Journal of Ubiquitous Systems and Pervasive Networks
Volume 10, No. 1 (2018) pp. 01-09

Signal-Layer Security and Trust-Boundary Identification based on Hardware-Software Interface Definition

Georg MACHER[a]*, Harald SPORER[b], Eugen BRENNER[c], Christian KREINER[c]

[a] *AVL List GmbH, Graz, AUSTRIA, 8010*
[b] *TÜV Austria, Vienna, AUSTRIA, 1010*
[c] *Graz University of Technology, Graz, AUSTRIA, 8010*

Abstract

An important trend in the automotive domain is to adapt established functional safety processes and methods for security engineering. Although functional safety and cyber-security engineering have a considerable overlap, the trend of adapting methods from one domain to the other is often challenged by non-domain experts. Just as safety became a critical part of the development in the late 20th century, modern vehicles are now required to become resilient against cyber-attacks. As vehicle providers gear up for this challenge, they can capitalize on experiences from many other domains, but must also face several unique challenges. Such as, that cyber-security engineering will now join reliability and safety as a cornerstone for success in the automotive industry and approaches need to be integrated into the mainly safety oriented development lifecycle of the domain. The recently released SAE J3061 guidebook for cyber-physical vehicle systems focus on designing cyber-security aware systems in close relation to the automotive safety standard ISO 26262. The key contribution of this paper is to analyse a method to identify attack vectors on complex automotive systems via signal interfaces and propose a security classification scheme and protection mechanisms on signal layer.

To that aim, the hardware-software interface (HSI), a central development artefact of the ISO 26262 functional safety development process, is used and extended to support the cyber-security engineering process and provide cyber-security countermeasures on signal layer.

Keywords: *ISO26262, SAE J3061, automotive systems, hardware-software interfaces, cyber-security, functional safety*

1. Introduction

In the late 1970s self-contained embedded systems called Electronic Control Units (ECUs) were introduced into production vehicles. Since then, the complexity of embedded systems in the automotive industry has grown significantly. Embedded automotive systems are estimated to account for 80% of product innovations in the past decade and are responsible for 25% of current vehicle costs [1]. These embedded systems are enablers for increasing the degree of digitalization, finally leading to an increase of competitiveness on existing markets as well as opening the door to new markets (e.g., data-driven business models). At the same time, the required dependability of these systems is raising: lack of safety, reliability, availability, integrity etc. of the system might lead to critical system failure having a severe impact on human health, environment, or property.

Exploiting the rising vehicle-to-vehicle and vehicle-to-infrastructure paradigms, future vehicles will have multiple inter-vehicle connections as well as capabilities for (wireless) networking with other vehicles and non-vehicle entities (such as charging stations and traffic lights) [2]. The resulting inter-connectivity increases attack surfaces and their damage potential.

Before the introduction of wireless connections and automated driving functionalities, vehicles were physically isolated machines with mechanical controls. Embedded automotive system technologies offered great benefits, but they also brought up new risks for the users safety. Therefore, functional safety engineering methods and processes become industry standard and critical part of the development.

In this context, the rising vehicle-to-vehicle and vehicle-to-infrastructure connectivity causes that automotive systems are developing from stand-alone systems towards systems of systems, interacting and coordinating with each other and influencing vehicle actions. Connections are thus not restricted to internal systems (e.g. steering, sensor, actuator, and

[*] Corresponding author. Tel.: +43-316-787-2974
E-mail: georg.macher@avl.com

DOI: 10.5383/JUSPN.10.01.001

communications) but also include other road users and the infrastructure and bringing up cyber-security issues. Consequently, new challenges regarding the manageability of systems are emerging caused by the increasing gap between cross-domain expertise required and the pervasiveness of novel technologies and software functions. An important trend in the automotive domain is to adapt established functional safety processes and methods for security engineering (e.g. the recently available SAE J3061 [3]).

In the course of this paper, we follow this trend and focus on signal-layer. Therefore, analyse a way to identify trust boundaries and attack vectors on complex systems via signal interfaces based on the hardware-software interface (HSI).

Although functional safety and cyber-security engineering have a considerable overlap regarding many facets, the identification of trust boundaries for the safety- or cyber-security-related aspects of complex automotive systems and the definition of system borders in ISO 26262 context completely differ. Here, appropriate systematic approaches to support this identification of trust boundaries are essential.

The HSI is a central development artefact of the ISO 26262 [4] functional safety development process. This artefact is the last development artefact of the system development and the starting point for parallel development of hardware and software. The HSI definition thus requires mutual domain knowledge of hardware and software and is usually not only consisting of signal interface information but several additional device configurations and linked constraints. In relation to this approach, we propose an extension for the HSI to support the cyber-security engineering process and also propose a security classification scheme and related protection mechanisms on signal layer.

The paper is organized as follows: Section 2 presents an overview of related works. In Section 3, a description of the proposed approach and detailed information about the individual items is provided. A brief evaluation of the approach is presented in Section 4. Finally, Section 5 concludes with an overview of the approach presented.

2. Related Work

The only currently available guideline for automotive cyber-security engineering, SAE J3061 [3] establishes a set of high-level guiding principles for cyber-security by: (a) defining a complete lifecycle process framework, (b) providing information on some common existing tools and methods, (c) supporting basic guiding principles on cyber-security, and (d) summarizing further standard development activities.

SAE J3061 states that cyber-security engineering requires an appropriate lifecycle process, which is defined analogous to the process framework described in ISO 26262 [4]. The guidebook recommends an initial assessment of potential threats (TARA - threat analysis and risk assessment) and an estimation of risks for systems that may be considered cyber-security relevant or are safety-related systems, to determine whether there are cyber-security threats that can potentially lead to safety violations. Apart from that, no further recommendations on how to proceed with this estimated risk, set-up a security classification scheme or give guidance for required protection mechanisms is given.

In the process implementation section (section 8 of SAE J3061), the details of the activities in each of the cybersecurity lifecycle phases are discussed. For each lifecycle phase, the activities as well as possible ways of their implementation are described. The first generation functional safety standards did not tackle the challenges of highly connected “systems-of-systems”. In particular, the arising security issues were not considered in the context of safety at this time.

Nevertheless, safety engineering approaches and development processes are already well-established in the automotive domain and therefore a way of integrating cybersecurity engineering in the existing domain-specific process landscape is strongly demanded in the automotive industry.

The unambiguous definition of the hardware-software interfaces (HSI) is vital in the context of the road vehicles – functional safety standard ISO 26262 [4]. Therefore, this development artefact seems to be the perfect starting point for identification of trust boundaries and attack vectors via signal interfaces. However, neither the current functional safety standard version nor automotive process reference model of Automotive SPICE [5] prescribe a specific methodology for the development of this artefact.

Also publications related to HSI definition in the automotive domain are rare. The most prominent definition of HSI is given by the functional safety standard ISO 26262 [4]. In this context, the HSI definition is one of the most important and essential work-products. The HSI document is the last development artefact of the system development phase and the starting point for parallel development of hardware and software. The majority of information concerning how to specify the interface in relation to functional safety can be found in Clause 7.4.6 of Part 4 of the standard. Additionally, the informative Annex B of Part 4 of ISO 26262 provides information concerning the possible content of the interface definition.

The Automotive Software Process Improvement and Capability dEtermination reference model [5] is based on the international standard ISO 15504 [6] and is primarily used in Europe, as well as in some parts of Eastern Asia. As mentioned, the model also does not address the demand for a hardware-software interface directly, but some hints on HSI specification can be extracted from general interface topics of the system engineering processes (SYS.3 and SYS.4) and software engineering processes (SWE.3 and SWE.5).

In [7] a model-based development (MBD) approach for an ISO 26262 aligned HSI definition is presented. This work combines spreadsheet tools (such as Excel) and MBD tools in a bidirectional manner to enable a tool-independent method of engineering HSI definitions with spreadsheet tools and transformation of the generated information into a reusable and version-able model representation. A domain-specific modelling approach for mechatronic systems with an integrated HSI definition feature is presented in [8]. The approach of this work has mainly been created for the development of embedded mechatronic based electric/electronic systems (E/E systems) in the automotive field and is based on a domain-specific language tailored for the specific needs of domain experts. The focus of this work was particularly set to simplify the work of domain experts who disfavour system modelling approaches (like UML or SysML).

In our work [9] we described the linking of such a HSI based attack vectors identification in relation to the afore mentioned Automotive SPICE reference model and also in relation to the ISO 26262 - road vehicle functional safety standard. Other works postulate the problematic of defining HW/SW interfaces are part of an emerging domain-independent paradigm for contract-based design. The contracts specify the input and output behaviour of a component and provide a guaranteed behaviour [10]. Such an approach can be used for software component safety contracts [11] as well as contract-based embedded system development [12, 13].

Nevertheless, these approaches are not yet very common in the automotive domain.

Contract-based design paradigms are an emerging domain-independent paradigm for interface definition. The contracts specify the input and output behaviour of a component and provide a guaranteed behaviour [10]. Such an approach can be used for software component safety contracts [11] as well as contract-based embedded system development [12].

According to Avizienis et al. [23] dependability is a superordinate concept regrouping different system attributes such as reliability, safety, security, or availability and non-functional requirements for modern embedded systems. These different attributes, however, may be contradicting and thus might lead to different development targets. Moreover, the non-unified methods to manage these different attributes might lead to inconsistencies identified in late development phases only. Fig. 1 provides an overview of the attributes (aspects) of dependability, the analysis methods available for the automotive domain for the different attributes as suggested in [24], and a common dependable development block indicating the fact that each aspect needs to be addressed within a consistent engineering framework.

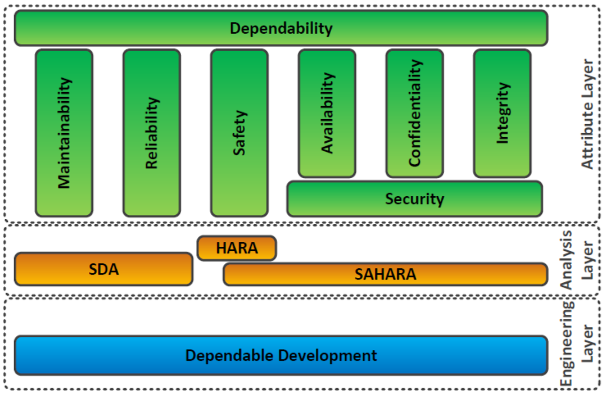

Figure 1 Overview of Dependability Attributes and Analysis Methods suggested in [24]

Systems dependability features are thus challenging research domains, which require a continuous development process and mutual domain expertise. Dependable systems rely on mature quality management and development methods such as requirements / systems engineering, system analyses (e.g., FMEA), design and validation plans.

Since all dependability attributes need to be supported by the same system architecture, they require a common engineering basis and a mutual understanding of focuses, language concepts and mutual dependencies. Table 1 shows a mapping of safety and security oriented engineering terms regarding the initial analysis step, which is the HARA and TARA [15].

Most modern embedded automotive systems are representative of safety and security relevant use-cases, due to their usually safety and security related nature and complexity regarding elements of the system, underlying functionality and the related external systems.

Table 1 Mapping of Safety and Security Oriented Engineering Terms.

		Safety Engineering	*Cybersecurity Engineering*
Analysis subject	**Risk**	**Hazard**	**Threat**
	System inherent deficiency	**Malfunction**	**Vulnerability**
	External enabling condition	**Hazardous situation**	**Attack**
Analysis Category	**Impact analysis**	**Severity**	**Threat criticality***
	External risk control analysis	**Controllability**	**Attacker skills*, know-how**
	Occurrence analysis	**Exposure**	**attack resources*, attack surfaces**
Analysis results	**Design goal**	**Safety goal**	**Security target**
	Design goal criticality	**ASIL**	**SecL***

3. Hardware-Software Interface Definition with Security Extension

In the context of the road vehicles - functional safety standard ISO 26262 [4] the HSI definition is probably the most crucial and essential work-product. The HSI specification no longer consists of only a single spreadsheet description of all signals between hardware and software, but also consists of supplementary information, such as resource consumption objectives, HW specifics, and controller module configurations. Establishment of the HSI requires mutual knowledge of hardware and software components and is usually the result of a collective workshop of hardware, software, and system experts.

Most of the best practices for security-by-design suggested by [22] are covered by applying the process landscape of ISO 26262 and SAE J3061. The most crucial best practices that are not included are (a) the identification of trust boundaries and (b) the establishing of a layered defense (defense-in-depth) approach. Here the HSI definition can provide means for identification of trust boundaries and a layered defense.

Trust boundaries for the safety-related aspects of complex automotive systems are determined by a function-oriented definition of system borders where dangerous malfunctions are controlled. This is called “item definition” in the ISO 26262 terminology. In cybersecurity, by contrast, trust boundaries are used to describe a boundary where program execution or data protection change their levels of "trust". The term refers to any distinct boundary within which a system trusts all other sub-systems that are within this boundary. Trust boundaries can be related to privileges, integrity, control units or communication networks, and can also refer to points or attack surfaces where attackers can intervene. In order to clearly distinguish (sub-) system boundaries, the term “Feature definition” is used for the cybersecurity aspects of a product [3, Sect. 8.3.1], while “item definition” is used for functional safety aspects.

Appropriate systematic approaches to supporting the identification of trust boundaries are essential. In this work we propose a way to identify trust boundaries and attack vectors on complex systems via signal interfaces based on the hardware-software interface (HSI). This follows the concept of a layered cybersecurity defense approach mentioned before.

Table 2 itemizes essential HSI attributes extracted from standards (ISO 26262 [4], AutomotiveSPICE [5], and SAE J3061 [3]), scientific papers like [8, 14], and the authors' experiences [9]. Security related information have been added (marked in coloured text) to support identification of attack vectors via their signal interfaces of the systems. To bridge the shortcoming of SAEJ3061 of not providing any guidance on how to proceed with the cyber-security metric estimated by TARA, we propose to entail the estimated security level (here named SecL) of the TARA to the related system and its interfaces. To that aim, cyber-security relevant signals inherit their security level from the threat analysis and risk assessment (TARA; requested by SAE J3061 [3]) of the system they belong to. Depending on the related security level, the signal shall be protected against cyber-security attacks. Such an approach is also proposed for automotive systems in general by [16, 17] and for in-vehicle infotainment systems in particular by [18]. The identification of trust boundaries and gateways which protect the boundaries is both crucial and cumbersome for complex system and network structures. Here, enhancing the HSI definition with supplementary cyber-security information and related signals helps to determine trust boundaries and attack vectors by focusing on signals and thus identifying controllers that can intervene with the involved signals. All controllers that can directly intervene the involved signals are identified by analysing the signal interfaces of a specific system. These control units are within the same trust boundary and for this reason equally trusted. An access to the trust boundary is only possible via devices with connections outside the trust boundaries.

Table 2 Essential HSI Attributes, Comments and Origin

Layer	*Attribute*	*Comment*	*Origin*
conceptual	**signal name**	**significant name**	**[14]**
	signal description	**short signal description**	**ISO 26262 Part 6**
	signal direction	**input or output**	**[14]**
	signal source/sink	**actuator or sensor related to signal**	**[8]**
	ASIL	**Automotive Safety Integrity Level**	**ISO 26262 Part 4**
	Security Level (SecL)	**security metric**	**SAE J3061 and [15]**
physical	**supply voltage**	**-**	**[8]**
	physical min value	**-**	**ASPICE**
	physical max value	**-**	**ASPICE**
	physical unit	**-**	**ISO 26262 Part 6, ASPICE**
	accuracy	**% range of value**	**ISO 26262 Part 6**
	HW interface type	**digital, analogue, bus ...**	**ISO 26262 Part 6, ASPICE**
	HW pin	**pin number or identifier**	**ISO 26262 Part 5**
data	**message ID**	**for bus communications**	**[8]**
	message offset	**for bus communications**	**[8]**
	cycle time internal	**xCU internal refresh rate**	**ISO 26262 Part 6, ASPICE**
	cycle time external	**cycle time of digital signal from external**	**[8]**
	trigger	**identifier of trigger**	**ISO 26262 Part 6**
	operation mode	**information if any special operation mode required**	**ISO 26262**
	HW diagnostic feature	**diagnostic feature description**	**ISO 26262**
	memory type	**-**	**ISO26262**
	data protection	**special security information**	**ASPICE**
	timing dependencies and sequence order	**-**	**ASPICE**
presentation	**SW signal name**	**signal identifier for ASW**	**[14]**
	initial value	**-**	**[8]**
	SW data type	**-**	**ASPICE**
	scaling LSB	**fixed-point arithmetic scaling**	**[14]**
	scaling offset	**fixed-point arithmetic scaling**	**[14]**
	SW min value	**-**	**ASPICE**
	SW max value	**-**	**ASPICE**
	SW accuracy	**% range of value**	**ISO 26262**
	SW unit	**physical unit representation**	**ASPICE**
	default value	**default value in case of invalid signal**	**[14]**
	detection time	**time to fault diagnosis**	**ISO 26262**
	reaction time	**reaction time after fault detection**	**ISO 26262**

These devices are referred to as gateways and are responsible to prevent from attacks and the misuse of trust of the control units within a trust boundary. Thus, starting with the signals from the HSI definition enables a structured and methodical approach for the identification of trust boundaries and gateways.

The realization of a secure context in vehicle systems then requires the coordinated application of different security technology best practices. Nevertheless, currently no standardization for the coordination of security technology practices has been established and it is up to the manufacturers to decide how to provide a secure context. Therefore, we propose the design guidelines for signal security for the different security levels (based on [19]) summarized in Table 3.

As the table depicts, for different security levels different security technology practices shall be applied. This means, that for non-security-relevant signals (*SecL = 0*) no additional requirements are stated. For signals identified as security-related, different security technologies shall be applied. In such a way that for *SecL = 1* the origin and integrity of the messages shall be verified, while for *SecL = 2* additionally to these *SecL = 1* measures, also the volumes of messages shall be checked, abnormal behaviour and intrusion shall be detected, and immutable device identification must be ensured. Therefore, each security level must also imply the security technology practices assigned to the lower levels.

Table 3 Design Guidelines for Signal Security

Layer	*Attribute*
SecL = 0	no additional requirements
SecL = 1	verify origin of message verify integrity of message
SecL = 2	check volumes of messages detect abnormal behaviour immutable device identification intrusion detection
SecL = 3	encrypted communication data encryption
SecL = 4	establishing of private communication channel correct cycle detection blocking of unapproved and inappropriate messages

This HSI based identification of trust boundaries based on signal flows supports the following SAE J3061 sections:

- 5.5 Implement Cybersecurity in Development & Validation
- 8.4 Product Development: System Level
- 8.4.3 Refine Functional Cybersecurity Concept into Technical Cybersecurity Concept
- 8.4.4 Specify Technical Cybersecurity Requirements

Since the emergence of connectivity features, however, security has become increasingly important for and has received significant attention by the automotive domain. Therefore, it is apparent that security defects share a significant number of common themes and patterns, making it possible to define and categorize them [25]. Security flaws can be found both at the code level and at the design level. The top ten automotive cybersecurity vulnerabilities account for 87.5 percent of all reported vulnerabilities in 2015.

4. Application of the Proposed Approach

This section demonstrates the application of the proposed approach based on an automotive use-case example of a battery management system (BMS) for electrified hybrid powertrains. Electrified hybrid powertrains (a combination of one or more electric motor(s) and a conventional internal combustion engine) are currently the most common variant of hybrid powertrains. The variety of powertrain configuration options increases the complexity of the powertrain itself as well as the required control systems (software functions and control units).

Several different types of energy sources can only be utilized perfectly if the control systems are properly designed and perfectly configured. To that aim connectivity features and external real-time data are more and more integrated into control strategy decisions.

The full HV battery system consists of the BMS, the battery satellite modules (grouping battery cells in modules and communicating via dedicated bus), and a fan control for cooling of the battery cells. This system is connected to various powertrain control units, the charging interface (enabling the communication with battery charging stations), the on-board diagnostic interface (OBD), and via a dedicated gateway to the vehicle infotainment systems (including the driver interface (HMI) and also a wireless internet connection).

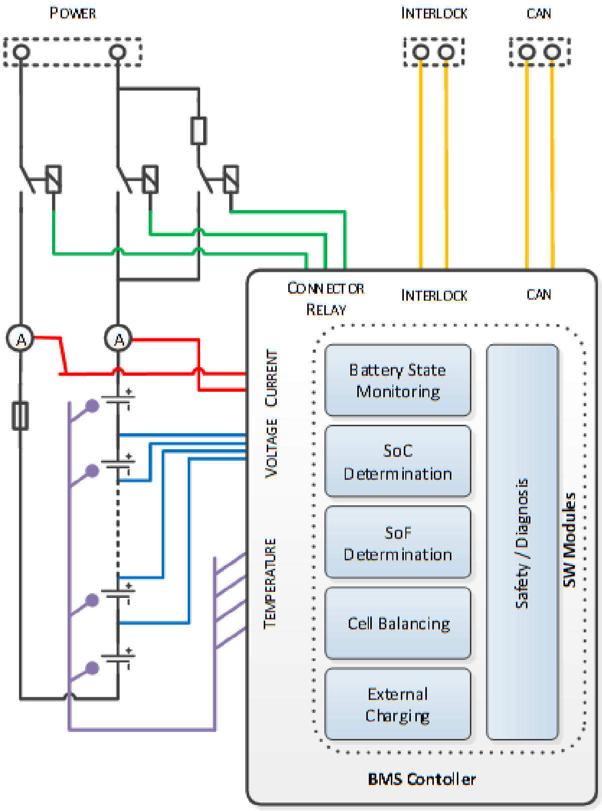

Figure 2 Depiction of the BMS in Context of ISO 26262 Item Definition

For the initial assessment of cyber-security threats and safety hazards the SAHARA method [15] can be applied. This method identifies safety goals (SG) and cyber-security attacks which can violate the respective SG. Additionally, the SAHARA method assigns a quality label for the safety impact (ASIL; standardized by ISO 26262 [4]) and cyber-security factor (SecL; proposed by [15]) of the system. These labels are further inherited by components and sub-systems of this system (e.g. also including the signals required for the systems operation).

Table 4 Excerpt of the HSI Definition of the BMS Use-case

HSI		BMS					
signal name		common	throttle position	vehicle speed	vehicle state	battery voltage	battery current 1
signal description			actual throttle position	actual vehicle speed	actual vehicle state	actual battery voltage	actual battery current (sensor 1)
sensor/actuator			VCU	VCU	VCU	BMS	I_sens1
direction			in	in	in	in	in
ASIL			ASIL B(D)	ASIL B(D)	ASIL D	ASIL D	ASIL B(D)
SecL			2	2	2	0	0
source(CAN/ANA/DIG)			CAN	CAN	CAN	ANA	ANA
physical unit		physical	--	--	--	V	A
physical range lower limit			--	--	--	10	5
physical range upper limit			--	--	--	500	200
supply voltage			--	--	--	--	--
signal tolerance	%		1	1	--	5	5
interface			CAN A	CAN A	CAN A	analog	analog
pin			Port B12	Port B12	Port B12	Port B33	Port A11
refresh rate	ms	data	10	100	100	10	10
cycle time	ms		1	10	10	10	10
message ID			0x185	0x188	0x198	--	--
message offset			0	0	0	--	--
trigger						timer	timer
operation mode			normal	normal	normal	normal	normal
HW diagnostic			CRC	CRC	CRC	voltage range	redundancy
register-type			RAM	RAM	RAM	RAM	RAM
data protection			--	--	--	--	--
dependency			--	--	--	--	--
signal type (V / % / deg)		SW	%	kmph	--	V	A
variable name			VCU_THrPos_Pctg	VCU_VehSpd_kmph	VehState_Ctl	BMS_BatActUSens	BMS_BatActISens1
initial value			0	0	--	0	0
signal range lower limit			0	0	--	10	0
signal range upper limit			100	400	--	450	200
Scaling LSB				163	--	1	1
Scaling Offset				35	--	0	0
accuracy			0,5	0,5	--	1	1
default value			0	0	--	0	0
type			uint8	uint16	uint8	uint16	uint16
detection time	ms		50	200	200	50	50

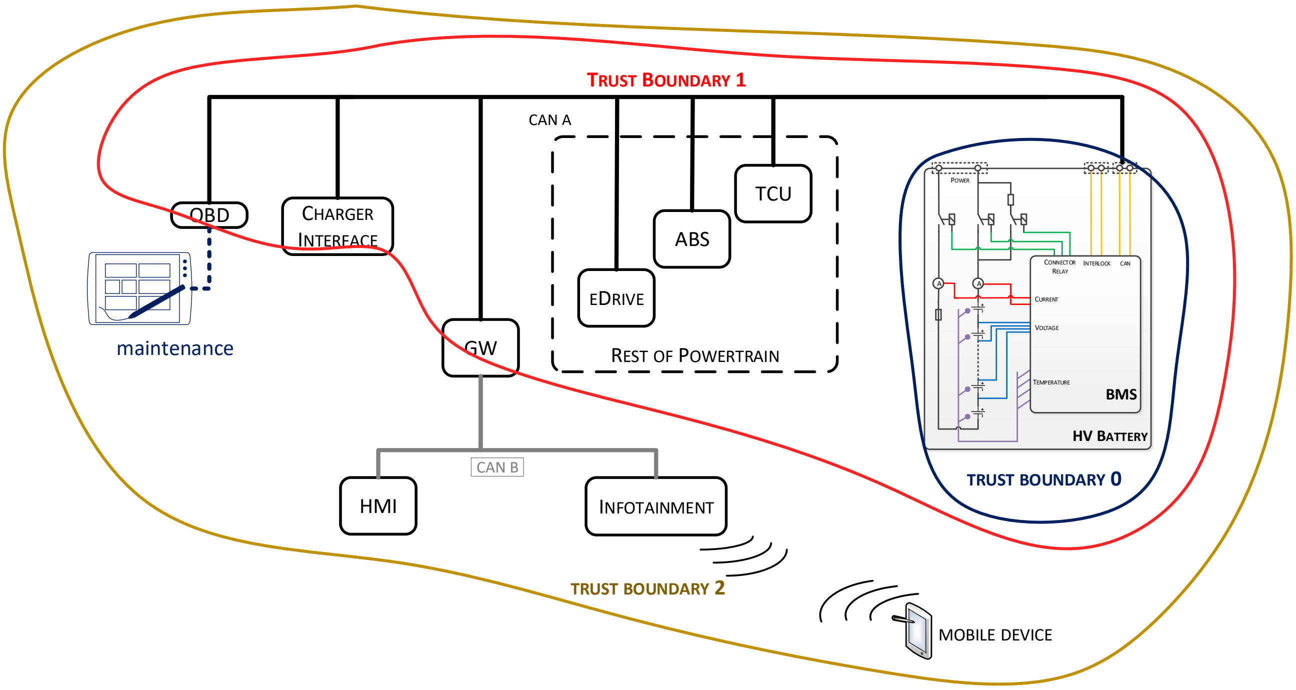

Figure 3 Trust-boundary Layers of Use-Case

Figure 2 depicts the conceptual building blocks of the complete HV battery from safety point of view (always implying the actuation chain from sensors via controller to the actuators). This is called “item definition” in the ISO 26262 terminology.

These main building blocks of the BMS are:

- Power contactors - connection with HV system
- Interlock - de-energizing HV system when tripped
- CAN - automotive communication interface
- Relay - main contactor and output unit of the BMS
- Temperature sensors - feedback of actual cell temp
- Voltage sensors - feedback of actual cell voltages
- Current sensors - feedback of actual current flow
- Fuse - protective circuit breaker in case of fault
- Cells - electro-chemical energy storage
- BMS controller - monitoring and control unit

Trust boundaries for the safety-related aspects of complex automotive systems are determined by a function-oriented definition of system borders where dangerous malfunctions are controlled. In cyber-security, by contrast, trust boundaries are used to describe a boundary where program execution or data protection change their levels of "trust". Trust boundaries can refer to points or attack surfaces where attackers can intervene. Thus, appropriate systematic approaches supporting the identification of trust boundaries are essential.

Based on the HSI identification of the interfaces providing the signals (see Table 3 - line 14), devices connected to this interface can be easily identified and trust-boundaries for the specific system identified. This enables a complete identification of involved controllers for a further analysis of interfaces and the establishment of barriers for cyber-security attacks. To do this, we started with the ISO 26262 item of the BMS (Figure 2) and filtered the content of the HSI for signals related to the BMS (Table 3). The first step of the approach identifies controllers which have access to the signals related to the BMS based on the HSI definition. These controllers either generate the signals directly or are connected to the same communication bus. The second step identifies the inner trust boundary 0, which includes signals directly connected to the BMS and simultaneously the gateways to the trust boundary (CAN connection of the BMS). These two steps are repeated for the remaining signals to establish further trust boundaries, as depicted in Figure 3.

This determination of trust boundaries and especially constraints of the security attributes of a signal ensures also a more consistent integration of the dedicated security measures, since also constraints for the signal supplier and interaction with other systems are highlighted. As an example, information exposure issues make up 12.8 percent of all vulnerabilities in embedded automotive systems [25] and access control issues make up 13.1 percent of all vulnerabilities

As can be seen in Figure 3, trust boundary 1 covers the first layer of all signals related to the BMS system and also includes the charger interface, which appears as a gateway to trust-boundary 1 and therefore enables cyber-security attacks on the BMS. Additionally, if the on-board diagnostic connector (OBD) does not provide protection mechanisms for trust-boundary 1 (usually the case in common vehicle designs), maintenance systems are included in trust boundary 1 as well as any OBD device connected to the car. A fact that has been often overlooked in the past and enabled security attacks recently described in [20, 21].

From the ISO 26262 aligned HSI of the battery management system (depicted in Table 3) it can be seen that the SecL of the directly connected signals (battery current 1 and battery voltage) are treated as 0 (not security relevant) while the signals provided via CAN bus (thus provided from outside of trust boundary 0 in Figure 3) are assigned a SecL = 2. This result from the fact that in order to raise a security attack, these signals would have to be manipulated in the vehicle directly at the battery management system and that these signals are within the same trust boundary 0 (see depiction of trust boundaries in Figure 3). On the other hand, the *SecL = 2* indicates a possible cyber-security vulnerability and thus requires built-in security solutions exhibiting a defense-in-depth approach are required.

As mentioned in previous section, the realization of the protection mechanisms on signal level requires coordinated design of multiple security technologies and currently no standardization for the coordination of security mechanisms has been established in the automotive domain. Thus, we follow the design guidelines for signal security we proposed in Table 3. Therefore, the three CAN signals required by BMS (assigned *SecL = 2*) have to be verified by the origin of message (this requires an immutable device identification) and message integrity (e.g. CANs CRC). Also, a detection of abnormal behaviour of the CAN bus including a check of message repeat rate and intrusion detection is required.

5. Conclusion

Vehicle manufacturers are currently gearing up for the newly arising cyber-security challenges. Although security standards do not need to be created from scratch for the automotive domain, they are frequently strongly related to the safety processes.

Functional safety and cyber-security engineering have an overlap regarding many facets, but some development artefacts (such as the definition of system borders (item definition vs. trust boundaries)) often differ completely. To that aim, we have proposed a way to identify trust-boundaries and developed security design guidelines for the signal security of complex systems via signal interfaces defined in the hardware-software interface (HSI) definition. We used the example of a battery management system to demonstrate a structured method for security boundary identification on system level based on the HSI definition required by the ISO 26262. This approach is based on the capitalization of an in-depth treatment of functional safety on signal-level for the determination of essential security architecture requirements on system level.
Additionally, we proposed design guidelines for signal security based on the cyber-security level (SecL) of the system assigned via initial security assessment (SAHARA).

Acknowledgments

This work is supported by the EMC2 project and the DEIS projects. The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement no 621429 (project EMC2) and the Horizon 2020 research and innovation programme under grant agreement no 732242 (project DEIS).

Furthermore, we would like to express our thanks to our supporting partners, the experts of the SoQrates working group.

References

[1] G. Scuro, “Automotive industry: Innovation driven by electronics.” http://embedded-computing.com/articles/automotive-industry-innovationdriven-electronics/ , 2012.

[2] P. Bisson, F. Martinelli, and R. R. Granadino, “Cybersecurity Strategic Research Agenda - SRA,” European Network and Information Security (NIS) Platform - NISP - Working Group 3 (WG3), vol. v0.96, pp. 1–201, August 2015.

[3] Vehicle Electrical System Security Committee, “SAE J3061 Cybersecurity Guidebook for Cyber-Physical Automotive Systems,” 2016.

[4] ISO - International Organization for Standardization, “ISO 26262 Road vehicles Functional Safety Part 1-10,” 2011.

[5] The SPICE User Group, “Automotive SPICE Process Assessment / Reference Model V3.0,” July 2015.

[6] ISO - International Organization for Standardization, “ISO/IEC 33000 Series on Process Assessment,” 2014.

[7] G. Macher, H. Sporer, E. Armengaud, and C. Kreiner, “A Versatile Approach for ISO26262 compliant Hardware-Software Interface Definition with Model-based Development ,” in SAE Technical Paper, SAE International, 2015.

[8] H. Sporer, G. Macher, C. Kreiner, and E. Brenner, “Resilient Interface Design for Safety-Critical Embedded Automotive Software,” in Sixth International Conference on Computer Science and Information Technology (J. Zizka et al., ed.), CCSIT ’16, Zurich, Switzerland, pp. 183–199, Academy & Industry Research Collaboration Center (AIRCC), 2016. https://doi.org/10.5121/csit.2016.60117

[9] G. Macher, H. Sporer, E. Brenner, and C. Kreiner, “Supporting Cyber-security based on Hardware-Software Interface Definition,” in Systems, Software and Services Process Improvement - 23nd European Conference, EuroSPI 2016, Graz, Austria, September 14 - 16, 2016. Proceedings (C. K. et. al, ed.), Communications in Computer and Information Science, Springer, 2016. https://doi.org/10.1007/978-3-319-44817-6_12

[10] A. Cimatti and S. Tonetta, “A Property-Based Proof System for Contract-Based Design,” in Software Engineering and Advanced Applications (SEAA), 2012 38th EUROMICRO Conference on, pp. 21–28, Sept 2012. https://doi.org/10.1109/SEAA.2012.68

[11] A. Soderberg and R. Johansson, “Safety contract based design of software components,” in Software Reliability Engineering Workshops (ISSREW), 2013 IEEE International Symposium on, pp. 365–370, Nov 2013. https://doi.org/10.1109/ISSREW.2013.6688922

[12] W. Damm, H. Hungar, B. Josko, T. Peikenkamp, and I. Stierand, “Using contract-based component specifications for virtual integration testing and architecture design,” in Design, Automation Test in Europe Conference Exhibition (DATE), 2011, pp. 1–6, March 2011. https://doi.org/10.1109/DATE.2011.5763167

[13] J. Iber, A. Höller, T. Rauter, and C. Kreiner, “Towards a Generic Modeling Language for Contract-Based Design,” in 2nd International Workshop on Model-Driven Engineering for Component-Based Software Systems (ModComp) 2015 Workshop Proceedings, p. 24, 2015.

[14] G. Macher, H. Sporer, E. Armengaud, E. Brenner, and C. Kreiner, “Using Model-based Development for ISO26262 aligned HSI Definition,” in EDCC Conference Proceedings, 2015.

[15] G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner, “SAHARA: A security-aware hazard and risk analysis method,” in Design, Automation Test in Europe Conference Exhibition (DATE), 2015, pp. 621–624, March 2015.

[16] D. Clare, S. Fry, H. Handschuh, H. Patil, C. Poulin, A. Wasicek, R. Wood, D. A. Brown, G. Cooper, D. Grawrock, A. Rajan, A. Tatourian, R. Venugopalan, D.

Wheeler, and M. Zhao, “Automotive Security Best Practices,” White Paper, pp. 1 –17, 2015.

[17] T. Hahn, S. Matthews, L. Wood, J. Cohn, S. Regev, J. Fletcher, E. Libow, C. Poulin, and K. Ohnishi, “IBM Point of View: Internet of Things Security.” white paper, April 2015.

[18] Windriver, “Improving Android Security for Automotive with a Defense-In-Depth Strategy,” White Paper, 2013.

[19] S. Otsuka, T. Ishigooka, Y. Oishi, and K. Sasazawa, “CAN Security; Cost-Effective Intrusion Detection for Real-Time Control Systems,” SAE Technical Paper 2014-01-0340, 2014.

[20] A. Greenberg, “Hackers cut a Corvette’s brakes via a common car gadget.” online, November 2015.

[21] K. Mahaffey, “Hacking a Tesla Model S: What we found and what we learned.” online, August 2015.

[22] Automotive Information Sharing and Analysis Center AUTO-ISAC, “Automotive Cybersecurity Best Practices Executive Summary”, 2016.

[23] A. Avizienis, J.-C. Laprie, and B. Randell. “Dependability and its Threats – A Taxonomy”. In R. Jacquart, IFIP Congress Topical Sessions, pages 91-120.2004. https://doi.org/10.1007/978-1-4020-8157-6_13

[24] Macher, G., Hoeller, A., Sporer, H., Armengaud, E. & Kreiner, C., ”A Comprehensive Safety, Security, and Serviceability Assessment Method”, In Proceedings of Computer Safety, Reliability, and Security - 34th International Conference, SAFECOMP 2015, Delft, The Netherlands, September 23-25, Springer International Publishing, 2015. https://doi.org/10.1007/978-3-319-24255-2_30

[25] RogueWave Software,”Top Automotive Security Vulnerabilities of 2015”, white paper, roguewave.com, 2015.