 2007;1(1):169–84.
[21] Zimmermann Hubert. OSI reference model—the ISO model of architecture for open systems interconnection. IEEE Transaction on Communications 2010.
[22] Casey Eoghan. Digital evidence and computer crime. Academic Press; 2004.
[23] Stefan Kiltz, Tobias Hoppe, Jana Dittmann, A new forensic model and its application to the collection, extraction and long term storage of screen content off a memory dump, In: 16th International conference on digital signal processing (DSP2009), IEEE Catalog Number CFP09452-CDR, 5–7 July 2009, Santorini/Greece. isbn 978-1-4244-3298-1.
[24] Jan Pelzl, Secure hardware in automotive applications. In: 5th Escar conference—embedded security in cars, November 6/7 2007, Munich, Germany.
[25] Trusted Computing Group, 〈https://www.trustedcomputinggroup.org/〉; 2010.
[26] Bogdanov A, Eisenbarth T, Wolf M, Wollinger T, Trusted computing for automotive systems. In: Automotive security—VDI-Berichte 2016, 23. VDI/VW Gemeinschaftstagung Automotive Security, Wolfsburg, Germany, 27–28, November 2007, VDI-Verlag; 2007. pp. 227–237. isbn 978-3-18-092016-0.