Security Threats to Automotive CAN Networks – Practical Examples and Selected Short-Term Countermeasures

Tobias Hoppe, Stefan Kiltz, and Jana Dittmann

Otto-von-Guericke University of Magdeburg
ITI Research Group on Multimedia and Security
Universitaetsplatz 2
39106 Magdeburg, Germany
{tobias.hoppe,stefan.kiltz,
jana.dittmann}@iti.cs.uni-magdeburg.de

Abstract. The IT security of automotive systems is an evolving area of research. To analyse the current situation we performed several practical tests on recent automotive technology, focusing on automotive systems based on CAN bus technology. With respect to the results of these tests, in this paper we discuss selected countermeasures to address the basic weaknesses exploited in our tests and also give a short outlook to requirements, potential and restrictions of future, holistic approaches.

Keywords: Automotive, IT-Security, Safety, Practical tests, Exemplary threats and countermeasures.

1 Introduction / 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 increasing information exchange between automotive systems and the outside world: For example, [2] demonstrated a technique to inject forged traffic information into navigation systems using the wireless protocols RDS (Radio Data System and TMC (Traffic Message Channel). And future technologies like car-to-car (C2C) [3] or car-to-infrastructure (C2I) communication are already discussed to implement several new automotive applications.

Looking at these trends and the high *safety* risks of such 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 disclose sensitive data but also directly endanger the safety of its human users (drivers, occupants) and environment [4].

In this paper we illustrate that already today the IT security of current automotive systems has to be addressed more forceful. We demonstrate this by summarising

M.D. Harrison and M.-A. Sujan (Eds.): SAFECOMP 2008, LNCS 5219, pp. 235–248, 2008.

results of several practical tests we performed on current automotive hardware based on the controller area network (CAN) bus system [5]. The basic weaknesses exploited in these tests are identified to discuss potential countermeasures for the future. Though suggestions for holistic approaches for long-term solutions are shortly introduced, we do focus on short-term countermeasures which address the basic weaknesses identified so far and might help achieve a reasonable security compromise until such a major redesign in future.

The paper is structured as follows: In the following section 2 we shortly present the state of art of automotive IT security measures, starting with existing applications. In the section 3 we describe our practical tests investigating attacks on exemplary automotive components, which have been partly extended for this publication and illustrate potential impacts to safety and comfort. They also serve as a basis for section 4 to identify what security aspects have been violated and which basic weaknesses have been exploited in these tests. In that section we also discuss potential countermeasures (some of which could already been demonstrated practically) as well as their potential and restrictions. The last section 5 concludes this paper with a summary and an outlook.

2 State of the Art

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. As state of the art, IT security mechanisms based on encryption or digital signatures can already be found in today's cars [6], but only in a very local scope protecting single components or functionalities:

Anti-theft systems like central locking or the immobiliser use 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 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 by [7]. It yields a manufacturer specific master key allowing an attacker access every car after sniffing two messages from a distance up to 300 ft.

Other potential attack targets 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, therefore the integrity of flash updates has to be ensured, too. In the context of the HIS ("Herstellerinitiative Software") group in Germany [8] several car

manufacturers joined and developed a common specification for secure flashing, which employs digital signatures as cryptographic 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 the following section 3 by presenting results from practical tests we performed in the past months.

3 Practical Demonstration of Exemplary Automotive IT Security Threats

Several practical test setups have been created to demonstrate IT security threats of current automotive technology, to analyse potential safety implications and to define and evaluate first countermeasures. In this section we summarise the basic principles and results of these tests to give an overview on our previous work. While most of these tests have been described in more detail in previous publications, we also have extended some of them recently to offer new results for this publication.

The tests were performed on a test setup consisting of real automotive hardware. It contains a wiring harness and different electronic control units (ECUs) of a recent model (built in 2004) of a big international car producer. Cars of this series use the CAN bus for the communication between the separate devices. Supported by different bus interfaces, a PC system can be used to investigate or interact with the automotive system. Fig. 1 illustrates this test setup.

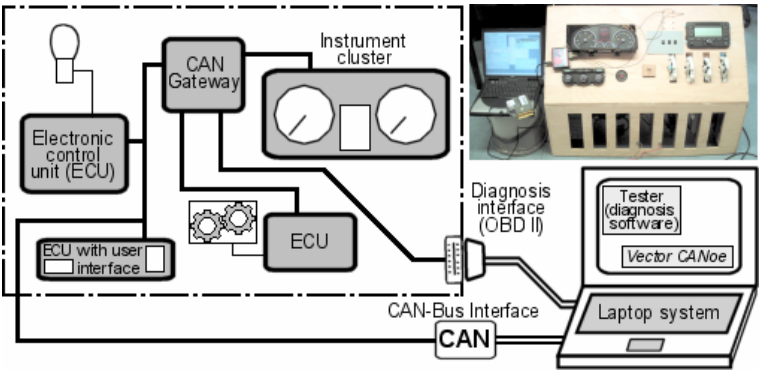

Fig. 1. Illustration of the practical test environment of automotive hardware

3.1 Analyses on the Electric Window Lift

The first potential attack target we investigated was the electric window lift. Early practical tests performed on this target were done within a simulation environment. For this purpose we used a simplified car environment which is part of CANoe, an established development and simulation software from Vector Informatik [9] widely used throughout the automotive industry.

In this test, a few lines of malicious code have been added to any ECU attached to the simulated Comfort CAN subnetwork. By waiting until some condition occurs (in this case when the car's speed exceeds 200 km/h) the code then 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 window opens and blocks until the driver reacts by pushing the "close" button. More details about this test can be found in [10] (as well as [4] and [11]).

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

After identifying the CAN messages relevant for triggering the window 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 practical test on current real automotive hardware constitutes a Denial-of-Service (DoS) attack on the window lifts (availability aspects).

The implications of a successful attack can affect both, *comfort* (the window cannot be moved any more) and *safety* (if the shocked driver loses control).

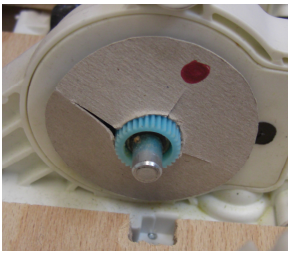

Fig. 2. Electric window lift (section 3.1), Indicator bulb, off (section 3.2)

3.2 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 an 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 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.

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 an additionally attached device like a developer's 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 [11] this attack proved to be quite powerful: The indicator bulbs (see right part of Fig. 2) stay completely dark most of the time,

while (apparently due to timing reasons) sometimes only a short, weak glowing appeared (though this is not expected to be noticeable through orange glass covers).

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

3.3 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 [12], the airbag control module was removed from the system. This might be done by an attacker to endanger the cars occupants (by the loss of a safety system), but much more common purposes are monetary interests. Unfortunately, as more and more police and press reports state, the theft of airbag systems is already quite common.

Within the attack examined, the attacker tries to suppress several signs of this removal which might sooner or later raise suspicion. One example clearly visible to the driver is the airbag warning lamp within the within the instrument cluster which indicates a failure (or absence) of the airbag control system. 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 [12] 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 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, but monitor other messages usually transmitted by the respective device – in this case by the airbag control module. In [12] we preliminarily addressed this problem by removing the airbag system from the gateway's device list. To the gateway it looks as if no airbag system was installed in the first place (which is an option in some countries), therefore no error condition is generated and the airbag warning light is not triggered. However, for an attacker this approach still had a few drawbacks. One is the removed device list entry which might raise attention when listing it during a diagnostics session. An attentive driver might also discover that, directly after entering ignition state, the airbag lamp does not show up shortly during the startup checks.

In additional tests we conducted for this publication, we could also practically demonstrate a more appropriate solution: we identified the relevant CAN message the gateway ECU expects from the airbag control module. This allows emulating also the

general communication of the airbag control system (beyond the diagnostics protocol already covered). By replaying this message 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 this message also contains a bit flag to set and unset the airbag lamp in the instrument cluster, also a successful startup check could be emulated this way by the malicious device.

While not reducing *comfort* (the driver will not notice any lack of functionality in regular operation) potential *safety* implications in emergency cases could be severe.

4 Analysis of the Underlying Problems; Capabilities and Restrictions of Potential Countermeasures

In this section we identify basic weaknesses in today's automotive systems that made the exemplary attacks in our practical tests possible. Based on this, potential countermeasures for future systems are discussed, some of which have already been tested in our test environment.

In the practical tests described in section 3, we accessed the car's IT infrastructure from within its internal bus systems. In the scope of this paper, 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 can be found where wires of the corresponding buses are located). But an attacker could also 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).

Consequently, also the internal communication of a car will have to be secured more in future. The following five central security aspects and privacy concerns known from IT security help to identify weaknesses in section 4.1 and discuss potential countermeasures afterwards:

- Confidentiality / Privacy
- Integrity
- Availability
- Authenticity
- Non-Repudiation

4.1 Analysis of Underlying Problems Relevant for the Exemplary Tests

The exemplary attacking strategies that we utilised in the practical tests primarily exploited drawbacks of the CAN bus protocol frequently employed in today's automobiles. 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 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 4, for none of them sufficient measures are provided at the CAN bus level, yet:

Confidentiality / Privacy: A message sent onto a CAN bus can at least be 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 is amongst these nodes and transmits the message into another subnetwork, even more nodes are affected. So in general, each of the receiving nodes can principally read the up to 8 bytes transported with each message. However, in some applications the transmitted information might be regarded 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. Encryption or anonymisation would reduce threats like these.

Integrity: With reference to the example given at the beginning of this subsection, 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 knowledge of a secret (private) key.

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 showed, malicious nodes can easily spoof messages usually sent by others. Receiving devices cannot detect that these come 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.

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. To ensure availability in the face of DoS-attacks is a difficult problem in general. The specification of the oncoming FlexRay bus system [13] 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 a *safety* measure against unintended malfunctions than to address *security* viewpoints.

Non-repudiation: After an incident like the spoofing attacks in our practical tests it is hard for the attacked devices to deliver proof of their innocence (i.e. that they did really receive such a malicious command or, respectively, 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.

In the following two subsections, 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.

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. In section 4.3 some current efforts of automotive IT security researchers are described.

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. In section 4.2 we therefore focus on discussing first concepts that might help to address basic weaknesses which made our practical tests succeed, which are mainly the missing *authenticity* measures in CAN communication.

4.2 Discussion of Short-Term Countermeasures to Address the Demonstrated Threats, Their Potential and Restrictions

To implement a minimal protection against basic attacking techniques like the ones presented in the practical tests, in this subsection we discuss two different approaches:

Approach a) 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 detect them. In the desktop IT domain such components are usually called Intrusion Detection Systems (IDS) [14]. 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 where 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.

With reference to the practical attacks investigated in section 3, we already identified several patterns which could be applied to detect such attacks. We shortly introduce these patterns below, one of which we have already tested in practice and discussed in more detail in the context of [15].

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 our examples this applies to the state of the window switches (first part of section 3.1) as well as to the message triggering the warning lights (section 3.2). 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 to achieve unauthorised actions by the receivers. However,

since removing existing messages is a lot 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. We could already demonstrate the effectiveness of this approach practically: in [15] we implemented this detection pattern for a prototypical IDS component and successfully tested it within our setup for the attack on the warning lights described in section 3.2.

Pattern 2: Obvious Misuse of Message-IDs

In the practical tests, unauthorised messages have been put on the bus by a device different 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 will also arrive at the original sending ECU. Currently, from the perspective of an attacker, this is no serious problem, because that device is not expected to evaluate this type of message, if this is usually only sent by 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. This 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, which 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 section 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 regarded information available from the data link layer (OSI level 2), 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 which 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 which 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 existing bus specifications).

Discussion of restrictions

However, with respect to the three patterns mentioned above, a few restrictions can be identified: As already mentioned, pattern 1 is only applicable to messages transmitted cyclically. It cannot be applied to message types that only appear occasionally (e.g.

which are only sent once as an indication of some event). Furthermore, pattern 1 and pattern 2 can obviously only be used to detect an incident, as long as the original sender is still present and functional. Pattern 3 is principally capable of compensating these restrictions of pattern 1 and 2. However, if malicious messages are sent by the same device (i.e. the attacker managed to modify the original sending ECU directly, e.g. by injecting malicious code), their low-level characteristics do not differ. Another expected problem might be that different ECUs can use the same CAN controllers (same manufacturer, same product line). Amongst these, the differences can be expected to be much smaller. So an interesting point of research would be finding appropriate features with an adequate resolution also for these cases. Also the problem of a legitimate swap of an ECU (e.g. due to component failure) would have to be addressed.

Approach b) Proactive Forensics Support

Assuming that IT security related attacks will increase in future, also post-incident inquiries on automotive systems might get more and more common (driven by police, insurance companies etc.). As the practical attack introduced in section 3.3 shows, finding a responding and faultless airbag control system during a diagnosis session is no reliable indication against a theft suspicion. Currently on the one hand diagnostics are only designed to detect unintended failures (safety violations) and are not secured against intended attacks (security violations). On the other hand, it would be too time consuming to dismantle a huge set of potentially affected cars to look for the physical presence of the components – and a clever attacker could have even placed dummies.

To also speed up the search for suspected security incidents, the diagnosis system 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. If present, also the intrusion detection components discussed above could notify the black box about suspicious events, e.g. to increase the logging intensity. To protect this sensitive data and avoid additional costs for the regular components, it could be stored in a single protected device like a black box and additionally get configured to be privacy preserving for the drivers.

When discussing this approach, also a few downsides of this approach have to be mentioned. Although memory devices are constantly getting cheaper and more powerful, especially the physical protection requirements would make such a black box relatively expensive without an obvious benefit to the customer. In the past, such a system for safety purposes (accident recorder) was already offered as option (e.g. [16] by an international car manufacturer. Although due to concerned customers it was made possible to erase the stored information at any time, they did not accept the system and it finally did not establish at the market in great numbers. So maybe privacy concerns were not addressed well enough in the system and its marketing. Another problem would be that malicious code, once present in the system, might try to flood the data recorder by spoofing useless information. This way an attacker might try to overwrite stored evidence or to hide them in a vast number of irrelevant entries.

4.3 The Need for Long-Term Solutions for Holistic Automotive IT Security Concepts, 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. [17]). This subsection gives a short overview on selected approaches currently discussed, their potential and remaining restrictions.

Looking at the special requirements of automotive systems and their role in every day life yields a few important requirements 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 restrictions 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, current IT security measures rely 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 [6], 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 in the Trusted Computing Group (TCG) [18]. 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 [19]). 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.

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. So we expect a major redesign of automotive components and networks to be necessary in that stage. With reference to the results of our practical tests in section 3, the following example illustrates this: A car manufacturer might decide to utilise such an automotive Trusted Computing basis only for securing the different kinds of software updates (flashing, update media etc.) and selected sensitive information like the mileage counter. Consequently, this will not cover attacks from the bus level, if the communication between single, protected ECUs 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).

Other remaining questions are 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.

Fig. 3. Exemplary low-tech attack on multimedia system interfaces

Besides the fact that every future automotive security solution will only be a compromise between the achievable security level and the resulting costs, the following last scenario demonstrates that even a technically perfect IT security solution (if actually possible) could not be expected to provide a full protection against intended attacks without respecting the human factor, as already known from the desktop IT security 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.

As an 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 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 (Fig. 3). 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 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) also 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.

5 Summary and Outlook

With the focus on CAN based attacks on automotive IT systems, in this paper we motivated the development of more efficient automotive IT security measures in the future. Based on the results from our practical tests, we identified basic weaknesses in today's automotive communication networks and discussed future countermeasures. In this publication we focused on short-term solutions addressing the most basic weaknesses that made our test results possible. We discussed a few exemplary approaches for such mechanisms (some of which we already tested in practice) with their individual advantages, potential and restrictions. In the long run, holistic long-term solutions will be inevitable. We shortly introduced some basic approaches that are currently discussed by automotive IT security researchers and also discussed exemplary advantages, potential and restrictions of these more holistic approaches.

Acknowledgments. The work described in this paper 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.

European Commission
European Regional Development Fund
INVESTING IN YOUR FUTURE

References

1. Kaspersky, E.: Viruses coming aboard?, Viruslist.com Weblog January 24, 2005 (June 2008), http://www.viruslist.com/en/weblog?discuss=158190454&return=1
2. Barisani,A., Daniele, B.: Unusual Car Navigation Tricks: Injecting RDS-TMC Traffic Information Signals. In: Can Sec West, Vancouver (2007)
3. Car-2-Car Communication Consortium (June 2008), http://www.car-2-car.org/
4. Lang, A., Dittmann, J., Kiltz, S., Hoppe, T.: Future Perspectives: The Car and its IP-Address - A Potential Safety and Security Risk Assessment. In: Saglietti, F., Oster, N. (eds.) SAFECOMP 2007. LNCS, vol. 4680. Springer, Heidelberg (2007)
5. BOSCH CAN, Website (June 2008), http://www.can.bosch.com/

6. Wolf, M., Weimerskirch, A., Wollinger, T.: State of the Art: Embedding Security in Vehicles. EURASIP Journal on Embedded Systems 2007, 16 (2007); Article ID 74706, 16 pages, 2007. doi:10.1155/2007/74706
7. Press release of Ruhr-Universität Bochum: Remote keyless entry system for cars and buildings is hacked, may 31st, Link (2008), http://www.crypto.rub.de/imperia/md/content/projects/keeloq/keeloq_en.pdf
8. HIS: Herstellerinitiative Software (June 2008), http://www.automotive-his.de/
9. Vector Informatik (June 2008), http://www.vector-informatik.com/
10. Hoppe, T., Dittmann, J.: 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)
11. Hoppe, T., Kiltz, S., Lang, A., Dittmann, J.: 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, pp. 165–183. VDI-Verlag (2007) ISBN 978-3-18-092016-0
12. Hoppe, T., Dittmann, J.: 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, pp. 341–353 (2008) ISBN 978-3-88579-222-2
13. FlexRay - The communication system for advanced automotive control applications (June 2008), http://www.flexray.com/
14. Stakhanova, N., Basu, S., Wong, J.: A Taxonomy of Intrusion Response Systems. nternational Journal of Information and Computer Security 1(1), 169–184 (2007)
15. Hoppe, T., Kiltz, S., Dittmann, J.: IDS als zukünftige Ergänzung automotiver IT-Sicherheit. In: DACH Security 2008, June 24-25, 2008, Technische Universität Berlin (to appear, 2008)
16. Website Kienzle-Automotive, product page of the Unfalldatenspeicher UDS system (June 2008), http://kienzle-automotive.com/index.php?108&tt_products=33
17. Jan Pelzl: Secure Hardware in Automotive Applications. In: 5th escar conference – Embedded Security in Cars, November 6./7, Munich, Germany (2007)
18. Trusted Computing Group (June 2008), https://www.trustedcomputinggroup.org/
19. 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, pp. 227-237, (2007) ISBN 978-3-18-092016-0