IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 16, 2020, accepted December 1, 2020, date of publication December 7, 2020,
date of current version December 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3043070

An Overview of Automotive Service-Oriented Architectures and Implications for Security Countermeasures

MARCEL RUMEZ[1], DANIEL GRIMM[2], REINER KRIESTEN[1], AND ERIC SAX[2]
[1]Institute of Energy Efficient Mobility, Karlsruhe University of Applied Sciences, 76133 Karlsruhe, Germany
[2]Institute for Information Processing Technologies, Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

Corresponding author: Marcel Rumez (marcel.rumez@hs-karlsruhe.de)

This work was supported by the German Federal Ministry of Education and Research (BMBF) within the Research Programme ICT 2020 through the AUTO-SIMA Project under Grant 13FH006IX6.

ABSTRACT New requirements from the customers' and manufacturers' point of view such as adding new software functions during the product life cycle require a transformed architecture design for future vehicles. The paradigm of signal-oriented communication established for many years will increasingly be replaced by service-oriented approaches in order to increase the update and upgrade capability. In this article, we provide an overview of current protocols and communication patterns for automotive architectures based on the service-oriented architecture (SOA) paradigm and compare them with signal-oriented approaches. Resulting challenges and opportunities of SOAs with respect to information security are outlined and discussed. For this purpose, we explain different security countermeasures and present a state of the section of automotive approaches in the fields of firewalls, Intrusion Detection Systems (IDSs) and Identity and Access Management (IAM). Our final discussion is based on an exemplary hybrid architecture (signal- and service-oriented) and examines the adaptation of existing security measures as well as their specific security features.

INDEX TERMS Automotive SOA, service-oriented architectures, connected vehicles, cybersecurity, firewall, intrusion detection system (IDS), access control.

I. INTRODUCTION

Modern vehicles are transforming from hardware- to software-defined platforms, making software the main driver for new innovations. Among the functions that are only made possible by software are, for example, increased road safety through automated driving functions, better integration of vehicles into everyday life through integration into the Internet with app stores for users, and more ecological mobility through electrification and shared mobility services [1]. From simple microcontrollers in the past, Electronic Control Units (ECUs) are therefore also developing into highly sophisticated computing machines that perform these advanced tasks such as artificial intelligence for autonomous driving. The electrical and electronical architecture (E/E architecture), consisting of the ECUs, its networking and the software running on it, is one of the key factors for new innovations in the automotive industry. This is also supported by the expectation that the growth of the software and E/E market will significantly exceed the growth of the entire automotive market [2]. In order to support new technologies such as connectivity, electrification, shared mobility or autonomous driving, we experience a real paradigm shift in the field of E/E architecture design [1]. Current vehicle architectures have up to 150 ECUs [3] and roughly three million implemented functions [4], which are interconnected via bus systems such as Controller Area Network (CAN) and organized by different domains (e.g., powertrain, chassis). The previous highly ECU focused development methods with statically defined transmitter-receiver dependencies at design time are very limited with regard to extensions and modifications [5], [6].

In order to increase the update and upgrade capabilities of vehicle systems in the future, due to dynamic customer requirements and market changes, future architectures will be more centralized. For this transformation, SOAs are

The associate editor coordinating the review of this manuscript and approving it for publication was Ahmed Farouk.

increasingly being introduced, which allow dynamic communication relationships at run time without static dependencies mapped on ECUs [7]. SOAs are recognized as one of the key elements to provide more flexibility, a better abstraction of low-level hardware and sensors and integration of external services and on-demand functions [1]. In combination with the rise of standardized operating systems, SOA allows dynamic, transparent and simplified access to information throughout the entire vehicle ecosystem. Therefore, if an application requires a sensor information, the information is provided by an internal control unit of the vehicle or over-the-air (OTA) by an Original Equipment Manufacturer (OEM)-backend for environmental data. Furthermore, changes or the integration of new functions in vehicles that are already in the field are much easier, since other information or functions can be subscribed as services [6], [8].

Furthermore, new network technologies and protocols are used to integrate this new communication paradigm. The reliable CAN protocol [9] that has been used for decades for internal vehicle communication is increasingly being replaced by automotive Ethernet technology in order to meet the changing requirements with respect to data rate, interoperability and information security [2]. On higher layers 5-7 of the ISO/OSI model [10] protocols like Scalable service-Oriented MiddlewarE over IP (SOME/IP) or Data Distribution Service (DDS) are used, which are specified in the AUTomotive Open System ARchitecture (AUTOSAR) Adaptive standard [11]. While some of these technologies are specific to the automotive industry, together with generic computing platforms based on standardized operating systems, there is a general trend in the evolution of E/E architecture towards ordinary IT network architectures [12].

With the higher degree of connectivity as well as the growing number of vehicle services [4], information security requirements are also increasing significantly. Security will become more and more a quality factor of vehicles [13]. Attacks on previous vehicle architectures (signal-oriented communication, mainly based on CAN protocol) have been demonstrated in recent years by various publications [14]–[16].

As a result, the subject of automotive security has gained a major focus within academic facilities, industry and standardization committees. On the one hand, this has led to various approaches from academia for countermeasures to protect against vehicle attacks [17], [18]. On the other hand, various standards and guidelines such as SAE J3061 [19], ISO/SAE 21434 [20] or AUTOSAR Secure Onboard Communication (SecOC) [21] were specified.

A. PROBLEM

The change from signal-oriented to service-oriented communication in vehicle architectures leads to a fundamental paradigm shift. Since communication design is no longer exclusively statically specified, previous security measures cannot be directly adopted in SOAs. Dynamicity in communication, rising data rates, a vast amount of new protocols, the integration of third-party services and standardized software components pose new challenges and requirements to automotive security measures. These were not in the scope of security concepts for the former signal-oriented E/E architectures.

B. SOLUTION

Existing security solutions used in the signal-oriented world have to be examined for the transfer to the service-oriented paradigm due to their changed communication characteristics. Since signal-oriented approaches are partly based on classic Information Technology (IT) measures, available IT solutions for use in automotive SOAs should also be investigated. The convergence of IT networks and E/E architectures further motivates this approach.

C. CONTRIBUTION

We present a comprehensive overview of current standards, protocols and developments in the field of automotive SOAs and E/E architectures. In particular, we sketch the newly emerging challenges for the security of SOAs. Furthermore, we explain various security measures (firewalls, access controls, IDSs) in general as well as approaches published specifically for vehicles. In our discussion, we analyze the applicability of existing approaches from the automotive sector and IT for SOAs. For this purpose we elaborate the deployment of the different security measures for an exemplary hybrid E/E architecture, which integrates the signal- and service-oriented world. Finally, we identified necessary research directions for future work to pave the way for a better security of SOAs.

This work is structured as follows (s. also figure 1): In section II, we explain principles of signal- and service-oriented communication in E/E architectures. In section III, we provide information about existing automotive protocols, vulnerabilities as well as attacks regarding SOA security and outline deviations to classical IT. Furthermore, section IV presents various countermeasures and classifies research approaches published for automotive networks. Section V includes an investigation and discussion concerning the applicability of existing security approaches in automotive SOAs. Finally, we summarize our work in section VI and outline possible research topics for future work (s. section VII).

II. SERVICE-ORIENTED AND HYBRID E/E ARCHITECTURES

The increased integration of service-orientation in the E/E architecture creates new challenges for the cyber security of vehicles. To evaluate the transferability of existing countermeasures, it is necessary to clarify the differences between the paradigms of signal-oriented and service-oriented communication. These differences manifest in terms of architecture in general, the runtime stacks used and the communication protocols.

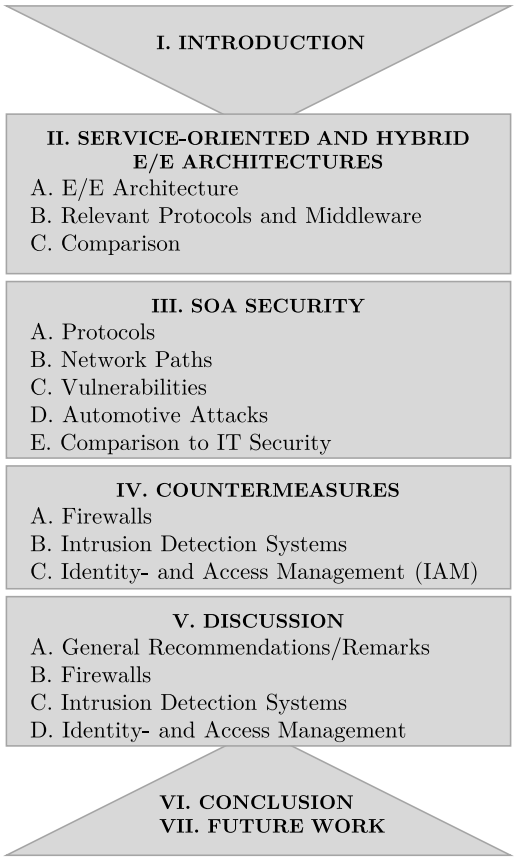

FIGURE 1. Structure of the research article.

A. E/E ARCHITECTURE

The distributed system of ECUs communicates via different bus systems. Together, the network of interconnected bus systems, ECUs and software running on them forms the E/E architecture. The most relevant bus system technologies are CAN, Local Interconnect Network (LIN), FlexRay, Media Oriented Systems Transport (MOST) and Automotive Ethernet. In current vehicles, multiple instances of any of these technologies can be used. To give an example, the E/E architecture of a 2019 Audi A8 consists of seven CAN buses, one MOST bus and one FlexRay [23]. For diagnostic purposes, an additional CAN bus and an Ethernet port are included. Besides, various LIN and proprietary systems are part of the architecture. LIN is especially used as very low-cost option for connecting to simple devices such as seat control. However, in this work we want to focus on future architectures that will develop due to the main drivers for innovation in automotive industry such as automated driving and connectivity [24]. While today CAN is the dominant system, the diversity of bus systems will decrease in the future and Ethernet will take over the leading role in the vehicle [25]. CAN will possibly still be used as a cost-effective alternative for low data rates and high reliability. The hybrid and hierarchical nature of future architectures, consisting of both Ethernet and conventional bus systems is shared among different researchers [6], [25]–[27]. In general CAN is used today for the signal-oriented paradigm, whereas Ethernet is mainly used for service-oriented communication.

1) CAN

CAN is a serial multi-master bus that supports different data rates with a maximum of 1 $\frac{\text{MBit}}{\text{s}}$ (High-speed CAN) and with CAN with Flexible Data Rate (CAN-FD) [28] up to about 8 $\frac{\text{MBit}}{\text{s}}$, depending on the physical transceiver. Modern in-vehicle networks consist of multiple CAN buses, that are interconnected via gateways and a backbone network, if necessary [25]. Each message contains a unique Identifier (ID) that defines the priority of a message. A message's payload is broken down into different signals of arbitrary length, however the sequence of signals in a message with a specific identifier is specified in the course of vehicle development.

On a specific CAN bus, a message with a specific ID is sent by no more than one ECU. All ECUs in the vehicle network can subscribe to any message ID. However, a gateway may be needed to route the messages, if sending and receiving ECU are attached to different buses. CAN messages are sent cyclically, which is mostly used to realize control and feedback loops with multiple ECUs or event-based, usually for interaction with the user (e.g., activation of the turn indicator, window regulator). Finally, the transmission of a specific ID can also be requested by another bus participant.

2) AUTOMOTIVE ETHERNET

Autonomous driving and connectivity increase the need for bandwidth [1]. In addition to diagnostics and flashing of ECUs, Ethernet is therefore increasingly used for communication within the vehicle. In modern Ethernet networks, each participant communicates only with its connected switch, which establishes collision-free point-to-point connections between all connected devices. Special physical layers are required for the vehicle to meet environmental requirements such as freedom from interference. With the 100BASE-T1 standard [29] (formerly known as BroadR-Reach standard) data rates of 100 MBit s are possible, with 1000BASE-T1 even 1 $\frac{\text{GBit}}{\text{s}}$ [30]. Further increases in data rates above 1 $\frac{\text{MBit}}{\text{s}}$ were recently published [31]. To bridge the gap of data rates between CAN-FD and the 100BASE-T1 standard but still maintaining the low costs of CAN, the 10BASE-T1S physical layer was specified [32], [33]. This can be a viable alternative to CAN for low-cost and hard-realtime requirements in the future. As an extension of the IEEE 802.1Q standard [34], Ethernet Time Sensitive Networking (TSN) has been introduced to the automotive domain to achieve higher real-time capability. Additionally, the IEEE 802.1Q standard specifies so-called tagged Virtual Local Area Networks (VLANs), that allow for logical separation of sub-networks.

Above the automotive-specific physical layer, the IT standard protocol stack TCP/IP (Internet Protocol (IP), Transmission Control Protocol (TCP), User Datagram Protocol (UDP)) is used. Furthermore, automotive-specific protocols such as Diagnostics over IP (DoIP) and SOME/IP were introduced.

TABLE 1. Comparison of AUTOSAR Classic and Adaptive platform. (adapted from [22].)

Property	Classic Platform	Adaptive Platform
Operating system	based on OSEK	based on POSIX
Code execution	from ROM	loaded into RAM
Application address space separation	no	yes
Scheduling	fixed	dynamic (preemptive multitasking, varying number of tasks)
Compiling	ECU compiled as whole	Applications are installed in POSIX processes
Configuration	Compiled system configuration	Loaded at runtime from file
Supported protocols / paradigms	signal-oriented	signal-oriented, SOA (DDS, SOME/IP)

3) RUNTIME PLATFORMS

More and more the software architecture of ECUs is based on different operating systems and standardized runtime stacks [5]. For example, the average annual growth rate of development efforts for operating systems and middleware is expected to be as high as the growth rate of development effort for automated driving functions [2]. The most important ones are the two platforms specified by the AUTOSAR foundation, namely AUTOSAR Classic and AUTOSAR Adaptive [7], [11], [22]. The Classic Platform is suitable for highest real-time requirements, whereas the Adaptive Platform makes use of stronger computing power and was designed to introduce the SOA paradigm to automotive industry. The current standard to which this article refers is Release 19-11. Between 2013 and 2016, the number of ECUs with AUTOSAR increased from about 50 million to almost 300 million [35]. Besides AUTOSAR, different platforms for infotainment domain exist, e.g., GENIVI,[1] QNX[2] [5], [36]. We focus on AUTOSAR-based ECUs, since these fulfill most of the safety-critical functions. A successful attack on safety-critical functions (e.g., control steering [14]) can cause major harm to the vehicles passengers and their surrounding. Thus, security for these platforms is also of great importance. Table 1 sums up the major differences. From the comparison, it can be seen that AUTOSAR Adaptive is far more dynamic than the Classic Platform. Applications are dynamically scheduled and installed. Furthermore, the Adaptive Platform relies on a subset of the standard POSIX interface for embedded real-time systems. Thus, the interface for all applications and software components to the machines operating system is standardized. In addition, the paradigms for communication of the applications via a network differ between the Adaptive Platform and the Classic Platform. Although the Classic Platform specifies the use of SOME/IP, it does not support the SOA paradigm. Instead only a translation from signal-oriented CAN communication to Ethernet is implemented, that defines a static mapping of signals to a specific service.

4) EXEMPLARY FUTURE ARCHITECTURE

In figure 2, an exemplary E/E architecture of a future vehicle is shown. The trend towards autonomous driving and connectivity leads to centralization and stronger computing platforms [1], [7]. For example, a central computing cluster is introduced that runs AUTOSAR Adaptive. The Adaptive platforms may consist of multiple processors and ECUs that are networked via switched Ethernet. The physical separation of functional domains (chassis, Advanced Driver Assistance Systems (ADAS), infotainment, body) still exists in our example, but for hierarchical Ethernet networks, this separation can also be a virtual separation using a different VLAN tag for each domain. The chassis and ADAS related functions are connected in a hierarchical Ethernet topology to the central unit. Since the functions must fulfill strict safety and timing requirements, these platforms run AUTOSAR Classic. For control functions with strong needs on reliability and low cycle times, CAN may be used to connect the ECUs to the switched Ethernet network. In our example, besides the central cluster also the I/O cluster and the connectivity control gateway are AUTOSAR Adaptive platforms. External communication is handled for smart charging, diagnostics and wireless connectivity through the connectivity control. In our example (s. figure 2), the AUTOSAR Classic platforms in the infotainment and body domain are connected via CAN or LIN to the I/O cluster because infotainment and body functions are mostly user I/O driven with low data rates. Hence, the cheaper CAN / LIN is preferred to Ethernet.

Summing up, figure 2 outlines a hybrid architecture of AUTOSAR Classic driven ECUs, that communicate mostly in signal-oriented paradigm via CAN and the Adaptive platforms that form a SOA using Ethernet. Furthermore, the architecture contains high-performance POSIX-based computing clusters as well as low-performance microcontrollers.

B. RELEVANT PROTOCOLS AND MIDDLEWARE

For automotive SOA, different protocols and middleware approaches are relevant. Most important are SOME/IP and DDS, since they are standardized in AUTOSAR and hence available as first-class solution to the industry. Furthermore, an approach from research for bringing SOA to CAN is outlined.

1) COMMUNICATION PATTERNS

In signal-oriented and service-oriented communication, different communication patterns are relevant. The typical control flow of the patterns is shown in figure 3. Figures 3a to 3c

[1]GENIVI Alliance, https://www.genivi.org/
[2]BlackBerry Limited, https://blackberry.qnx.com/

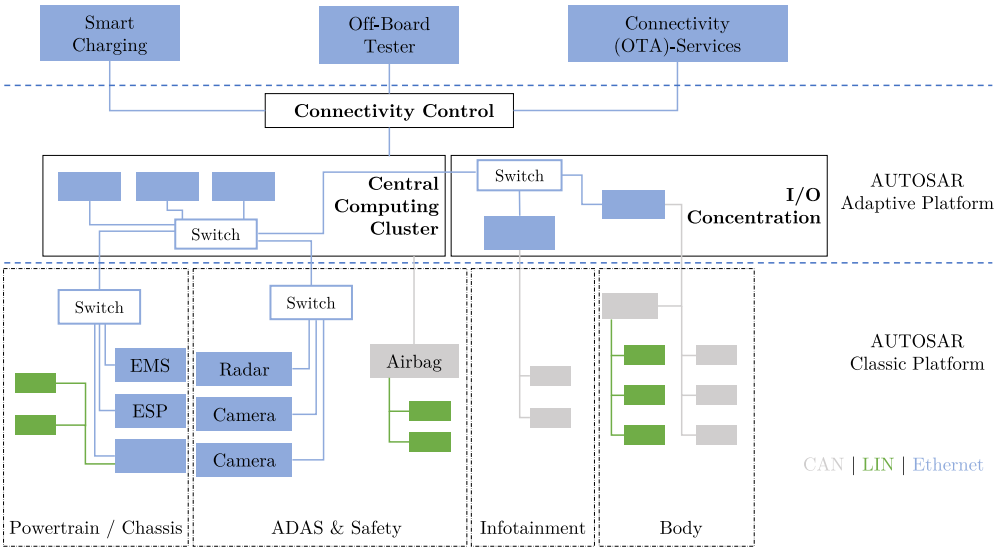

FIGURE 2. Exemplary hybrid E/E architecture based on AUTOSAR Classic & Adaptive. ECU colors indicate the respective bus systems they are attached to. (adapted from [7].)

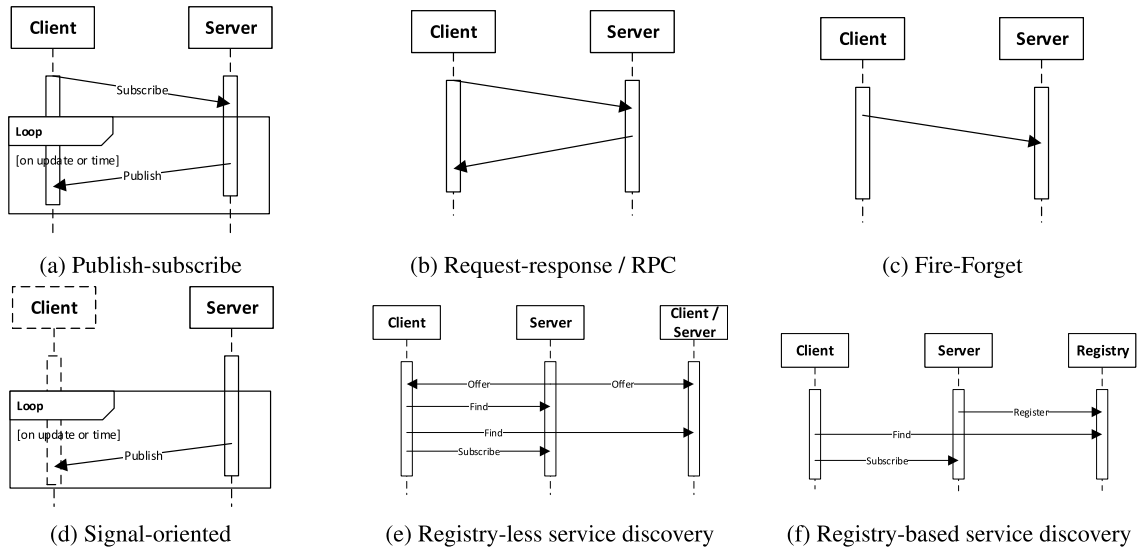

FIGURE 3. Overview of communication patterns in automotive architectures.

show the typical patterns of a SOA. The publish-subscribe pattern, the Remote Procedure Call (RPC) as well as the fire-forget pattern are driven by the clients. Thus, data is only transmitted to the clients if it is necessary. In comparison, 3d shows the signal-oriented communication, where the data is published as broadcast disregarding if any subscriber is present. However, the subscription to the publisher has to be made statically.

Therefore the main difference between the communication patterns of SOA and signal orientation is the time when the communication path is configured (time, when client is able to exchange data with the server). Generally, communication paths can be configured at design-time, at startup of the vehicle or at run-time. In signal-orientation, the configuration occurs fully static at design-time. It is specified, which ECU is allowed to send payload by using a static message ID. In addition, also the subscribers are assigned to IDs at design-time. In comparison, the SOA paradigm allows for configuration at startup or run-time. When the subscription to a service is specified at design-time, the communication path is instantiated at startup by just subscribing to the well-known server(s) that provides the service. For runtime configuration, a method to discover services and subscribe to them is required. Figure 3e and 3f show two options for service discovery. It can be either registry-based, where a dedicated registry server handles the offering and finding of services, or registry-less, where the discovery protocol enables servers and clients to discover each other.

2) SCALABLE SERVICE-ORIENTED MIDDLEWARE OVER IP (SOME/IP)

The specification of SOME/IP is not only a communication protocol but rather a middleware and standardized in AUTOSAR. The middleware simultaneously creates a certain degree of abstraction between the application and the network. For example, it is not necessary for an application to know which ECU provides a desired function. If the function is integrated on the same ECU, a local connection is established between the software components. If the function is located on another ECU, the middleware establishes a corresponding network link between two applications.

Communication Principles: The SOME/IP specification defines three basic types for the communication between clients and servers. Services in SOME/IP consist of events, fields and methods.

- **Events**: Event-based communication in SOME/IP provides a way for clients to subscribe required information from a provider. The provider is able to transmit the information to the clients at a specified time interval or if a value has been changed. In addition, various event groups can be created which contain different fields for subscribed clients.
- **Fields**: Services allow the definition of fields which can be read or modified by clients using get- and set-methods. In addition, they include the same notification concept of *Events*. However, the usage between *Fields* and *Events* is different, because fields are intended for status-like properties that refer to a history, whereas events shall be used for information that is only valid for a short time.
- **Methods**: In addition, *RPC* are available. These allow a client to execute a method provided by the server. There are two different variants. With the first variant, a return value is transmitted by the provider. With the *fire-forget* option, the client does not receive a return value on calling. Therefore this variant is mainly suitable for control commands.

The publish-subscribe pattern is implemented by fields and events. With the standard methods and get/set field methods, the request-response pattern is included. Finally, the fire & forget pattern can be fulfilled by SOME/IP methods using the respective option.

In addition, the SOME/IP Service Discovery (SOME/IP-SD) protocol allows for publication and subscription to events and fields. If clients want to use certain services, these have to be known or must be discoverable. The SOME/IP Service Discovery (SOME/IP-SD) protocol provides two different mechanisms to find or announce services, which are executed during system startup. With *offerService* a provider can announce his offered services as broadcast in the network. On the other hand, *findService* allows clients to find certain services that may not have been announced. SOME/IP-SD is registry-based.

Transport: *SOME/IP* uses the TCP/IP or UDP/IP stack for the transmission. To enable data exchange between AUTOSAR Adaptive and Classic (Inter-Platform Communication), SOME/IP is specified for both platforms. This allows a Classic ECU, which is additionally connected to legacy bus systems such as CAN, to act as a gateway. Signals are converted into a service in order to be able to offer it for Adaptive ECUs. In case the Classic control unit does not provide SOME/IP communication, signal-to-service conversion is only feasible on an Adaptive ECU.

3) DATA DISTRIBUTION SERVICE (DDS)

Another middleware that is recently used in automotive industry is DDS [37] which is standardized by the Object Management Group (OMG) and used across many industries. DDS is a data centric middleware based on the publish-subscribe pattern to control the flow of data between different nodes. Since Release 18-10 of the AUTOSAR Adaptive Platform published in November 2018, the DDS standard is implemented for automotive industry [11]. AUTOSAR includes only a subset of the various DDS standard parts.[3]

Communication Principles: The Data-Centric Publish-Subscribe (DCPS) interface of DDS organizes the data flow in DDS based communication. The information that is produced is published as *samples* in different *topics* that are identified by a unique name. A DDS communication consists of one or more *domains*, that contain the various topics. The domains are separated of each other, therefore no data exchange between different domains is possible. One essential element of DDS is content and data rate filtering of the published data for the subscriber of a topic. Hence, only data of interest is published to a topic and transmitted via the middleware. For example, only samples with a value > 300 and a message rate of five messages per second. The topics form a logical global data space defining the data model. Each topic is associated with a type, hence the middleware is able to manipulate the data correctly and perform type checking.

The OMG specified also a remote procedure call standard via DDS to enable the use of a request-reply pattern. This standard consists of a high-level abstraction on top of DDS that re-uses the substantial parts of DDS to define *requesters* as clients and *repliers* as services. In this case, every request-reply connection consists of two topics, a request topic and a reply topic. AUTOSAR Adaptive also provides this concept to enable the RPC pattern via DDS.

Transport: Regarding the underlying data transport model, Real-time Publish-Subscribe Protocol DDS Interoperability Wire Protocol (DDSI-RTPS) is used. DDSI-RTPS specifies that at least the communication via UDP/IP has to be supported by every DDS implementation, however most vendors also provide a TCP implementation and also shared memory access is possible. The use of further transport protocols is possible as long as the notion of specific unicast addresses and ports is supported and incomplete or erroneous

[3] 14 DDS standards in May 2020.

TABLE 2. Comparison of relevant protocols and middleware for automotive SOA.

		SOME/IP	DDS	eSOC [38]	Signal-oriented
Communication patterns	Publish-subscribe	x	x	x	
	Fire & Forget	x	x	x	
	Request-Response / RPC	x	x	x	(x)
	Static subscribe				x
Discovery / path configuration	Fully static				x
	Link to server	x			
	Link to data		x	x	
Supported protocols	CAN			x	x
	TCP/IP	x	x		
	UDP/IP	x	x		x
Security	Authentication		x		
	Encryption		x		
	Access control		x		
	Security Tagging		x		
	Brokerless		x	x	x

messages are identified. In AUTOSAR, the DDS specification allows the use of both UDP and TCP.

4) EMBEDDED SERVICE-ORIENTED COMMUNICATION (eSOC)

The *embedded Service-Oriented Communication* protocol outlined by Wagner *et al.* [38] introduces service-oriented communication based on the CAN protocol. They specify a 64 Bit service descriptor and a shorter 29 Bit identifer to implement a SOA, that maintains the priority principles of CAN and enables a simple integration with SOME/IP. The short identifier fits into the standard CAN identifier, hence no overhead is introduced by means of eSOC during operation. During startup of an eSOC communication, the 64 Bit service descriptor is used for service initialization via a *register server*. The register server is not working as broker for the communication, it only assigns the shortened identifier to the service provider. The eSOC protocol supports publish / subscribe and request / response as the two basic communication principles. With the first one, information is sent event-based or with a specific frequency from the server to the client. Using the second principle, methods can be called remotely and data can be exchanged as one-time transmission. An additional find/offer pattern allows for service discovery between the requesting and providing ECUs.

C. COMPARISON

The main differences between the relevant protocols are summarized in table 2. For comparison, the traditional signal-oriented approach is also included here. The CAN standard also specifies a request frame. However, this is typically not used to implement a request-response pattern in the traditional signal-oriented approach. The main differences manifest in the discovery of services if the discovery protocols of SOME/IP and DDS are used. DDS is the most dynamic, i.e. a subscriber only subscribes to a specific topic, which means that no static link to a server is established. Service IDs are invoked during discovery and the servers are unknown at first. In contrast, SOME/IP is server-based because a client must subscribe to events from a specific server. In addition SOME/IP requires a system-wide mapping of services to specific ports on UDP/TCP and specified service IDs during design time. However, AUTOSAR also allows for a design- or startup time configuration of the communication path for DDS and SOME/IP, if the discovery protocols are not used.

III. SOA SECURITY

With the introduction of SOA in vehicle networks, new challenges arise with regard to existing security measures. On the one hand, this can be attributed to the new protocol diversity, on the other hand, the communication paths are no longer statically defined. In the following sections we will describe these issues in more detail.

A. PROTOCOLS

Previous communication protocols such as CAN were based exclusively on OSI layer 1 & 2 (exception: diagnostic protocols). In table 3, we classified common automotive network protocols based on the ISO/OSI model. By using automotive Ethernet on layers 1 & 2, conventional protocols from Classic IT are used on layers 3 & 4 such as TCP/IP. In addition to benefits such as better interoperability or higher bandwidths, this development brings new risks with regard to information security. Over the past few years, numerous hacker tools have been developed in the IT world to simplify the use of such techniques [39]–[41]. The amount of different protocols used on Ethernet increases the complexity of networking and packet handling and induces higher risks of protocol design and implementation flaws. Hence, new vulnerabilities of the protocol stack may be uncovered in the future [41]. Furthermore, for protecting the information asset *confidentiality*, the protocols provide the feature to encrypt header- and payload data on different OSI layers. Thus, when integrating security measures, it has to be taken into account that certain

TABLE 3. Common automotive protocols within the ISO/OSI model. (Diagnostics specific protocols are italicized.)

<table>
<tr><th>ISO/OSI Layer</th><th colspan="6">Automotive Protocols</th></tr>
<tr><td>7</td><td>UDS, OBD</td><td rowspan="3">SecOC</td><td>UDS, OBD</td><td>UDS, OBD</td><td rowspan="3">SOME/IP, DDS</td><td rowspan="4">AVB/TSN</td></tr>
<tr><td>6</td><td></td><td></td><td></td></tr>
<tr><td>5</td><td></td><td></td><td>TLS/DTLS</td></tr>
<tr><td>4</td><td colspan="2">ISO TP, SAE J1939, TP 1.6</td><td>FlexRay TP</td><td colspan="2">TCP/UDP</td></tr>
<tr><td>3</td><td colspan="2"></td><td></td><td colspan="2">IPv4, IPv6, IPSec</td></tr>
<tr><td>2</td><td colspan="2" rowspan="2">CAN</td><td rowspan="2">FlexRay</td><td colspan="3" rowspan="2">Automotive Ethernet</td></tr>
<tr><td>1</td></tr>
</table>

information may not be available on every network node throughout the whole network path.

The security measures already included in the SOA protocols and signal-oriented communication are outlined in table 2. For message authentication and freshness of messages in signal-oriented communication, the AUTOSAR Classic standard introduced SecOC [21]. A message authentication code as well as a freshness value are attached to the payload of a message to prevent different attacks such as replay and spoofing. For the the message authentication code, a symmetric key is required. However, SecOC is a lightweight protocol for low-cost embedded devices and does not allow for encryption of the payload data on CAN. As SOME/IP and DDS are included in AUTOSAR, they can rely on transport layer protocols for security that are specified in AUTOSAR. AUTOSAR Adaptive specifies the following concepts:

- DTLS for secure communication over UDP
- TLS for secure communication over TCP
- IPSec for secure communication over IP

However, SOME/IP does not provide any additional measures for security. One approach to address this issue was presented by the researchers Iorio *et al.* [42], who provide a security framework for Ethernet-based SOME/IP communication. In contrast, DDS includes an additional standard for security [43]. With the additional DDS security measures, access control, authentication, encryption and tagging of data (e.g., 'confidential') is provided. The access control mechanism enables the developer to separately grant read and write permissions to DDS Domains and Topics.

B. NETWORK PATHS

In comparison to signal-oriented communication, the network paths are only defined at system runtime or startup. As a result, the ability to extract static characteristics (e.g., cycle times, routing tables) from the communication matrix specified in development is only partially feasible. For this reason, corresponding impacts on the design of security measures such as IDSs or firewalls have to be investigated (for a detailed analysis, s. section IV). As a result, securing the vehicle also involves these systems such as OEM back ends or the infrastructure. Furthermore, it has also to be considered that the system boundary (vehicle) is extended by externally used OTA services. As a result, securing the vehicle also includes these systems such as OEM-backend or infrastructure.

C. VULNERABILITIES

Since integration of SOA technology in vehicle architectures is still in its infancy (e.g., Volkswagen MEB [44]), no attacks of this kind are known at the time of this article. However, appropriate security analyses can already be performed for upcoming protocols and possible architecture designs. A detailed article on SOME/IP presented by Kreissl [45] included a threat analysis and risk assessment as well as suggestions for appropriate countermeasures for 18 identified threats. Furthermore, the SOME/IP-TP protocol amendment (SOME/IP-Transport Protocol Segmentation over UDP) was shown to be vulnerable for selective Denial-of-Service attacks [46]. An early work analyzed the vulnerabilities of the diagnostic protocol DoIP [47] and found it to be vulnerable in various ways such as spoofing, denial-of-service, or fingerprinting via OEM-specific fields in the protocol. As outlined in section III-A, with standard protocols and operating systems, common vulnerabilities are introduced. One example is the well-known buffer overflow vulnerability caused by IP fragmentation, which may also be found in an AUTOSAR implementation [46]. Another example is the possibility to use ARP cache poisoning, if the ARP table is not statically defined [48].

Besides the outlined threat analysis results we have to keep in mind, that with the SOA also AUTOSAR Adaptive is introduced to the automotive domain as runtime platform. With the standardized POSIX kernel and common operating systems, also the vulnerabilities of the kernel and operating system are common to all vehicles running the specific AUTOSAR Adaptive Platform implementation [41]. Hence, any possible vulnerability of the runtime platform has a larger impact to industry than with OEM-specific runtime platforms.

D. AUTOMOTIVE ATTACKS

In order to secure future vehicle architectures, it is essential to investigate already known attacks regarding their exploited vulnerabilities. The researchers Sommer *et al.* [49] published a taxonomy for automotive attacks as well as a database containing 162 published attacks [50]. The taxonomy contains 23 different categories and multiple layers of abstraction such as a short description of the attack, violated security properties or an explanation of exploited vulnerabilities. Since attacks of vehicles usually consist of several attack steps, the database also provides an extended

version, which contains 413 sub-steps of captured attacks. Based on this information, detailed attack analyses [51] can be performed. Besides academic institutions, there are also commercial repositories for automotive incidents provided by industry [52], [53]. The datasets of these repositories not only focus exclusively on attacks in vehicles but also include known vulnerabilities in backends or vehicle apps.

Furthermore, the United Nations Economic Commission for Europe (UNECE) WP.29 Working Party on Automated and Connected Vehicles (GRVA) [54] is expected to come into force in 2021, which will be the first regulation concerning information security in vehicles. The regulation will be mandatory for all OEMs of the associated member states for International Whole Vehicle Type Approval (IWVTA) [55] of new vehicles. In addition, a list of high-level threats and vulnerabilities as well as a mapping to real attack methods based on vehicle systems is included.

Vehicles with a mature SOA or hybrid architecture are currently not very often found on the roads. Accordingly, no attacks focusing specifically on SOA have yet become public. Nevertheless, due to known vulnerabilities of SOA technologies (see Section III-C) and successfully performed attacks on previous vehicle architectures, future SOAs will most likely be exploited and attacked. Especially due to the increasing spread of SOA technology, the exploitation of vulnerabilities is only a matter of time.

E. COMPARISON TO IT SECURITY

Although there is a trend in in-vehicle networks towards common IT networks in terms of physical (i.e. Ethernet) networking, there are still major differences that need to be taken into account in automotive security measures [12]. These specific requirements are not new to SOA but are generic to automotive security. However, the gained dynamics and flexibility of the networking of a SOA comes at the expense of the static predefinition of the network traffic. This was one of the advantages of automotive networks in terms of security compared to IT networks, because static behavior allows simple rule-based control [56].

In general, security for conventional IT networks does not have to worry about safety-critical systems, while failing automotive security is responsible for physical damage in the worst case. For example, taking over the steering or powertrain functionality can have a direct physical impact, which is why particularly high security precautions must be taken here. Closely related to this are the requirements for determinism and real-time of the security measures [57]. For example, a firewall that erroneously blocks a brake command by a false alarm or delays it too much by a slow analysis are not suitable for use in a vehicle. Another difference between IT security and vehicle security is the user interaction. While in IT every user can be confronted with security measures on his personal device, automotive security must run completely automatically in the background. In contrast to IT, a user reaction while driving is not possible here. Finally, the automotive-specific protocols and technologies (e.g., CAN, DoIP, SOME/IP) and system platforms (e.g., AUTOSAR Classic and Adaptive) require tailor-made solutions. Where common IT technology is used (e.g. infotainment systems), common IT security measures may also be sufficient. However, with increasingly centralized E/E architectures, a clear separation between safety-critical or automotive-specific systems and generic IT components is not always possible.

IV. COUNTERMEASURES

The protection of IT networks and automotive architectures should generally follow the defense-in-depth principle [61]–[63]. This approach is based on the integration of different protection layers to minimize the risk of a successful attack. Therefore, an attacker has to overcome several protection mechanisms to intrude into the system completely. In this article, we focus on countermeasures of firewalls, IDSs as well as IAM and examine below their applicability for service-oriented architectures.

A. FIREWALLS

Firewalls are part of the access control group and used in IT as an integral part to implement previously defined security strategies in the form of access restrictions [61], [64]. According to RFC 4949 [65], a firewall represents a gateway that allows data traffic between two different networks (e.g., the internet and internal company network) to protect resources against threats from other networks. Over the last decades different types of firewall systems have been established, which are divided into different classes (packet filter, proxy and application-level) [66] and shown in table 4. These filtering types can be used in different architectures and are often integrated as a combination of them.

Classic firewall packet filters are based on OSI layers three and four (network and transport layer). This allows the integration of filter tables based on protocol information of these layers to enforce predefined access control policies. An extension of the static packet filters are dynamic ones. These are able to store already analyzed packet information in order to consider them in subsequent filter decisions [64]. This type of filter is also called stateful filter, because access decisions depend on past behavior.

Proxy firewalls, which also work on the transport level of the ISO/OSI model, represent a further development. The firewall works as a broker between two networks and provides only a certain amount of services that a client can access, for example, from a network to another. This allows defining the amount of services in the firewall for each client. Compared to packet filters, special context information can be included in the rules [61]. It is also possible to limit the number of simultaneous links and throughput rates to avoid Denial of Service (DoS) attacks.

The application filters (application firewall) are specified on the top level of the ISO/OSI model (application layer). This type extends the filtering capabilities of the proxy firewall through precise information about the applications used. Therefore these filters are able to analyze application-specific

TABLE 4. Overview about established firewall types.

Firewall type	OSI Layer	IT protocols	Filter techniques	Config
Packet filter	3,4	TCP/IP, UDP/IP	Stateful, White/Blacklists Header-specific	static
Proxy	4	TCP/UDP	content-based	static/dynamic
Application-Level	7	http, FTP, SMTP	DPI, IDS	dynamic

TABLE 5. Classification of reviewed automotive Firewalls.

Reference	[18]	[58]	[59]	[60]
Filter type	packet	packet	packet, application	packet
Protocol	CAN, Ethernet, FlexRay	Ethernet	Ethernet	CAN, Ethernet
Location	Gateway	Gateway	Gateway	Gateway
Context	-	-	Payload	-
Filter technique	static	dynamic	dynamic	dynamic

TABLE 6. Classification of reviewed automotive IDS (adapted from [17]).

Reference	[68]	[69]	[70]	[71]	[72]	[73]
Feature	Cyber	Cyber	Cyber&Physical	Physical	Physical	Physical
Protocol	CAN	CAN	CAN	CAN	CAN	CAN
Context	-	-	yes	-	-	-
Evaluation	Real Data	Real Data	Real Data		Real HW	Real HW

user data (Deep Packet Inspection (DPI)), because for each application a dedicated proxy within the firewall is provided [67]. Furthermore, an authentication between the user (client) and the proxy can be established to make spoofing attacks more difficult [61]. Depending on the implementation variant, these firewall types are also capable of performing certain IDS features.

1) AUTOMOTIVE FIREWALLS

In recent years, the firewall technology of traditional IT has also become interesting for the automotive domain. A classification of the following reviewed approaches is shown in table 5. In 2014, Seifert and Obermaisser [18] presented a gateway firewall for time- and event-triggered as well as stream communication. The authors used a hierarchical timed automata to describe the different communication behaviors. For example, they defined an interarrival time with min and max thresholds for event-triggered communication in CAN networks or dynamic parts of FlexRay to detect deviations of them. They also explained that the presented approach explicitly works for a statically defined communication.

Another firewall approach was published by Pesé *et al.* [58], who considered both software and hardware aspects as well as specific automotive requirements. For the detection of attacks they used three different filter types. A classic packet filter was implemented on a Field Programmable Gate Array (FPGA), whereas the rate-limiting and stateful packet inspection were software-based. Their evaluation contains investigations on end-to-end latency and jitter, throughput, memory consumption and CPU load. Furthermore, the authors consider it possible to extend the approach for filtering DoIP or SOME/IP protocols on application level.

Another gateway firewall design presented by Holle and Shukla [59] for an automotive Ethernet switch includes packet and application filter. However, the authors do not explain the precise details about these filtering techniques. In addition, they provide an overview of future Ethernet architectures and illustrate the necessity for integrating firewalls.

The researchers Luo and Hou [60] performed a risk analysis for potential attacks by using an attack tree. For this purpose they analyzed different attack vectors (e.g., sensors, ECUs, interfaces) and calculated the probabilities of exposure based on the identified attack paths. Based on this, they derived a security concept for a gateway firewall including stateful packet filter and IDS features by using information entropy techniques. They also point out the limitation that traditional embedded real-time operating systems do not support any kind of access control and integrate a Discretionary Access Control (DAC) on their used FreeRTOS.

B. INTRUSION DETECTION SYSTEMS

Compared to firewalls or encryption methods, which are defined as proactive measures, IDSs are assigned to reactive

protection measures [17], [61], since IDSs detect possible attacks only when they occur. In contrast, pro-active measures are intended to preventively minimize the potential attack surface of the system. For detection of attacks, IDSs analyze data traffic in networks, evaluate system log files or examine user behavior. A basic distinction is made between host-based and network-based, depending on their location [61], [66]. Host-based IDS (HIDS) are used exclusively on workstations or servers to detect anomalies within this system boundary. An anomaly could be an attempt by an user to modify system-critical files or generally to bring the system into a risky state.

In addition, network-based IDS (NIDS) analyze the data streams within networks by creating a copy of all packets in order to check these with regard to protocol deviations or malicious user data. The detection techniques of both systems can be divided into two different types. Signature-based techniques are based on features extracted from already known attacks and regularly updated. However, this implies the limitation that only very similar attacks can be detected. In contrast, new types of attacks remain partially undetected. On the other hand, anomaly-based methods detect deviations in relation to predefined features that are extracted from a normal behavior. Statistical, protocol-specific, rule-based and heuristic techniques are often used for this purpose [61].

1) AUTOMOTIVE IDS

In addition to traditional IT, there is also an increasing research effort on IDS approaches for vehicles to detect attacks at an early stage as well as to support pro-active countermeasures such as firewalls. Due to growing dynamic communication parts within automotive architectures, with protocols like SOME/IP, firewalls based on static filter tables are not able to ensure sufficient security [59]. Furthermore, the already mentioned UNECE WP.29 specifically calls for the integration of in-vehicle IDS. An overview of such approaches for the automotive domain described below is shown in table 6.

An early work by Müter *et al.* [56] introduced eight different anomaly detection sensors (e.g., formality, location, plausibility of messages) to identify intrusions on the CAN bus. Especially, they compared the applicability of the sensors with respect to six different criteria, e.g., if the sensor can be developed only based on the vehicles specification or if messages of different bus systems need to be considered for detection. In recent years, various features have been developed which allow the detection of anomalies. A comprehensive survey for intra-vehicle IDS approaches was published by Al-Jarrah *et al.* [17] in 2019. According to their detailed analysis, three main categories were identified for classification the works (flow-based, payload-based, hybrid). For each main category, seven characteristics (technique, features, dataset, attack type, performance metrics and benchmark models) were defined to allow comparability. The authors also analyzed in more detail the features used for intrusion detection. Vehicles differ from classic IT by the fact that they consist of a composition of sensors, processing logic and actuators. In addition, they have interfaces to external systems (e.g., OEM backend). Therefore, vehicles can be defined as a Cyber-Physical System (CPS). Compared to IT, besides cyber features, which there are also physical features for detection. Cyber features include protocol and communication properties (e.g., cycle time, message length) [68], [69]. In contrast, physical features (e.g., speed, location, signal courses) can be used to determine the current vehicle state [70], [71]. According to the review of 42 IDS publications in [17], 81% used cyber features, 5% physical and 10% a mixture of both. However, there are further approaches based on physical features which use physical characteristics (e.g., voltage courses of transmitted bits, characteristic impedance) of sender/receiver and transmission channel [72], [73].

Furthermore, Grimm *et al.* [12] provide a classification for security monitoring within the automotive domain by defining three aspects (vantage, operational area, action). Each aspect includes different categories such as external communication or sub-network root. Each category is then assigned to automotive network representatives (e.g., central gateway, backbone network, network switch ECU on Ethernet). In addition, they also outline that IDS research today is focused on the signal-oriented communication, and only few works tackle intrusion detection for Ethernet and SOA.

C. IDENTITY- AND ACCESS MANAGEMENT (IAM)

An access control can generally be defined as automated prevention of unauthorized access from subjects to resources. Furthermore, certain rules (policies) can be specified to control and enforce this purpose. Therefore an access control ensures a rule-compliant use of resources. Over the last decades of the IT century, various access control models for authorization have been developed or extended accordingly. The models control which subject (e.g., user or service) in a system is authorized to access a specific resource/object (e.g., file, service). The access can be granted on different authorization levels (read, write, execute), which are defined by access policies for each subject or resource. In the following, the four most significant access control models are presented in a chronological order.

1) DISCRETIONARY ACCESS CONTROL (DAC)

In the DAC, the owner is exclusively responsible for assigning permissions to a resource [67]. For example, in a company this instance could be a head of department who owns a lot of data. In order to assign and manage the permissions of each subject, an Access Control List (ACL) is mapped to each resource, which defines the access rights [61].

2) MANDATORY ACCESS CONTROL (MAC)

The Mandatory Access Control (MAC) provides an access model for strictly controlling and enforcing permissions. Compared to DAC, permissions on resources are not mapped

to subjects, but rather security labels are assigned [61]. The labels may contain different levels of security (public, confidential, secret or top secret). At the same time, these labels are also assigned to the participating subjects of an organization. If a subject wants to access a resource, a central instance checks whether both have the appropriate label or security level.

3) ROLE-BASED ACCESS CONTROL (RBAC)

With Role-based Access Control (RBAC), authorizations are assigned as a set to specific roles. For example, the roles can correspond to different departments (purchasing, shipping, board of management) of a company. This eliminates the need to assign individual authorizations for each subject or object. The control and enforcement of RBAC is done centrally on a system or network. If a new employee is hired, he or she only has to be assigned to one role and thus receives the stored set of authorizations. In addition, this reduces the administrative effort of the responsible IT staff member, since changes can be made at a central location [61].

4) ATTRIBUTE-BASED ACCESS CONTROL (ABAC)

The basic idea of the Attribute-based Access Control (ABAC) approach is to allow or deny access to resources of service users via attributes [74], [75]. A distinction is made between the following attributes:

- *User attributes*: Describe the service user in more detail (e.g., age, department, title).
- *Action attributes*: Describe the action performed on the resource (e.g., read, write, delete).
- *Resource attributes*: Describe the object that is accessed, e.g., bank account, document.
- *Environment attributes*: Attributes that relate to time, place or dynamic aspects of the access control scenario deal, for example, with access permissions only to certain times.

By applying attributes, the rigid coupling between users and roles as well as roles and authorizations (s. RBAC) designed more flexible.

To fulfill the dynamic requirements of distributed systems, a dynamic access control decision is needed. For this purpose the user attributes are evaluated at runtime and compared with a stored security policy. This enables defining permissions without prior knowledge of specific subjects [74]. If a subject wants to access an object (1), the ABAC module checks whether the access is allowed based on the policy (2) and enforces the access decision (3) (allow, block) accordingly. Therefore it is no longer necessary to specify an individual subject. The eXtensible Access Control Markup Language (XACML) standard [76] is recommended for implementation. Furthermore, the standard supports the integration of subject and object attributes within access policies, which are the core of ABAC. The XACML architecture and associated modules define the authorization infrastructure necessary for an ABAC model.

5) ACCESS CONTROL TECHNIQUES

For access control models already mentioned, there are basically two different techniques for mapping access permissions to subjects and objects. Both variants can be specified using an access control matrix (s. table 7) [61]. This type is usually used with the DAC model. The permissions can be mapped either to the capabilities or to the objects (ACL).

a: CAPABILITY TABLE

This type of table specifies a set of permissions for a particular subject [77]. In table 7, the capabilities for each subject are defined horizontally for the three different objects (file 1, 2, 3). A capability can be represented by different formats (token, key or ticket). If a subject wants to access an object, the underlying system or application checks whether the permissions within the transferred capability are sufficient for the requested action.

b: ACCESS CONTROL LIST

This lists are mostly used within operating systems, applications or network devices (e.g., switches, routers). These include entries of different subjects who are authorized to access a certain object with defined permissions [61]. Furthermore, ACLs are also used in RBAC to assign permissions for objects to specific roles.

6) AUTOMOTIVE IAM

Until the introduction of AUTOSAR Adaptive 18-03 in 2018 [78], a comprehensive IAM for vehicular embedded systems was missing, with the exception of diagnostic applications. For this kind of functions a lightweight access control exists by using the Security Access service [79]. This service provides an authentication and authorization procedure for diagnostic applications based on challenge-response principle. For initiating an extended diagnostics session, an external diagnostics device sends a request to an in-vehicle ECU. Thereupon, the ECU responds with a random number to the diagnostics device, which calculates a key by using a secret algorithm. Subsequently, the device sends this key to the ECU, which checks the correct calculation and thus the validity of the authentication. As a final step, the authorization is performed by granting the execution of extended diagnostic services. However, this procedure has to be considered as unsecure, since various research studies have already shown its weaknesses and resulting vulnerabilities [15], [80], [81]. Through the integration of a fine-grained IAM, many of the exploited vulnerabilities of vehicles would have been avoided [45].

For these reasons, in recent years there has been increased academia research on approaches as well as specification efforts for managing permissions in vehicles.The approaches presented below are classified in table 8. In 2016, Kim *et al.* [82] published an decentralized approach for an authorization management by integrating an ABAC module into the AUTOSAR Classic Platform. The predefined

TABLE 7. Access control matrix for illustrating ACLs and capabilities (adapted from [61]).

Access Control Matrix					
	Subject	**File 1**	**File 2**	**File 3**	**File 4**
Capability	**Larry**	Read	Read, write	Read	Read, write
	Curly	Full control	No access	Full control	Read
	Mo	Read, write	No access	Read	Full control
	Bob	Full control	Full control	Full control	No access
				ACL	

TABLE 8. Classification of reviewed automotive IAM.

Reference	[82]	[83]	[84]	[86]
Model	ABAC	individual	CapBAC, ABAC	individual
Technique	Policies	Policies	ACLs, Policies	Policies
Context	-	-	Time	Vehicle states

attributes consist of CAN message, ECU service, and environment properties, which are used to create access control lists on each network node.

Another approach regarding a policy-based secure communication was presented by Hamad [83]. They focused on a trustworthy creation of security policies during the entire development process with various stakeholders (OEM and Tiers). The policies include certain specifications (e.g., security properties such as integrity or security protocol requirements) for provided or required services interfaces within a software component. Furthermore, a security module was introduced for each ECU, which acts as a distributed proxy firewall for in-vehicle TCP/IP communication. This allows fine-grained access control of each application to the underlying network stack based on stored policy.

The authors Gupta and Sandhu [84] focus their research on the increasing connectivity of the vehicle with its environment and the resulting risk for possible security vulnerabilities. For this the authors first explain different vehicular Internet of Things (IoT) architectures based on clouds and fogs. They further explain the structure of their authorization frameworks including different level and types of access control approaches (ACLs, CapBAC, ABAC) for static and dynamic communication. In addition, they illustrate their approaches based on use cases such as single cloud and multi cloud scenarios.

An overview and analysis of traditional (e.g., DAC or MAC) as well as modern access control models (e.g., CapBAC, Task-Role-Based Access Control (T-RBAC), Privacy-Aware Role-Based Access Control (P-RBAC)) with focus on CPS were published by Lopez and Rubio [85]. The authors identify requirements for access controls based on an exemplary future industrial system. Furthermore, they analyzed the applicability of existing access controls by introducing a comparison criteria such as dynamicity, scalability or flexibility, which are rated with high, medium or low.

With the introduction of AUTOSAR Adaptive, an IAM module was specified. This module enables a comprehensive rights management based on services. In detail, for each application or service (subjects) a list of access permissions for other resources or services (objects) can be assigned. Thus the AUTOSAR Adaptive specification is based on a capability-based access control. If a service wants to access an associated object during runtime, the modules Policy Decision Point (PDP) and Policy Enforcement Point (PEP) are specified for controlling and enforcing the permissions outside of applications and services. Furthermore, AUTOSAR Adaptive offers the possibility to use the integrated fine-grained access control of the supported DDS protocol for Domains and Topics.

An approach for the integration of a dynamic access control into an automotive middleware was published by Hugot *et al.* [86]. The authors first explain the previous weaknesses of automotive ECUs with regard to access management. Furthermore, they specify requirements for an access control (controls should be performed on OSI layer 5 and higher, middleware should control services depending on current vehicle state). As an example, the authors refer to an Adaptive Cruise Control (ACC) which is only allowed to communicate with a Transmission Control Module (TCM) when switched on. The presented approach is based on a dynamic MAC, which controls information flows between source and destination as well as their sequences by using a state machine. They also show the integration of their approach into the SOME/IP protocol by explaining and evaluating three different methods.

V. DISCUSSION

In this section, we explain potential challenges and benefits of security measures mentioned in section IV for the integration in SOAs. The analysis and discussion is based on the hybrid E/E architecture and different deployment points shown in figure 2.

A. GENERAL RECOMMENDATIONS/REMARKS

When securing networks, it can generally be noted that predominantly static communication (signal-oriented) is easier to secure than dynamic communication (service-oriented). This can be argued with the fact that static communication does not change during runtime. For example, no new network nodes are added and the nodes send predefined messages based on a specification. For the configuration of a firewall or IDS, this means that filter rules or normal behavior can be derived directly from the specification. This paradigm change based on SOAs also brings new challenges in other

domains such as the traditional IT [87]. For this reason, we believe it is essential to examine the requirements for new systems to determine whether dynamic communication is absolutely necessary or a static specification is also feasible. In general, when securing a new E/E architecture, a threat and risk analysis [88] should be performed first in order to derive appropriate security measures. However, in this article we want to analyze certain measures regarding signal and service-oriented communication without integrating them into a specific one.

For a more precise analysis of the different security measures we make the following assumptions:

- An OEM backend server has always the same and static network address from the vehicle point of view.
- The signal/service mapping is statically defined.
- The OBD interface is used by various external devices.

B. FIREWALLS

Traditional IT firewall techniques have existed for over 25 years and were introduced as a countermeasure for security design flaws in protocols and software. Firewalls can be classified as measures which do not solve the existing shortcomings of today's communication protocols. Rather, they attempt to limit the impact of these shortcomings through additional inspections. However, there is a risk that an attacker could bypass these additional controls and exploit persistent protocol weaknesses [67]. Moreover, firewalls are not able to perform semantic checks on the data. Only application-level firewalls are capable of detecting signatures based on already known attack techniques. However, they are not able to check safety-critical messages in real time as well as to take into account the current vehicle condition with respect to the context [86]. If an end-to-end encryption (e.g., on layer 7) is used, it should also be noted that a firewall is not able to inspect the data on this layer.

Transferred to future automotive networks, design flaws of classic IT should not be adopted by using secure protocols. Therefore it is necessary that network protocols are able to ensure information security assets such as confidentiality, integrity/authenticity and availability on different ISO/OSI layers. It should also be considered whether firewalls in combination with secure protocols is the best solution (due to the limitations mentioned above and the purposes for which they were used in the past) or whether other measures available today are more appropriate.

For a more detailed analysis of firewall protection capabilities, we have examined the three deployment points below:

1) CLASSIC/ADAPTIVE PLATFORM GATEWAY

At this point the transformation of service and signal-oriented communication is performed. Coming from the service direction, each service is statically assigned to specific messages and their payloads. This allows the filtering of statically known assignments and ranges of signal values. In reverse direction, the gateway acts as a service provider in order to provide signal information from legacy device (e.g., engine speed, wheel speed) as a service to clients of the AUTOSAR Adaptive platform. From the perspective of a firewall, filtering is more challenging. In principle, any client is able to subscribe services (assuming the client has necessary permissions). This means that source IP addresses are not permanently static and predictable during the runtime. Therefore classical packet filters (OSI layer 3,4) are not suitable. There is only the possibility to filter certain properties based on higher protocols like SOME/IP or DDS. However, it should be considered whether application-related filtering can be performed comprehensively within firewalls at all or whether this should not be solved by using IAM features.

2) CONNECTIVITY ECU

Considering the interface on-board/off-board communication within the connectivity gateway, different possibilities for firewall filtering can be identified depending on external network nodes. In the first use case we analyze a connection between a vehicle and an OEM backend. In principle, filtering on ISO/OSI layers 3 and 4 could be performed. In detail it would be conceivable to block all IP addresses except the static IP of the backend. In addition, a restriction of allowed ports could be integrated in a more fine-granular manner.

3) OBD-GATEWAY

At this point, various external network participants are able to establish a connection to the vehicle's internal network. Examples are diagnostic devices from different manufacturers as well as On-Board Diagnostics (OBD) dongles allowing a connection to a smartphone or third-party backend. If the DoIP protocol is used for diagnostics, a dynamic IP address is assigned to each external device. This means that filtering on certain IP-addresses is not generally feasible. If a device is connected, a white-list can be activated which blocks all other IP addresses during this session. Furthermore, a restriction of available ports could be implemented on OSI layer 4. It should be noted that different applications could use the same port. In case the CAN protocol and corresponding transport protocol ISO TP is used for diagnostics, other aspects have to be considered. This impedes the integration of rate-limiting features due to the separation time or amount of consecutive frames without flow control. Moreover, it has to be ensured that different diagnostic services can be requested in parallel e.g., with the same CAN ID (End-of-Line scenario).

C. INTRUSION DETECTION SYSTEMS

To date, IDS in the automotive sector has mainly focused on detecting anomalies on the signal-based CAN bus, while SOA and Ethernet are not considered. The discussion on IDS for automotive SOA has to include four major differences and challenges in contrast to CAN. One point is, it is of importance what network data and features shall be analyzed, since the nature of the data differs to the signal-oriented paradigm. Furthermore, the deployment and placement strategy needs

TABLE 9. Applicability of in-vehicle anomaly detection sensors [56] - adopted and discussed w.r.t. SOA requirements.

Detection Sensor	Applicability criterion						Applicable to SOA
	Specif. Based	Num. of Messages	Num. of Bus Systems	Different Message Types	Payload Inspection	Semantic Based	
Formality	true	1	1	na.	false	false	yes, but less
Location	true	1	1	na.	false	false	yes
Range	false	1	1	na.	true	false	no
Frequency	false	n	1	false	false	false	difficult
Correlation	true	n	n	true	false	false	yes
Protocol	false	n	n	true	false	false	difficult
Plausibility	false	n	1	false	true	true	no
Consistency	false	n	n	true	true	true	no

to be discussed, because the architecture itself changes due to SOA. This also leads to the third point, the necessity of host-based detection in addition to network-based detection. Lastly, besides anomaly based IDS also signature-based IDS can be of interest.

1) FEATURES FOR (SOA NETWORK) INTRUSION DETECTION

Lots of CAN intrusion detection systems focus on payload analysis, which is possible since the data rates on CAN are low, and the structure of the payload is specified during vehicle development. Besides analyzing the signals, Al-Jarrah *et al*. [17] also mention the flow-based analysis. In traditional IT, this is the most common method to network-based IDS [89], where the features the IDS observes are captured at OSI layers 2 or 3. For the automotive SOA on Ethernet with multiple layers, headers, encryption and larger payloads in comparison to CAN, payload based inspection is not feasible. Hence, it makes sense to use flow-based IDS for Ethernet and SOA in automotive architectures. Indeed, the features an automotive SOA flow should contain are unclear today. In comparison, Müter *et al*. [56] formulated the information that should be observed for automotive IDS, especially for anomaly based detection on the CAN bus. They outlined eight so-called *sensors* and compared them w.r.t. their applicability. We analysed which sensor are also applicable to the SOA and what challenges arise in using them. The results are shown in table 9. As outlined above, payload analysis is difficult, thus the sensors*Range, Plausibility* and *Consistency* are not feasible in general on the network level.

For information on sensor/actor level, where the range is limited by physical constraints and data rates are medium, the sensor are still feasible. The publish-subscribe and request-response communication schemes are not any more driven by specification of the sender, but driven by data or the requester. Thus, for SOA the *Range, Frequency* and *Protocol* are not specified beforehand (marked in orange color), making the analysis more difficult (e.g., requiring machine learning techniques). *Formality* of the data is still specified to some extent, but less than for signal-orientation (more protocols with more degrees of freedom, e.g., message size). Hence, detection capabilities are also lower. In addition, the criterion *Number of Bus Systems* makes no sense to SOA, since the whole architecture is one hierarchical Ethernet topology. Thus, for SOA this criterion should e.g., consider the number of flows, number of data topics or VLANs under analysis. Then, the *Location* is a valid sensor to identify e.g., VLAN hopping. Nevertheless, location may not be feasible because with any newly introduced service in the vehicle, with an updated service requesting further data, or even on every startup, new communication paths are established. Hence, also the benign locations (i.e. benign communication paths) are highly dynamic. This is basically the same for the *Correlation* sensor. Summing up the comparison, lots of sensors can not be simply transferred to SOA or make no sense any more. Instrumenting each service with monitoring capabilities would be the same as payload based analysis, which is not feasible. Hence, intelligent features on middleware layer are necessary to develop suitable network based IDS for automotive SOA. Nevertheless, the features must be generic to all SOA middlewares so that they can be implemented throughout the industry. Besides the features of flow-based IDS in IT, we suggest to include features capturing Quality of Service (QoS) parameters (derived from SOA middleware or from Ethernet protocols), as QoS parameters are means to specify the priority of data and separate data of different criticality. Another interesting option would be to include statistical measures on the payload that do not require deep packet inspection (e.g., information entropy [90]) because these could also be computed on encrypted traffic and induce low computational overhead.

2) SIGNATURE-BASED AND HOST-BASED DETECTION

On the one hand, with standardized operating systems and common well-known IT protocols, future attacks on vehicles may target more than one vehicle type or manufacturer. On the other hand, shared knowledge on threats and vulnerabilities such as Auto-ISAC [53] enables the industry to develop attack signatures. Hence, in the future it is more feasible for Ethernet and automotive SOA than for CAN and the signal-oriented paradigm to use signature-based detection. This would complement the anomaly-based detection, enabling the car detecting known as well as unknown attacks with higher accuracy. In addition, with standardized POSIX operating systems, dynamic scheduling of processes and the

introduction of third-party services into vehicles, host-based intrusion detection must come into the focus of research and industry. Here, more dynamics are introduced also on the software and application layer with the introduction of SOA, not only on the network. Having in mind, that payload analysis or monitoring of each service on the network is not feasible, the observation of host behavior becomes even more important. The AUTOSAR foundation also paid attention to host-based IDS, as the coming IDS standard already defines some host-based monitoring capabilities for AUTOSAR-based ECUs [91].

3) DEPLOYMENT

Addressing the last point of the discussion, from an architectural point of view, CAN IDS can be deployed anywhere on the bus, since all data is broadcasted. On Ethernet and SOA, the deployment must consider the availability of the data and stronger computing resources for higher data rates. Hence, in our exemplary architecture in figure 2 for network-based detection all data has to be made available to the IDS. To give an example, the data exchanged between the Camera and the ESP ECU can not be observed in the central computing cluster, as (in comparison to CAN) the switches establish unicast communication between the ECUs. The relevant ECUs for capturing the network data are the ECUs that contain the switches. Either the switches (in hardware) or the respective ECU (in software) have to calculate flow information on Ethernet and the communication of services in our SOA. In addition, the connectivity ECU is required to capture the network behavior with regard to the external communication. Afterwards the calculated flow information can be collected and analysed on one of the Adaptive platforms high-performance controllers.

Thus, a decentralized monitoring of the SOA communication behavior all over the vehicles network with a decentralized or centralized IDS e.g., on the Central Computing Cluster enables us to observe the whole SOA with sophisticated analysis techniques such as machine learning. Monitoring and IDS systems for CAN and signal-oriented communication should still be deployed additionally on the Classic/Adaptive Platform Gateways, because signal-to-service mapping is performed here and the high-frequency real-time control commands can be observed. However, the integration of CAN IDS and SOA IDS into a vehicle-monitoring approach is still in its infancy. Nevertheless, the currently ongoing standardization of Intrusion Detection Systems for vehicles [91] provides the technological foundation to implement these monitoring capabilities.

D. IDENTITY- AND ACCESS MANAGEMENT

An analysis based on the database from Sommer *et al.* [49] indicates that no or only a limited and weak IAM (diagnostics) has been implemented on ECUs in the past. As a result, attackers were able to execute or modify various functions on an ECU. The integration of measures for secure connections (authentication/integrity) in CAN or Ethernet networks can minimize the risk for a successful attack by limiting man-in-the middle attacks. However, there is still the threat of insider attacks [92], which cannot be prevented by these measures. In this case, the attacker acts from a compromised ECU or generally from a network node which is considered trustworthy by the receivers involved. As a result, no violation of the security objects (e.g., authenticity or integrity) occurs. As a result it becomes even more important to implement the principle of least privileges [93] consistently. Through the specification of the IAM module of AUTOSAR this principle is increasingly applied for automotive systems by capability-based access control methods for services. However, it should be noted here that a pure specification of privileges for each service still contains security risks. Due to the fact that vehicles represent CPSs and therefore operate in different physical states, this aspect is crucial for access controls [86]. However, the challenge is to determine the current state correctly in order to derive appropriate access decisions based on it. Therefore, the current context has to be taken into account for the authorization of control commands for actuators. The future architectures as shown in figure 2 could give an advantage compared with highly distributed architectures. Since the central computing cluster contains all sensor/actuator information and executes extensive functional calculations, context analyses could be performed centrally in this cluster.

Another important aspect which should be considered in access control models is the revocation of privileges. If the model is based on capabilities, the revocation is much more difficult compared to ACLs. Transferred to automotive SOAs, all service privileges are stored in the manifest file according to the AUTOSAR specification during development or when adding new services in the life cycle. If the service is rolled out on different ECUs, each manifest file has to be modified in case of a revocation. To avoid incorrect configuration of permissions, it is necessary to verify and validate them sufficiently. The authors Hu *et al.* published different verification and test methods for access control policies and models [94]. With capability-based approaches, which are statically based on pre-defined permissions, verification is easier than with models such as ABAC, which additionally include various attribute information for decision making.

VI. CONCLUSION

To sum up the major differences in security between the old paradigm of signal-orientation and the recent developments of service-oriented architectures, we see two contrasting points. On the one hand, SOA is based on switched Ethernet with its secured protocol stack, which minimizes the attack surface. Furthermore, physical architectures are developed with security-by-design in mind and not only focusing on security by obscurity. On the other hand, the rising dynamics with regard to data, network connections and software applications places even stronger demands on vehicular security mechanisms. From a security point of view, the more is specified during design time, the less attack surface we have and

the more we can check and verify during runtime with simple measures derived from specification. However, without dynamic deployment of applications, dynamic network paths or POSIX operating systems, future automotive architectures will not have the capabilities to follow customer needs of frequent updates and new features. Hence, dynamic behavior will rise and thus we expect that all active measures (firewalls, intrusion detection, access control) will be required in future service-oriented and hybrid architectures.

But, as the capabilities of firewalls, IDS and access control mechanisms are developing towards each other, a structured approach is required for a suitable placement of the measures that coordinates what data is analyzed where. Such a system-wide strategy should also be accompanied by threat modeling results (e.g., [95]). In addition, there are several countermeasures to ensure certain security properties. However, it must be analyzed which measures in combination fulfill automotive boundary conditions (e.g., real-time behavior, bandwidth) most efficiently. Moreover, to capture the increasing dynamics, sophisticated defense mechanisms are required. Static rules, thresholds or access lists are no longer sufficient to defend future vehicles without false alarms or missed attacks. Contextual information, such as the vehicle state, environmental behavior, or information regarding external interfaces needs to be included in cyber security decisions and analytics.

VII. FUTURE WORK

In this work, we compared the conventional signal-oriented architecture of in-vehicle networks with its emerging successor, the service-oriented or hybrid architecture. We outlined the challenges with regard to the security of these future automotive network architectures. Focusing on firewalls, IDS and access control, we emphasized the need of progressive defense methods, that need to be combined to ensure vehicular security with rising dynamics. Suggestions for the security of an exemplary architecture are given to sketch the challenges that security architects have to face.

For future research, we see context-aware security methods as a promising direction to cope with the infinite amount of possible situations, a vehicle may be in. However, to date it is an unsolved question, what information is necessary to describe the security context of a vehicle and how this information is combined optimally with the different security mechanisms. Additionally, the E/E architecture of the future will not end at the vehicles external borders, but spans to the backend systems that can also provide services to the vehicle. Thus, offloading a vehicles security computations may be a feasible option (e.g., [96], [97]). Furthermore, from a technological perspective more efficient access controls with revocation capabilities, intelligent IDS features for encrypted traffic or high-throughput firewalls for sensory data are an open topic for research. Today, AUTOSAR Classic and Adaptive already include some security features, but for the future this will not be enough. Hence, also sophisticated security mechanisms to be developed shall ensure their compliance to AUTOSAR or the standard may have to be extended as well. In case the eSOC protocol is used in future vehicles, the protocol should be evaluated more closely with focus on security.

ACKNOWLEDGMENT

The authors would like to thank the reviewers (Houssem Guissouma, Jaqueline Henle, Marc Schindewolf, and Andreas Vetter) for their inspiring comments. *(Marcel Rumez and Daniel Grimm contributed equally to this work.)*

REFERENCES

[1] O. Burkacky, J. Deichmann, G. Doll, and C. Knochenhauer, "Rethinking car software and electronics architecture," McKinsey Company, New York, NY, USA, Tech. Rep., 2018. Accessed: May 10, 2020. [Online]. Available: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/rethinking-car-software-and-electronics-architecture

[2] O. Burkacky, J. Deichmann, and J. P. Stein, "Mapping the automotive software-and-electronics landscape through 2030," McKinsey Company, New York, NY, USA, Tech. Rep., 2019. Accessed: Apr. 4, 2020. [Online]. Available: https://mck.co/3mSkfiW

[3] J. Deichmann, B. Klein, G. Scherf, and R. Stützle, "The race for cybersecurity: Protecting the connected car in the era of new regulation," McKinsey Company, New York, NY, USA, Tech. Rep., 2019. Accessed: May 10, 2020. [Online]. Available: https://mck.co/2xcXm4G

[4] V. Antinyan. *Revealing the Complexity of Automotive Software*. Accessed: 2018. [Online]. Available: https://www.researchgate.net/publication/327285609_Revealing_the_Complexity_of_Automotive_Software

[5] M. Traub, A. Maier, and K. L. Barbehon, "Future automotive architecture and the impact of IT trends," *IEEE Softw.*, vol. 34, no. 3, pp. 27–32, May 2017.

[6] A. Vetter, P. Obergfell, H. Guissouma, D. Grimm, M. Rumez, and E. Sax, "Development processes in automotive service-oriented architectures," in *Proc. 9th Medit. Conf. Embedded Comput. (MECO)*, Jun. 2020, pp. 1–7.

[7] M. Tischer. *The Computing Center in the Vehicle: AUTOSAR Adaptive*. Accessed: 2018. [Online]. Available: https://assets.vector.com/cms/content/know-how/_technical-articles/AUTOSAR/AUTOSAR_Adaptive_ElektronikAutomotive_201809_PressArticle_EN.pdf

[8] M. Schweiker and T. Huck, "CHASSIS ARCHITECTURES–partitioning of cross-domain functions," in *Proc. 7th Int. Munich Chassis Symp.*, P. P. E. Pfeffer, Ed. Wiesbaden, Germany: Springer, 2017, pp. 367–379.

[9] *Road Vehicles—Controller Area Network (CAN)—Part 1: Data Link Layer and Physical Signalling*, Standard ISO 11898-1:2015, 2003.

[10] *Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model*, Standard ISO/IEC 7498-1:1994, 1994.

[11] AUTOSAR. *Adaptive Platform*. Accessed: Jun. 17, 2020. [Online]. Available: https://www.autosar.org/standards/adaptive-platform/

[12] D. Grimm, F. Pistorius, and E. Sax, "Network security monitoring in automotive domain," in *Advances in Information and Communication* (Advances in Intelligent Systems and Computing), vol. 1129, K. Arai, S. Kapoor, and R. Bhatia, Eds. Cham, Switzerland: Springer, 2020, pp. 782–799.

[13] O. Burkacky, J. Deichmann, B. Klein, K. Pototzky, and G. Scherf, "Cybersecurity in automotive: Mastering the challenge," McKinsey Company, New York, NY, USA, Tech. Rep., 2020. Accessed: May 10, 2020. [Online]. Available: https://www.mckinsey.com

[14] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, p. 91, Aug. 2015.

[15] J. Durrwang, J. Braun, M. Rumez, and R. Kriesten, "Security evaluation of an airbag-ECU by reusing threat modeling artefacts," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Dec. 2017, pp. 37–43.

[16] L. Pan, X. Zheng, H. X. Chen, T. Luan, H. Bootwala, and L. Batten, "Cyber security attacks to modern vehicular systems," *J. Inf. Secur. Appl.*, vol. 36, pp. 90–100, Oct. 2017.

[17] O. Y. Al-Jarrah, C. Maple, M. Dianati, D. Oxtoby, and A. Mouzakitis, "Intrusion detection systems for intra-vehicle networks: A review," *IEEE Access*, vol. 7, pp. 21266–21289, 2019.

[18] S. Seifert and R. Obermaisser, "Secure automotive gateway—Secure communication for future cars," in *Proc. 12th IEEE Int. Conf. Ind. Informat. (INDIN)*, Jul. 2014, pp. 213–220.

[19] "Cybersecurity guidebook for cyber-physical vehicle systems," SAE Int. J3061, 2016.

[20] "Road vehicles—Cybersecurity engineering," SAE Int. 21434, 2020.

[21] "Specification of secure onboard communication (SecOC)," AUTOSAR Classic R19-11:2019, 2019.

[22] AUTOSAR. *Standards*. Accessed: Jun. 17, 2020. [Online]. Available: https://www.autosar.org/standards/

[23] Audi Service Training I/VK-35, "Audi A8 (type 4N) Electrics and electronics: Self study programme 664," Audi AG, Ingolstadt, Germany, Tech. Rep., Aug. 2017.

[24] S. Apostu, O. Burkacky, J. Deichmann, and G. Doll. *Automotive Software and Electrical/Electronic Architecture: Implications for OEMs*. Accessed: 2019. [Online]. Available: https://mck.co/2Pswpzr

[25] V. M. Navale, K. Williams, A. Lagospiris, M. Schaffert, and M.-A. Schweiker, "(R)evolution of E/E architectures," *SAE Int. J. Passenger Cars Electron. Elect. Syst.*, vol. 8, no. 2, pp. 282–288, 2015.

[26] R. Johansson, R. Andersson, and M. Dernevik, "Enabling tomorrow's road vehicles by service-oriented platform patterns," in *Proc. 9th Eur. Congr. Embedded Real Time Softw. Syst.*, 2018, pp. 1–9. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02156243/

[27] M. Maul, G. Becker, and U. Bernhard, "Service-oriented EE zone architecture key elements for new market segments," *ATZelektronik Worldwide*, vol. 13, no. 1, pp. 36–41, Feb. 2018.

[28] R. B. GmbH, "CAN with Flexible Data-Rate: CAN-FD, version: 1.0," Robert Bosch GmbH, Gerlingen, Germany, Tech. Rep., Apr. 2012. Accessed: Jun. 17, 2020. [Online]. Available: https://can-newsletter.org/assets/files/ttmedia/raw/e5740b7b5781b8960f55efcc2b93edf8.pdf

[29] *IEEE Standard for Ethernet Amendment 1: Physical Layer Specifications and Management Parameters for 100 Mb/s Operation Over a Single Balanced Twisted Pair Cable: 100BASE-T1*, Standard IEEE 802.3bw:2015, 2015.

[30] *ISO/IEC/IEEE 8802-3:2017/AMD 4-2017 Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks—Specific Requirements—Part 3: Standard for Ethernet Amendment 4: Physical Layer Specifications and Management Parameters for 1 Gb/S Operation Over a Single Twisted-Pair Copper Cable: 1000BASE-T1*, Standard IEEE 802.3bp:2016, 2016.

[31] *IEEE Standard for Ethernet—Amendment 8: Physical Layer Specifications and Management Parameters for 2.5 Gb/s, 5 Gb/s, and 10 Gb/s Automotive Electrical Ethernet: 2.5GBASE-T1, 5GBASE-T1 and 10GBASE-T1*, Standard IEEE 802.3ch:2020, 2020.

[32] *IEEE Standard for Ethernet—Amendment 5: Physical Layer Specifications and Management Parameters for 10 Mb/s Operation and Associated Power Delivery Over a Single Balanced Pair of Conductors: 10BASE-T1S*, Standard IEEE 802.3cg:2019, 2019.

[33] H. Zweck. *10 Mbps Ethernet Technology and the Challenges Facing Automotive Microcontrollers*. Stuttgart. Accessed: 2019. [Online]. Available: https://assets.vector.com/cms/content/events/2019/vAES19/vAES19_01_Zweck_Infineon.pdf

[34] *IEEE Standard for Local and Metropolitan Area Networks–Timing and Synchronization for Time-Sensitive Applications*, Standard IEEE 802.1AS:2020, 2020.

[35] V. Informatik. *Introduction to AUTOSAR: Vector E-Learning*. Accessed: 2018. [Online]. Available: https://elearning.vector.com/mod/page/view.php?id=437

[36] R. Coppola and M. Morisio, "Connected car," *ACM Comput. Surveys*, vol. 49, no. 3, pp. 1–36, 2016.

[37] Object Management Group. *Data Distribution Service (DDS)*. Accessed: 2015. [Online]. Available: http://www.omg.org/spec/DDS/1.4

[38] M. Wagner, S. Schildt, and M. Poehnl, "Service-oriented communication for controller area networks," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2016, pp. 1–5.

[39] Rapid7. *Metasploit: Penetration Testing Software, Pen Testing Security*. Accessed: Jun. 17, 2020. [Online]. Available: https://www.metasploit.com/

[40] Offensive Security. *Kali Linux*. Accessed: Jun. 17, 2020. [Online]. Available: https://www.kali.org/

[41] F. Luo and S. Hou, "Cyberattacks and countermeasures for intelligent and connected vehicles," *SAE Int. J. Passenger Cars Electron. Electr. Syst.*, vol. 12, no. 1, pp. 55–66, Oct. 2019.

[42] M. Iorio, A. Buttiglieri, M. Reineri, F. Risso, R. Sisto, and F. Valenza, "Protecting in-vehicle services: Security-enabled SOME/IP middleware," *IEEE Veh. Technol. Mag.*, vol. 15, no. 3, pp. 77–85, Sep. 2020.

[43] Object Management Group. *DDS Security*. Accessed: 2018. [Online]. Available: https://www.omg.org/spec/DDS-SECURITY/1.1

[44] M. Wille and O. Krieger. *Ethernet & Adaptive AUTOSAR: Key Elements of the New Volkswagen E/E Architecture*. Accessed: 2017. [Online]. Available: https://assets.vector.com/cms/content/events/2017/vAES17/vAES17_01_Ethernet-Adaptive-AUTOSAR-at-VW_Krieger_Wille.pdf

[45] J. Kreissl, "Absicherung der SOME/IP Kommunikation bei Adaptive AUTOSAR," M.S. thesis, Inst. Inf. Secur. (SEC), Universität Stuttgart, Stuttgart, Germany, 2017. [Online]. Available: https://elib.uni-stuttgart.de/bitstream/11682/9482/1/ausarbeitung.pdf

[46] M. Atad and A. Geynis. *The Devil is in the Fragments*. Accessed: 2019. [Online]. Available: https://www.escar.info/images/2019_ESCAR_US_Atad_Lecture.pdf

[47] J. Lindberg, "Security analysis of vehicle diagnostics using DoIP," M.S. thesis, Dept. Comput. Sci. Eng., Chalmers Univ. Technol., Gothenburg, Sweden, 2011. [Online]. Available: http://publications.lib.chalmers.se/records/fulltext/143639.pdf

[48] A. Talic, "Security analysis of Ethernet in cars," M.S. thesis, Dept. Commun. Syst., KTH Roy. Inst. Technol., Stockholm, Sweden, 2017. [Online]. Available: https://people.kth.se/~maguire/DEGREE-PROJECT-REPORTS/171006-Ammar_Talic_with_cover.pdf

[49] F. Sommer, J. Dürrwang, and R. Kriesten, "Survey and classification of automotive security attacks," *Information*, vol. 10, no. 4, p. 148, Apr. 2019.

[50] F. Sommer and J. Dürrwang. *IEEM-HsKA/AAD : Automotive Attack Database (AAD)*. Accessed: Apr. 1, 2019. [Online]. Available: https://github.com/IEEM-HsKA/AAD

[51] R. Bolz, M. Rumez, F. Sommer, J. Dürrwang, and R. Kriesten, "Enhancement of cyber security for cyber physical systems in the automotive field through attack analysis," in *Proc. Embedded World Conf.*, 2020, pp. 453–459.

[52] Upstream Security Inc. *AutoThreat Intelligence Cyber Incident Repository*. Accessed: Nov. 13, 2020. [Online]. Available: https://upstream.auto/research/automotive-cybersecurity/

[53] Auto-ISAC. *Automotive Information Sharing and Analysis Center*. Accessed: Jun. 25, 2020. [Online]. Available: https://automotiveisac.com/

[54] UNECE. *Draft New Un Regulation on Uniform Provisions Concerning the Approval of Vehicles With Regard to Cyber Security and of Their Cybersecurity Management Systems*. Accessed: 2020. [Online]. Available: https://www.unece.org/fileadmin/DAM/trans/doc/2020/wp29grva/GRVA-05-05r1e.docx

[55] UNECE. *Uniform Provisions Concerning the International Whole Vehicle Type Approval (IWVTA)*. Accessed: 2018. [Online]. Available: https://www.unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/2018/R000e.pdf

[56] M. Muter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," in *Proc. 6th Int. Conf. Inf. Assurance Secur.*, Aug. 2010, pp. 92–98.

[57] F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, E. Schoch, B. Wiedersheim, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: Implementation, performance, and research challenges," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 110–118, Nov. 2008.

[58] M. D. Pesé, K. Schmidt, and H. Zweck, "Hardware/software co-design of an automotive embedded firewall," SAE Tech. Paper 2017-01-1659, 2017.

[59] J. Holle and S. Shukla, "Gatekeeper for in-vehicle network communication," *ATZelektronik Worldwide*, vol. 13, no. 6, pp. 40–43, Dec. 2018.

[60] F. Luo and S. Hou, "Security mechanisms design of automotive gateway firewall," SAE Tech. Paper 2019-01-0481, 2019.

[61] S. Harris and F. Maymi, *CISSP All-in-One Exam Guide*. New York, NY, USA: McGraw-Hill, 2016.

[62] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *Black Hat USA*, vol. 2014, p. 94, Aug. 2014.

[63] M. Ziehensack and R. Pallierer. *Secure Automotive Ethernet for Automated Driving*. Accessed: 2016. [Online]. Available: https://www.elektrobit.com/tech-corner/secure-automotive-ethernet-automated-driving/

[64] K. Scarfone and P. Hoffman, "Guidelines on firewalls and firewall policy," NIST, Gaithersburg, MD, USA, NIST Special Publication 800, 2009, p. 41.

[65] R. Shirey. *Internet Security Glossary, Version 2*. Accessed: 2007. [Online]. Available: https://tools.ietf.org/html/rfc4949

[66] M. E. Whitman and H. J. Mattord, *Principles of Information Security*. Boston, MA, USA: Cengage, 2011.
[67] C. Eckert, *IT-Sicherheit: Konzepte-Verfahren-Protokolle*. Berlin, Germany: De Gruyter, 2018.
[68] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through Hamming distance," in *Proc. AEIT Int. Annu. Conf.*, Sep. 2017, pp. 1–6.
[69] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2016, pp. 130–139.
[70] A. Wasicek, M. D. Pesé, A. Weimerskirch, Y. Burakova, and K. Singh, "Context-aware intrusion detection in automotive control systems," in *Proc. 5th ESCAR USA Conf*, 2017, pp. 21–22.
[71] O. Avatefipour, "Physical-fingerprinting of electronic control unit (ECU) based on machine learning algorithm for in-vehicle network communication protocol," M.S. thesis, Dept. Elect. Comput. Eng., Univ. Michigan-Dearborn, Dearborn, MI, USA, 2017. [Online]. Available: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/140731/Thesis%20manuscript_v3.pdf?sequence=1
[72] M. Kneib and C. Huth, "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 787–800.
[73] M. Rumez, J. Durrwang, T. Brecht, T. Steinshorn, P. Neugebauer, R. Kriesten, and E. Sax, "CAN radar: Sensing physical devices in CAN networks based on time domain reflectometry," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2019, pp. 1–8.
[74] V. C. Hu, "Guide to attribute based access control (ABAC) definition and considerations," NIST, Gaithersburg, MD, USA, NIST Special Publication 800, 2013, p. 162.
[75] D. Servos and S. L. Osborn, "Current research and open problems in attribute-based access control," *ACM Comput. Surveys*, vol. 49, no. 4, pp. 1–45, Feb. 2017.
[76] OASIS. *Extensible Access Control Markup Language (XACML) Version 3.0*. Accessed: 2013. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
[77] V. C. Hu and K. Scarfone, "Guidelines for access control system evaluation metrics," NIST Interagency, Gaithersburg, MD, USA, Tech. Rep. NISTIR 7874, 2012.
[78] AUTOSAR. *Adaptive Platform 18.03*. Accessed: 2018. [Online]. Available: https://www.autosar.org/standards/adaptive-platform/adaptive-platform-1803/
[79] *Road Vehicles—Unified Diagnostic Services (UDS)—Part 1: Specification and Requirements*, Standard ISO 14229-1:2013, 2013.
[80] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *Def Con*, vol. 21, pp. 260–264, Aug. 2013.
[81] M. Ring, T. Rensen, and R. Kriesten, "Evaluation of vehicle diagnostics security–implementation of a reproducible security access," in *Proc. SECUREWARE*, 2014, p. 213.
[82] D.-K. Kim, E. Song, and H. Yu, "Introducing attribute-based access control to AUTOSAR," SAE Tech. Paper 2016-01-0069, 2016.
[83] M. Hamad, "A multilayer secure framework for vehicular systems," Ph.D. dissertation, Inst. Comput. Netw. Eng., TU Braunschweig, Braunschweig, Germany, 2020. [Online]. Available: https://www.researchgate.net/publication/341597533_A_Multilayer_Secure_Framework_for_Vehicular_Systems
[84] M. Gupta and R. Sandhu, "Authorization framework for secure cloud assisted connected cars and vehicular Internet of Things," in *Proc. 23rd ACM Symp. Access Control Models Technol.*, Jun. 2018, pp. 193–204.
[85] J. Lopez and J. E. Rubio, "Access control for cyber-physical systems interconnected to the cloud," *Comput. Netw.*, vol. 134, pp. 46–54, Apr. 2018.
[86] V. Hugot, A. Jousse, C. Toinard, and B. Venelle, *OMAC: Open Model for Automotive Cybersecurity*. Bochum, Germany: Ruhr-Universität Bochum, 2019.
[87] B. Rudra and O. P. Vyas, "Investigation of security issues for service-oriented network architecture," *Secur. Commun. Netw.*, vol. 9, no. 10, pp. 1025–1039, Jul. 2016.
[88] J. Dürrwang, K. Beckers, and R. Kriesten, "A lightweight threat analysis approach intertwining safety and security for the automotive domain," in *Proc. Int. Conf. Comput. Saf., Rel., Secur.*, 2017, pp. 305–319.
[89] N. Moustafa, J. Hu, and J. Slay, "A holistic review of network anomaly detection systems: A comprehensive survey," *J. Netw. Comput. Appl.*, vol. 128, pp. 33–55, Feb. 2019.
[90] M. Muter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Proc. 4th IEEE Intell. Vehicles Symp.*, Baden-Baden, Germany, Jun. 2011, pp. 1110–1115.
[91] E. Metzker. (2020). *Reliably Detecting and Defending Against Attacks: Requirements for Automotive Intrusion Detection Systems*. [Online]. Available: https://assets.vector.com/cms/content/know-how/_technical-articles/Security_Intrusion_Detection_AutomobilElektronik_202003_PressArticle_EN.pdf
[92] C. W. Probst, *Insider Threats Cyber Security* (Advances in Information Security), vol. 49. Boston, MA, USA: Springer, 2010.
[93] J. H. Saltzer, "Protection and the control of information sharing in multics," *Commun. ACM*, vol. 17, no. 7, pp. 388–402, Jul. 1974.
[94] V. C. Hu, R. Kuhn, and D. Yaga, "Verification and test methods for access control policies/models," *NIST Special*, vol. 800, p. 192, May 2017.
[95] T. Rosenstatter and T. Olovsson, "Towards a standardized mapping from automotive security levels to security mechanisms," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Piscataway, NJ, USA, Nov. 2018, pp. 1501–1507.
[96] G. Loukas, Y. Yoon, G. Sakellari, T. Vuong, and R. Heartfield, "Computation offloading of a vehicle's continuous intrusion detection workload for energy efficiency and performance," *Simul. Model. Pract. Theory*, vol. 73, pp. 83–94, Apr. 2017.
[97] G. Loukas, T. Vuong, R. Heartfield, G. Sakellari, Y. Yoon, and D. Gan, "Cloud-based cyber-physical intrusion detection for vehicles using deep learning," *IEEE Access*, vol. 6, pp. 3491–3508, 2018.

MARCEL RUMEZ received the B.Eng. and M.Sc. degrees in automotive engineering from the Karlsruhe University of Applied Sciences, Germany, in 2013 and 2015, respectively. He is currently pursuing the Ph.D. degree with the Institute for Information Processing Technologies, Karlsruhe Institute of Technology. Since 2015, he has been a Research Assistant with the Group of Automotive Security, Institute of Energy Efficient Mobility, Karlsruhe University of Applied Sciences.

DANIEL GRIMM received the B.Sc. and M.Sc. degrees in electrical engineering and information technology from the Karlsruhe Institute of Technology (KIT), Germany, in 2015 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Group of Systems Engineering, Institute for Information Processing Technologies (ITIV). Since 2018, he has been a Research Assistant with the Group of Systems Engineering, ITIV, KIT.

REINER KRIESTEN received the Dr.-Ing. degree. He is currently the Head and a Speaker with the Institute of Energy Efficient Mobility, University of Applied Sciences Karlsruhe. His main areas of research are software (SW) and systems engineering of cyber-physical and embedded systems and research in automotive security. Since 2003, he has been with Robert Bosch GmbH, where his applied research is based on a strong connection to the automotive industry, such as due to SW/system engineering and project management activities for automotive gateways and body computers.

ERIC SAX received the Dr.-Ing. degree. He is currently the Head with the Institute of Information Processing Technology, Karlsruhe Institute of Technology. He was the Head of test engineering with the MBtech Group. His main areas of research are processes, methods, and tools in systems engineering and data-driven and service-oriented architectures supported by the idea of machine learning. A tight link to industry derives from the fact that he was responsible for E/E at Daimler Buses from 2009 to 2014.