# Evaluation of Vehicle Diagnostics Security – Implementation of a Reproducible Security Access

Martin Ring, Tobias Rensen and Reiner Kriesten

University of Applied Sciences Karlsruhe
Karlsruhe, Germany
Emails: {rima0003, reto1014, krre001}@hs-karlsruhe.de

***Abstract*—Modern cars typically possess a network of numerous Electronic Control Units (ECUs) which are connected with each other by several bus systems. In addition to the necessary on-board communication by means of which the ECUs exchange information without any influence from outside, there is a strong need for interaction with off-board systems. In this context, the vehicle diagnostics can be mentioned as a significant example. It is highly important that the connection between diagnostic testers and the car is secured against unauthorized access. This paper examines the development of a procedure as well as a software tool for granting a reproducible access to individual car ECUs without any professional testers. If this access can be achieved by self-developed tools, a possible security danger exists as malicious diagnostic routines (not existing in professional car testers) can be activated by using this access. If the ways to achieve this access are known, it is possible to work on improving the defence.**

***Keywords–security access; safety; diagnostics security; data busses; communication standard.***

## I. Introduction

The increasing number of vehicle electronics [8] in modern cars leads to a permanently rising focus on safety and security aspects. Whereas safety can be described as the fact that the vehicle acts adequately in critical situations, security addresses the maturity of the car system against attacks from outside.

Concerning the safety issues, the International Standardization Organisation (ISO) has released the automotive specific standard ISO 26262 [17]. However, the standardization of security issues has not yet reached the same level.

Especially, the connectivity of modern cars to the outside world is a critical factor. Use cases like diagnostics exchange, navigation information, interaction with mobile devices and personalized services can be easily found. [3][4][5][12]

The easiest way to interact with the automotive network is via the On-Board-Diagnostics (OBD) connector. This connector serves as central access to all ECUs available in a car. For safety critical diagnostic functions, a so-called security access is implemented in the diagnostics standard [18].

We investigated if a self-written program can reliably achieve security access to modern vehicles by means of seed and key methods. Figure 4 describes the principles behind this practise. After a security request from the tester a random number, a so-called seed, is sent back from the vehicle ECU. Afterwards, the tester performs a secret coding algorithm and sends back the calculated key which is evaluated in the ECU [18]. The respective approach can be briefly described as follows:

- Recording of the security access between vehicles and testers in order to get the overall protocol sequence and information.
- Implementing of a software tool which replaces the car and requests keys from the tester in order to get the possible seed and key pairs.
- Testing the seed and key pairs for their reliable use. This implies in particular that they are independent of date, vehicle and ECU specific information like the Vehicle Identification Number (VIN).

Before the diagnostic data can be analysed, it is important to know how to interpret the payload in the CAN message, which is described in Section III. Section IV describes the fundamentals needed to simulate an ECU. The simulation of the ECU is described in Section V. Lastly, Section VI shows the analysis of the key exchange and which parameters are significant for its calculation.

ISBN: 978-1-61208-376-6

## II. Related Work

Only a small number of scientific writings are available on this subject. Especially, works focusing on a reliable procedure for gaining security access to the ECUs/network of an arbitrary car are rare. The related writings [3][4][5][12] mainly describe how to provoke a security hazard by means of additional components or a self-programmed code executed on existing components. This paper examines the possibility of provoking a hazardous situation by gaining access to needed software implementations, e.g., the ventilation of the Anti-lock Braking System (ABS) unit.

## III. Basics on automotive embedded systems

This section describes the fundamentals on embedded automotive systems needed for understanding this paper.

### A. Vehicle network: lower protocol layers

*1) Electric architecture:* Modern cars possess several bus systems for the communication between the ECUs, sensors and actuators. According to the AUTOSAR Standard [14], these devices are categorised in multiple networks, like body and comfort network,powertrain network or the infotainment network, seeFigure 1. The underlying bus system is further dependent on the necessary data rate, cost aspects, real-time-abilities, etc. However, the Controller Area Network (CAN) bus [16] is still the most popular bus in modern vehicles. As the diagnostics protocol usually is embedded in the CAN bus protocol, the latter is described more detailed in the following paragraph.

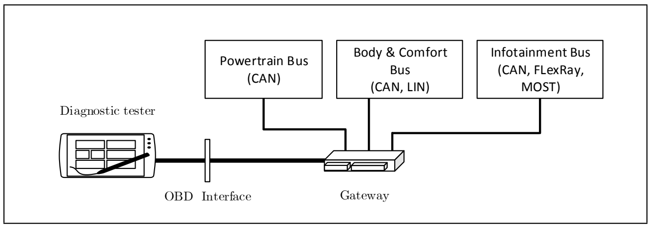


Figure 1. Vehicle network example.

*2) Information CAN bus:* The CAN bus is the most popular bus system in modern vehicles. In the U.S., it even is the standard for the OBD diagnostic since 2008. Regarding the physical characteristics, it uses a differential data transmission in order to resist electrical disturbances (to be seen as safety feature) and allows data rates up to 500 kbit/s [4].

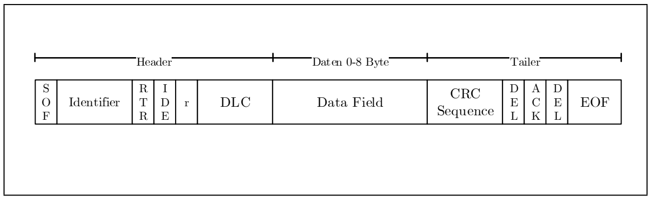


Figure 2. CAN packet structure [4].

Figure 2 shows the structure of a CAN message according to the standard ISO 11989. The most important parts of the message regarding diagnostic messages are the ID field containing the address of the ECU and the diagnostic payload located in the data field.

### B. Transport protocol

The transport protocol is standardized in the ISO 15765-2 [19] and is used for diagnostic purposes. This protocol is located one layer above the CAN protocol and allows upper services to transmit information with a data length of possibly more than 8 byte. The information of the Transport Protocol (TP) found in the most significant bytes of the CAN data field. These bytes are called Protocol Control Information (PCI). There are four different types of messages, the first nibble of the CAN data field contains the type information [5][11].

$0_h$ Single frame: contains the entire payload (less than 8 byte). The second nibble shows how much data the packet contains.

$1_h$ First frame: this is the first frame of a multi-packet payload. The next three nibbles contain the number of the whole diagnostic data.

$2_h$ Consecutive frame: this message contains the rest of the multi-packet payload. The second nibble contains the order of the sent message.

$3_h$ Flow control frame: this message is sent from the receiver of the multi-packet payload. This message is sent after the first frame [11].

### C. Vehicle networks: upper protocol layers

*1) Diagnostic protocol standards (Application Layer):* There are two popular diagnostic protocols: one is the Keyword Protocol (KWP) 2000 which is standardized in the ISO 9141 and ISO 14230; the other one is the Unified Diagnostic Services protocol (UDS) [18] which is standardized in the IS0 14229. The operation of both diagnostic protocols is almost identical. KWP 2000 was

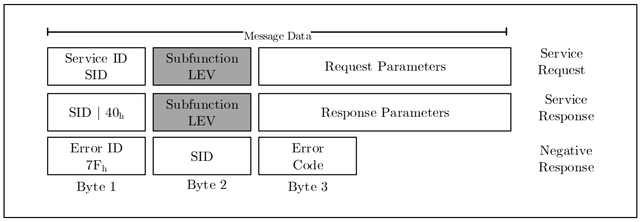


Figure 3. UDS diagnostic protocol [13].

originally designed for the proprietary bus system K-Line and is not used in modern cars anymore. Both protocols work with Service Identifiers (SID). Every SID represents a different action from an ECU which can be specified by its LEVs (subfunction levels); see Figure 3.

The provided services are defined in the standards. The services can be selected by the SID and LEV. These two bytes are the first two diagnostic data bytes of the message. There are three types of messages:

- The request message. This message is sent by the tester with the desired service.
- The response message. This message is sent from the ECU. The SID of the response message is calculated by logical or-linking the SID of the request message and $40_h$ (e.g., $27_h|40_h = 67_h$ ).
- The error message starts with $7F_h$, which is followed by the SID of the request and an error code with a length of one byte, as seen in Figure 3.

The control units communicate only after receiving a request from the diagnostic tester. There is a clear distribution of roles, in which the tester assumes the role of the client and the control unit works as server. This communication principle is also called request and response.

### D. Security Access in the diagnostic protocol

Today‘s security access is defined in the UDS standard. To access safety-critical-functions, the tester asks the ECU for a seed. After receiving this seed, the tester computes the according key, which is sent back to the ECU. If the received key is consistent with the expected key, access is granted [13]. Seed and key lengths, as well as the algorithm to compute the key, are not specified in the standard. Every vehicle manufacturer can implement an arbitrary seed length and algorithm. It is also not standardized if the seed is static or alternating. If the security access is used, the standard specifies that there are special LEVs to send the request for a seed and special LEVs for sending the key. All those subfunction levels can be found in the security access service (SID: $27_h$).

requestSeed: LEV $01_h$, $03_h$, $05_h$, $07_h - 5F_h$
sendKey: LEV $02_h$, $04_h$, $06_h$, $08_h - 60_h$ [19]

The process of the Security Access is shown in Figure 4.

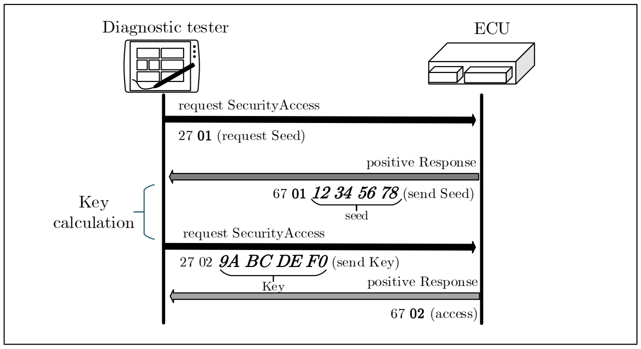


Figure 4. Security access timing sequence [11].

The message structure of the diagnostic messages from the tested vehicles follows the standardized protocols (with a few exceptions). The first byte of a single message contains the information about the transport protocol. In the message (listed below), the value is $02_h$. The zero (first nibble) stands for a single message and the two (second nibble) for two diagnostic data bytes. The second byte contains the SID and the third is the LEV (service and sub function).

Tester request data: **02 10 92** 00 00 00 00 00
ECU response data: **02 50 92** 38 37 30 32 39

## IV. TECHNICAL ACCESS SETUP FOR THE SECURITY EVALUATION

This section describes the physical setup in order to measure and record the diagnostic communications and the decoding strategy of the messages according to the given UDS standard.

In order to record the communication between the tester and individual vehicles, an additional client was added to the diagnostics line, a bus analysis tool running on the attached PC; see Figure 5 [15].

Thus, the existing communication between different cars and the tester could be easily recorded. In the second step the bus analysis tool was used for the simulation of

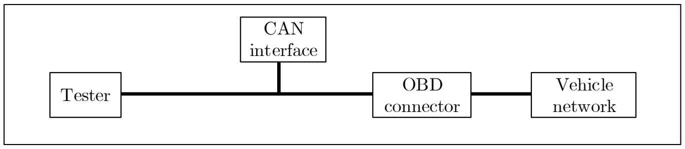


Figure 5. Recording strategy for diagnostics communication.

TABLE I. COMMUNICATION FROM BEGIN TO SECURITY ACCESS.

| CAN Data | description | send from |
|---|---|---|
| 02 **10 92** 00 00 00 00 00 | session request | tester |
| 02 **50 92** FF FF FF FF FF | session response | ECU |
| 02 **1A 87** 00 00 00 00 00 | session ECU info | tester |
| 10 16 **5A 87 01 22 05 14** | send ECU Info1 | ECU |
| 30 08 28 00 00 00 00 00 | send other parts | tester |
| 21 **FF 07 09 09 43 00 32** | send ECU Info2 | ECU |
| 22 **30 34 35 34 35 33 38** | send ECU Info3 | ECU |
| 23 **33 32** FF FF FF FF FF | send ECU Info4 | ECU |
| 02 3E 01 00 00 00 00 00 | tester present | tester |
| 02 7E 00 00 00 00 00 00 | tester present | ECU |
| 02 **27 01** 00 00 00 00 00 | Security req. | tester |
| 05 **67 01 F0 5E** 00 00 00 | send Seed | ECU |
| 04 **27 02 92 16** 00 00 00 | send Key | tester |
| 03 **67 02** 34 00 00 00 00 | pos. access | ECU |

the car. To be more precise, the bus analysis tool provides the messages which originally came from the real car; see Figure 6. It further has to be noticed that there is a reason for simulating the vehicle and not the tester; while having only a few attempts for the security access to car ECUs (afterwards, they deny any further access), professional testers can be stimulated an infinite number of times as in a typical environment they have to serve numerous vehicles and have to be permanently available.

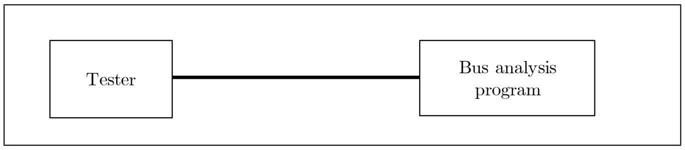


Figure 6. Simulation mode.

Table I shows an exemplary protocol sequence at the beginning of a security session. First, a handshake between the tester and the ECU is initiated by the tester including the exchange of specific ECU information. Afterwards, the seed and key messages appear for the authorization of the security access. In this context, it still has to be mentioned that most of the message data is standardized according to the UDS protocol.

### A. Vehicle selection

The choice of the investigated vehicles was influenced by the fact that since 2008 cars are offering the UDS protocol being typically embedded within the CAN bus. Considering this limiting conditions, six vehicles produced by four different manufacturers have been randomly chosen.

As a first result, it was not possible to perform a security access for one specific car platform as the corresponding services have not been implemented in the tester. In this case, only diagnostic routines which do not rely on the security access could be executed, e.g., reading/deleting error codes. Regarding all other tested car manufacturers, the security access could be recorded. To proceed, emphasis was put on two different cars of one manufacturer. The reason for this decision is mainly that this manufacturer implemented the security access according to the UDS standard. The security access was not implemented by all tested manufacturers, even though there is a standard [18] which recommends this access for certain safety critical functions. access to this vehicles was unlimited.

### B. Use cases for the execution of the security access

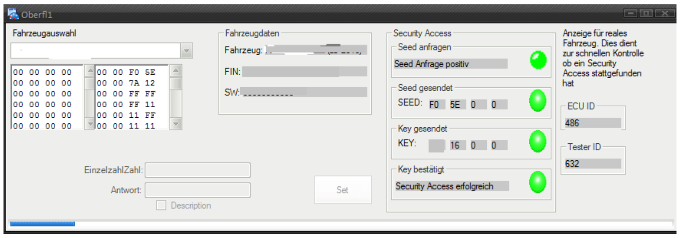
Table I displays the dial-up of the connection and the exchange of the seed and key data. Both the seed and the key are two bytes long which is car specific and not described in the standard. For both tested vehicles of this brand, the dial-up connection between the tester and the vehicle and also the security access are identical to the one shown in Table I, only the seeds, keys and ECU information differ. In the first vehicle, the security access appeared in the ABS ECU after selecting a specific safety function of this ECU. For non-safety-relevant diagnostic functions there was a request for the security access from the tester; see Table II. In contrast, the ECU obviously did not insist on the secure access, which affects the protocol sequence in the following way: the ECU sends zero information as key data (no security access needed) being also responded with zero bytes from the tester.

TABLE II. SECURITY ACCESS WITH ZERO BYTES.

| CAN Data | description | send from |
|---|---|---|
| 02 **27 01** 00 00 00 00 00 | Security req. | tester |
| 05 **67 01 00 00** 00 00 00 | send zeros | ECU |
| 04 **27 02 00 00** 00 00 00 | send zeros for key | Tester |
| 03 **67 02** 34 00 00 00 00 | pos. access | ECU |

## V. ECU SIMULATION FOR A REPRODUCIBLE SECURITY ACCESS

We implemented the communication behaviour of both ECUs (ABS / Airbag) existing in the different vehicles of which a security access was recorded; see Figure 5. The GUI of the simulation allows the selection of a car and the desired ECU. If a security access has been successfully performed the GUI displays a notification and the used seed and key data; see Figure 7. The seeds sent to the tester are arbitrarily chosen by the simulation program, so $2^{16} = 65536$ seed and key pairs exist, due to it‘s 16 bits length. Further, they can be written in a text-file before starting the simulation. After all seeds have been sent, the program generates a new file which stores the used seeds and its received keys. As already mentioned, the data exchange works only on request, which means that the whole simulation is controlled by the diagnostic tester.



Fahrzeugauswahl

```
00 00 00 00    00 00 F0 5E
00 00 00 00    00 00 7A 12
00 00 00 00    00 00 FF FF
00 00 00 00    00 00 FF 11
00 00 00 00    00 00 11 FF
00 00 00 00    00 00 11 11
```

EinzelzahlZahl:

Antwort:

□ Description

Fahrzeugdaten

Fahrzeug: [redacted]

FIN: [redacted]

SW: [redacted]

Security Access

Seed anfragen: Seed Anfrage positiv ●

Seed gesendet: SEED: F0 5E 0 0 ●

Key gesendet: KEY: 16 0 0 ●

Key bestätigt: Security Access erfolgreich ●

Anzeige für reales Fahrzeug. Dies dient zur schnellen Kontrolle ob ein Security Access stattgefunden hat

ECU ID: 486

Tester ID: 632

Figure 7. Panel for handling the ECUs and the security access.

## VI. SECURITY ACCESS ANALYSIS

In order to implement a tool which can reliably unlock different vehicles of the same model, it has to be analysed if the key algorithm is reproducible. This implies, in particular, the independence of the actual time and vehicle specific values such as the Vehicle Identification Number (VIN). In the following, the key algorithm is evaluated regarding its independence of date, VIN and ECU data.

### A. Data independence

The same seed was sent to the tester twice on different days. Each time the received key was identical. This shows that the key calculation is independent of date and time. Surely, this behaviour could be anticipated as it is unlikely that both vehicle and tester share the same timebase and use it for the seed/key calculation.

### B. VIN independence

In the tester, a VIN can be selected in order to determine the associated car. Therefore, one can assume that the seed and key data are dependent on the VIN. Thus, the traffic between the tester and the ECU was analysed and no VIN information was found. Furthermore, the tester was provided with two different VINs and access was requested using the same seed. As a result, the keys again did not differ. To conclude, the security access is independent of the VIN.

### C. Independence of ECU specific data

In order to assure that the key is only dependent on the given seed it is necessary to prove that the ECU specific information does not change the key data. Again, the simulation program twice requested keys while changing the ECU specific data; see Table III. Once more, the expected behaviour of independence could be confirmed.

TABLE III. CHANGED ECU INFORMATION.

| CAN Data | description | send from |
|---|---|---|
| 10 16 5A 87 01 22 05 14 | send ECU Info1 | ECU |
| 21 FF 07 09 09 43 00 32 | send ECU Info2 | ECU |
| 22 30 34 35 34 35 **33 38** | send ECU Info3 | ECU |
| 23 **33 32** FF FF FF FF FF | send ECU Info4 | ECU |

## VII. CONCLUSION AND FUTURE PROSPECTS

Evaluating the communication between modern vehicles and diagnostic testers enabled us to develop a software tool which grants security access to special electronic control units of modern vehicles. Using the developed software tool it was possible to extract the keys from the tested cars semi-automatically. As the respective process is not conducted fully automatically, the extraction of all keys for 16-bit seed and key pairs would take approximately 110 working hours. This workload could be reduced by an additional automation of the tester handling. It is also possible to generate a program which determines the possible algorithms of a given input and output vector. In a testrun, only 50 pairs were needed to determine the respective algorithm. The fact that it was possible to achieve security access can be considered as crucial because this access can be used to cause security critical and therefore dangerous conditions or unintended actions while the vehicle is in motion. Thus, it is recommended to improve the defence.

## REFERENCES

[1] K. Beiter, C. Rätz, and O. Garnatz, “Gesetzliche On-Board-Diagnose und ODX (Statutory On-board Diagnostics and ODX).” [Online]. Available: http://vector.com/portal/medien/diagnostics/odx/Gesetzliche_OnBoard_Diagnose_und_ODX.pdf-2014.07.21

[2] K. Borgeest, Elektronik in der Fahrzeugtechnik (Electronics in Vehicle Technology). Vieweg Verlag, 2007.

[3] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, “Comprehensive Experimental Analyses of Automotive Attack Surfaces.” [Online]. Available: http://www.autosec.org/pubs/cars-usenixsec2011.pdf-2014.07.21

[4] K. Koscher, A. Czeskis, F. Roesner, S. Patel, and T. Kohno, “Experimental Security Analysis of a Modern Automobile,” in 2010 IEEE Symposium on Security and Privacy, 2010. [Online]. Available: http://www.autosec.org/pubs/cars-oakland2010.pdf-2014.07.21

[5] C. Miller and C. Valasek, “Adventures in Automotive Networks and Control Units.” [Online]. Available: http://illmatics.com/car_hacking.pdf-2014.07.21

[6] T. Nosper, “Cotroller-Area-Network.” [Online]. Available: http://www.hs-weingarten.de/nosper/public/Download/Kapitel202.720CAN-Neues20Layout.pdf-2014.07.21

[7] K. Reif, Automobilelektronik (Automotive Electronics). Vieweg + Teubner Verlag, 2012.

[8] H. Richter, “Elektronik und Datenkommunikation im Automobil (Electronics and Data Communication in Automotive Applications),” Institut fr Informatik, Technische Universitt Clausthal, Tech. Rep. [Online]. Available: http://www.in.tu-clausthal.de/fileadmin/homes/techreports/ifi0905richter.pdf-2014.07.21

[9] F. Schäfer, OBD Fahrzeudiagnose in der Praxis (OBD Vehicle Diagnosis in practice). Franzis Verlag, 2012.

[10] T. Strang and M. Röckl, “Vehicle Networks CAN-based Higher Layer Protocols,” 2008. [Online]. Available: http://www.sti-innsbruck.at/sites/default/files/courses/fileadmin/documents/vn-ws0809/03-vn-CAN-HLP.pdf-2014.07.21

[11] J. Supke and W. Zimmermann, “Diagnosesysteme im Automobil (Diagnostic Systems in Automobiles).” [Online]. Available: http://www.emotive.de/documents/WebcastsProtected/Transport-Diagnoseprotokolle.pdf-2014.07.21

[12] M. Wolf, A. Weimerskirch, and T. Wollinger, “State of the Art: Embedding Security in Vehicles,” EURASIP Journal on Embedded Systems, April 2007. [Online]. Available: http://downloads.hindawi.com/journals/es/2007/074706.pdf-2014.07.21

[13] W. Zimmermann and R. Schnidgal, Bussysteme in der Fahrzeugtechnik (Bussystems in Automotive Engineering). Vieweg Verlag, 2007.

[14] Release 4.1 Overview and Revision History, AUTOSAR Std. [Online]. Available: http://www.autosar.org/fileadmin/files/releases/4-1/AUTOSAR_TR_ReleaseOverviewAndRevHistory.pdf-2014.07.21

[15] Handbuch CANoe (CANoe Manual), Vector Informatik GmbH.

[16] ISO 11898 CAN, ISO Std.

[17] ISO 26262 Safety, ISO Std.

[18] ISO 14229 Unified diagnostic services (UDS), ISO Std.

[19] ISO 15765-3 Implementation of Unified Diagnostic Services (UDS on CAN), ISO Std.

[20] CAPL Function Reference Manuel, Vector Informatik GmbH, November 2004.

[21] Programming with CAPL, Vector Informatik GmbH, Dezember 2004.

ISBN: 978-1-61208-376-6