Security with Software-Defined Networking in Automotive Networks

Mehmet Çakır
Dept. Computer Science, *Hamburg University of Applied Sciences*, Germany
mehmet.cakir@haw-hamburg.de

***Abstract*—Cars are constantly equipped with new functions and intelligence. As they become more open to its environment using Vehicle-to-Everything (V2X) communication technologies, the necessity of security requirements becomes apparent. The concept of Software-Defined Networking (SDN) provides centralized control over network flows and devices. This work shows the state-of-the-art of SDN-based security in automotives and their security requirements. Furthermore, the concept of SDN and security concepts of SDN are explained. Finally, expectations of the use of SDN in cars will be discussed.**

***Index Terms*—network security, automotive networks, Software-Defined Networking**

I. Introduction

Nowadays cars are equipped with many software-based functions. The components in a car become increasingly interconnected. This results in complex network architectures which are difficult to maintain. Simultaneously previous architectures are not designed farsightedly for innovations [1]. According to a study by fortiss a redesigning of the automotive communication is needed [2]. In addition, a high bandwidth communication backbone is proposed where software components communicate in a service-oriented manner [1].

Automotive Ethernet has emerged as the next high-bandwidth communication technology for in-car backbones [3]. The protocol standard IEEE 802.1Q Time-Sensitive Networking (TSN) [4] enables to meet real-time requirements of the automotive environment [3]. Checkoway et al. showed attack surfaces over wireless interfaces [5]. Miller and Valasek remotely controlled a 2014 Jeep Cherokee by exploiting security vulnerabilities [6]. In addition, traditional routers and switches require a lot of effort to manage due to their heterogeneity in terms of the control plane. Software-Defined Networking (SDN) separates the control and data plane and introduces a programmable network providing abstractions to centralize the network management [7].

In this work, we will examine the feasibility of SDN-based security in automotive networks.

The remainder of this work is structured as follows. Section II analyses the current state of automotive networks. Section III presents security standards and guidelines of automotive networks. Section IV shows existing security concepts for automotive networks. Section V introduces SDN. Section VI provides an overview of the OpenFlow protocol. Section VII covers SDN-based security in LANs. Expectations and concepts of SDN in cars are mentioned in Section VIII. Section IX reviews relevant conferences for this research. Finally, Section X concludes with an outlook on future work.

II. State-of-the-Art Automotive Networks

Current automotive networks mainly consist of Controller Area Networks (CANs). CAN was developed by Bosch in 1991 and was the first bus system in a production vehicle [8]. In automotive networks, the term Electronic Control Unit (ECU) is also used for nodes. Nodes communicate with CAN messages. CAN messages will be prioritized by their ID. The lower the ID of the message the higher its priority [9]. CAN does allow a maximum bandwidth of $500\,\mathrm{kbit/s}$. Media Oriented System Transport (MOST) allows $150\,\mathrm{Mbit/s}$ [10]. So-called CAN-to-Ethernet gateways enable to connect a CAN with an Ethernet backbone [11] [12]. The IEEE 802.3bs standard enables Terabit Ethernet [13].

Automotive networks are organized in domains. Pretschner et al. mention different requirements for communication deadlines, data complexity, and communication patterns depending on the domain [14]. For example the Domain *Safety Electronics* has hard deadlines whereas the Domain *Multimedia/HMI* has soft deadlines. The majority of automotive networks consist of domain-specific CAN-buses connected via a central gateway. This means, every ECU of the same domain is connected to the CAN-bus of this domain regardless of its location in the car. Figure 1 shows a domain-based Controller Area Network. There are seven CAN-buses each marked by its own color. The numbers marking the outer edges denote the number of ECUs connected to a bus. All buses are connected with each other by a centralized gateway.

High-bandwidth communication for traditional Domain-based architectures can be enabled at least for transition phases by splitting up CAN-buses into sub-buses and connecting each of these sub-buses via a CAN-to-Ethernet gateway to a switched Ethernet backbone [15]. Figure 2 shows a zonal architecture of an automotive network. Each domain-specific CAN bus is represented by its own color. The numbers marking the outer edges of each sub-bus denote the number of CAN nodes connected to this sub-bus. A corresponding gateway receives the CAN messages an ECU generates. Subsequently the gateway encapsulates the CAN message in an Ethernet frame and forwards the frame to the Ethernet backbone. An application on the gateway has to know to which Ethernet port and destination address a CAN-ID hast to be associated with. For incoming traffic the possibly encapsulated CAN message

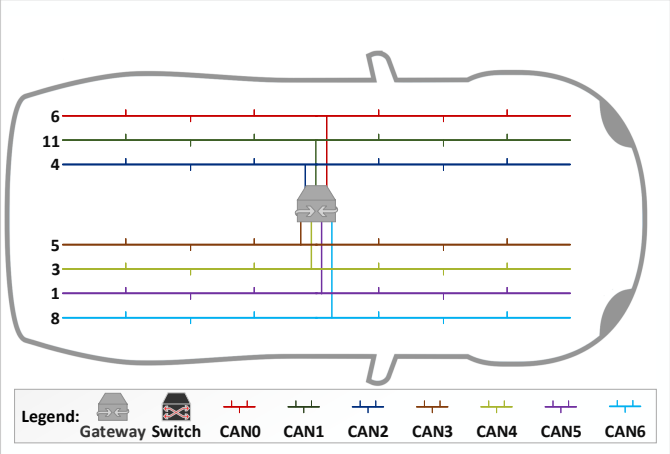

Fig. 1. Domain-based Automotive Controller Area Network (CAN)

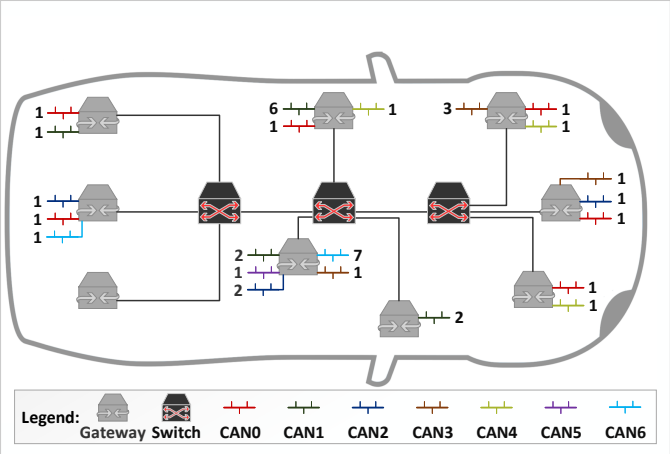

Fig. 2. Automotive network in a Zone Topology [15]

has to be decapsulated and given to the Ethernet port where the receiver node is connected. As this variation helps to speed up the traffic in the backbone, CAN-buses will still form the bottleneck.

III. Security Standards and Guidelines in Automotive Networks

Cars provide increasing sets of features like intelligent assistance systems or infotainment. With increasing connectivity the attack surface of cars increases too. This leads to the emergence of new potential vulnerabilities. Security standards mention what have to be considered to mitigate the chances of successful attacks. The ISO 26262 [16] and SAE J3061 [17] standards provide best practice process models for securing automotive Electric/Electronic (E/E) architectures. ISO 26262 addresses safety, whereas SAE J3061 addresses security. While these standards suggest what has to be considered in an E/E architecture there is no common language in assessing a level of cybersecurity in a vehicle. ISO and SAE aim to address security with their new ISO/SAE 21434 [18] standard throughout the supply chain to provide security by design and a common language.

Schnieder and Hosse show how to design attack-safe systems using the SAE J3061 standard [19]. Security requirements are derived in multiple successive steps. First, the physical boundaries and the sections to be protected are identified. In accordance with SAE J3061, a Threat Analysis and Risk Assessment (TARA) is performed. TARA identifies threats and rates risks. The results of the TARA essentially determine the design [19]. In the following subsections the steps *risk identification and assessment* of TARA are explained. Then, the next step formulating *cybersecurity goals* is explained with examples. Subsequently the derivation of *technical cybersecurity requirements* with a security in depth concept for automotive networks closes this section.

A. Risk Identification

The identification of risks requires to identify threats. Subsequently two methods are introduced. Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege (STRIDE) was introduced by Microsoft. It provides a qualitative method of analysis for gathering threats [20]. STRIDE maps threats (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) to Security-Attributes (Authenticity, Freshness, Integrity, Non-Repudiation, Confidentiality, Privacy, Availability, Authorization). By this mappings requirements for cybersecurity can be derived according to SAE J3061.

Attack trees are a supportive tool for identifying threats [19]. Figure 3 shows a generic attack tree. Nodes at the lowest level represent attempts performed by an attacker. If an attempt succeeds the attacker possibly reaches an intermediate goal. Intermediate goals are nodes between the lowest level nodes and the root of the attack tree. The achievement of intermediate goals can lead to the root. Intermediate goals can be logically linked. An AND link means that all intermediate goals must be achieved, whereas an OR link means that only one must be achieved. So the path from a node to the root shows the steps to reach the defined attackers goal at the root.

B. Risk Assessment

After the risk identification risks are assessed [19]. The risk assessment involves the *probability of access* (e.g., the probability of a successful attack and the *severity level of possible damage* (e.g., the amount of damage). The *probability of an access* is assessed by four criteria in HEAVENS[1] and OCTAVE[2].

- *Expertise* refers to knowledge about product categories and attack methods required for a successful attack.
- *System knowledge* refers to information about the system. Also the community size which may provide relevant knowledge for the attacker is important.

[1] **HEA**ling **V**ulnerabilities to **EN**hance Software **S**ecurity and Safety
[2] **O**perationally **C**ritical **T**hreat, **A**sset, and **V**ulnerability **E**valuation Framework

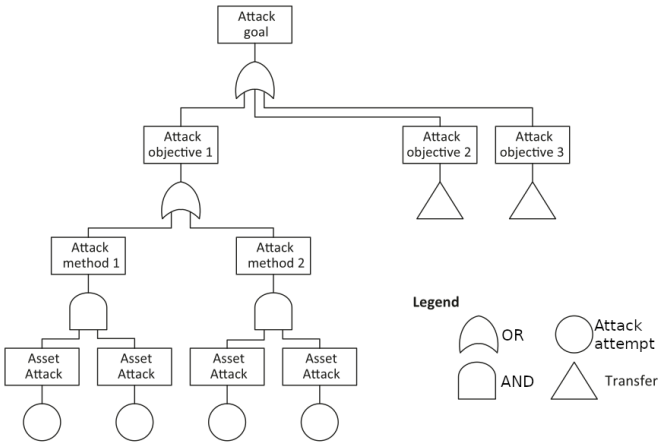

Fig. 3. Generic attack tree [19]

- *Equipment* refers to resources required to identify and exploit vulnerabilities.
- The *Window of opportunity* evaluates the available time for a successful attack according to the type of access and duration of the access.

In models like HEAVENS, the *severity level of possible damage* by unauthorized access considers the categories

- *Financial impact* (e.g., caused by loss of market share or damage claims.)
- *Comfort and availability limitations* from the perspective of the user (i.e. limitations of not security relevant systems like infotainment.)
- *Loss of confidentiality* (i.e. the violation of data protection regulations by unintentional disclosure of user data)
- *Safety relevance* assessed by the existing classification of hazards according to ISO 26262-3, if necessary - as with EVITA - taking into account the parameter of controllability of the vehicle in the corresponding operating situation.

Severity level and *probability* are linked in a risk matrix (see DIN IEC 62443-3 [21]). The results lead to protection levels and help which level of security is needed.

C. Cybersecurity Goals

After the risk assessment the formulating of cybersecurity goals is the next step. Cybersecurity goals are created for identified threats and describe, what to avoid or detect [22]. These goals are inverse to the respective threats. Examples for cybersecurity goals are [19]:

- *Prevention* of access over wired and wireless communication
- *Prevention* of unauthorized software updates
- *Prevention* of applying unauthorized and faulty configuration files

Cybersecurity concepts are derived from the cybersecurity goals. Cybersecurity concepts describe a superordinate strategy how the cybersecurity goals are to be achieved. Possible protection concepts in that strategy include [22]:

- Using of *access-protected communication* (e.g., authentication,VPN)
- Using of *digital signatures* for exchanged data like software updates and configuration files
- *Minimization of vulnerabilities* in developing and operation (e.g., guidelines for Secure Coding and static code reviews against these guidelines, vulnerability scans)
- *Switching off all interfaces* for debugging and diagnosis in operation of the vehicle
- *Using of existing and proven protection mechanisms* in hardware and software

D. Derivation of Technical Cybersecurity Requirements

At this stage the described security strategy is broken down to technical measures. One example is a holistic security concept like Defense in Depth [23]. Ihle and Glas mention that the probability of a successful compromise of the system under consideration is strongly reduced when several consecutive protection mechanisms are arranged [23]. For this, the authors discuss four layers of security. Schnieder and Hosse add a fifth layer [19]. Figure 4 shows the five layers:

- *Layer 5 - Protection of Critical Traffic Infrastructure:* Automotives are integrated increasingly into intelligent traffic systems. Intelligent traffic systems provide parameters for navigation systems as well as ACC Stop & Go and Heading Control [24] [25]. Unauthorized access on traffic servers or cooperative traffic lights can provoke accidents [26]. Due to the importance of the transportation sector crucial infrastructures must secured [27].
- *Layer 4 - Protection of the Connected Vehicle:* With increasing connectivity the vulnerability of cars increases too. Firewalls for example protect the car from unauthorized access.
- *Layer 3 - Protection of the Automotive Architecture:* The internal communication architecture must be secured by dedicated gateways. This ensures authorized access to the central internal communication systems. Furthermore Intrusion Detection and an addition of instant automated reactions are considered. It must be aware that a reaction can lead to critical states of the system.
- *Layer 2 - Protection of the Internal Communication:* There is an insufficient protection at this layer if communication protocols provide security only against random falsifications. Conscious manipulations like falsifications or insertion of messages must also be avoided. Additional security mechanisms must be introduced into the transmission protocols: (1) Ensuring the authenticity of the sender and receiver (Key/ID), (2) integrity of data (Encryption) as well as actuality of data (Timestamp,Message Counter).
- *Layer 1 - Protection of Individual Control Units:* The protection of individual ECUs forms a solid fundament for a holistic security concept. For example, interfaces for debugging, diagnostics and programming must be protected against unauthorized access. Code signing can

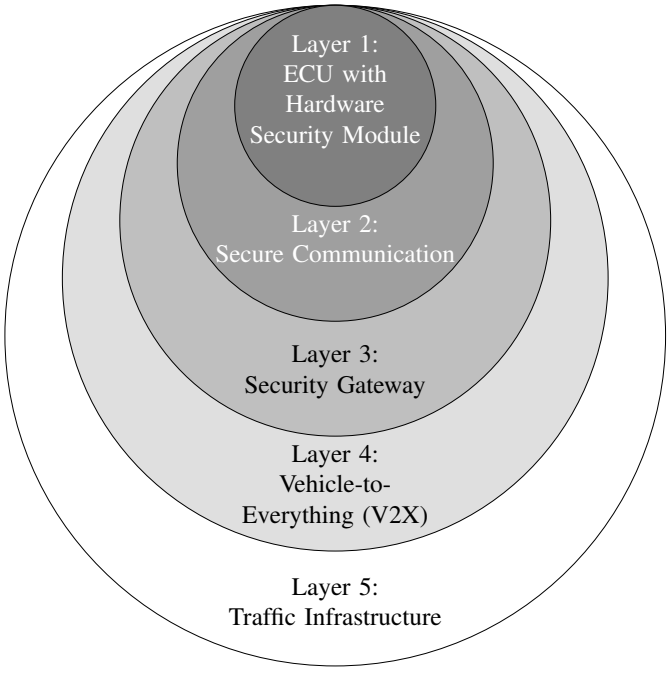

Fig. 4. Defense in Depth [19]

be used to ensure authorized software updates. Also the integrity of a ECU can be monitored in operation.

The SAE J3061 is currently a closely related document for automotive cybersecurity. However, it is only a best-practice model and not a standard for the industry. ISO/SAE 21434 is intended to supersede SAE J3061 and does not prescribe specific technologies or solutions related to cybersecurity. It intents to provide a common understanding of cybersecurity in automotive E/E architectures throughout the supply chain [28].

IV. Security Concepts in Automotive Networks

Cars become more open to the environment due to their connectivity. Examples like car platooning or communication with the external traffic infrastructure require several external interfaces.

Automotive Ethernet can enable new security solutions in automotive networks [29]. There are already models like the BMW X5, the Jaguar Land Rover XJ, the Volkswagen Passat and several other brands equipped with automotive Ethernet [30]. Since the majority of cars are based on domain architectures, Ethernet may coexist rather than completely replace legacy networks in the near future.

Ju et al. propose logically isolating domains connected with legacy networks and automotive Ethernet [30]. A connectivity domain which includes wireless interfaces like LTE, WiFi or OBD should be isolated logically from the cars internal network, to prevent access on critical components like brakes or steering control. They also define security levels to classify security functions required for secure communication according to the importance of the data.

Hu and Luo reviewed several secure communication approaches [31]. Message Authentication Codes (MACs) are used to provide authentication of a message at the receiver with a symmetric key approach. As encryption involves calculation, lightweight algorithms like Hash Message Authentication Codes (HMACs) [32] or additional hardware for calculation can help to reduce the latency.

Thing and Wu mention cloud computing security and adaptive security as additional considerations [33]. With the help of cloud computing security, the car's monitored network traffic could be sent to a cloud service which investigates the traffic behavior and may detect malicious events the car's security system itself could not detect. For adaptive security, the authors mean using adaptive reconfiguration of attack targets and deception tactics. Detection models with self-learning capabilities are also considered.

Automotive firewalls can detect and block packet injection by examining the CAN-ID or CAN-Payload for uncommon content. Intrusion Detection Systems (IDSs) analyze the transmitted traffic to detect anomalies and misbehavior of the flow and its content [31].

Higher security increases the end-to-end latency which is crucial for real-time traffic. It also increases the costs for hardware to compute fast enough to comply with real-time requirements. Security levels can limit which type of security is needed for different services or domains in an automotive network [31].

A. Mitigation Mechanism against Network Intrusion

Kwon et al. propose a mechanism which reconfigures ECUs and disables attack packets to mitigate damage by network intrusions [34]. If their Intrusion Detection System (IDS) detects a penetration attack the IDS instructs a mitigation manager, which is installed on a separate device or as a software module in a central gateway. The mitigation manager sends packets to ECUs which are expected to be damaged. Then the ECUs perform the following countermeasures: (1) The ECU reboots and switches its mode where only basic driving operations are permitted or security features are turned on. (2) The ECU broadcasts the CAN ID of the attack packet to ECUs in the same domain where every ECU receiving that attack packet discard packets with that CAN ID until the attack is over. This may lead a service being down during the attack, but saves processing attack packets in the ECUs and is therefore not suitable for critical CAN messages. If the attack addresses a head unit instead of a specific ECU the mitigation manager sends a mitigation message to the head unit to change its settings. This settings include external communication restrictions, access control for packets, application execution control to inject specific packets, and antivirus running. A topology, in which domain gateways are used, the mitigation manager sends a mitigation message to that gateway which is affected by an attack. The gateway drops packets by the CAN ID of the attack packet or changes the setting of the domain gateway into a security mode. In addition, the authors demonstrate a

mechanism to disable attack packets and then perform counter attacks against the compromised ECU.

B. Coexistence of Safety and Security

Lin and Yu show the trade-off between safety and security in Ethernet-based automotive networks based on secret key management, frame replication and elimination [35]. In general safety means to protect a system from a harmful impact by non-intentional actions like a disadvantageous design or configuration of the network. Security means to protect a system from a harmful impact by intentional actions like a crime motivated action by human.

1) Secret Keys: The authenticity of messages can be proven with secret keys. Secret keys can be used with various cryptography algorithms. For secret keys, the authors discuss the impact of cryptography algorithms on end-to-end latency. The authors consider end-to-end latency as the main safety requirement and authenticity as the main security requirement. The lower the end-to-end latency, the better the timing requirements can be met, thus providing greater safety. The stronger the cryptography, the harder it is to fake authenticity, resulting in higher security. In their authentication approaches they show methods how senders authenticate themselves to the receivers with Message Authentication Codes (MACs). A stronger cryptography increases calculation time. The result is the higher the security of the authentication approach the more the end-to-end latency increases and therefore the safety decreases.

2) Availability and Integrity: To increase the availability of frames, they can be sent multiple times instead of once. Frame integrity is checked to detect invalid message content. Time-Sensitive Networking (TSN) supports frame replication and elimination in standard IEEE 802.1CB. Frame replication is used to transmit a frame on multiple paths. Replication can be performed by the sender, a bridge, or a switch. The elimination can be performed by the receiver, a bridge or a switch. Replicated frames and redundant paths increase frame availability. Frame replication and elimination are able to enhance security. Suppose, in a network with two paths between a sender and a receiver and a switch on each path, a replicated frame is modified by an attacker. The sender and the switch can detect inconsistency by comparing two frames but can‘t decide which frame is the correct one. So both frames are rejected. Suppose the network has an additional path with a switch, there are now three replicated frames and one of it is modified by an attacker. The switch and the sender will not only detect inconsistency but also decide which frame is compromised. So the sender or the switch can recover the original frame. The authors discuss the impact of frame replication and elimination on frame availability as the main safety requirement and frame integrity as the main security requirement. The result is that frame replication and elimination can enhance safety and security at the same time. There are also two problems. One problem is how many replicated frames are needed to satisfy safety and security requirements for example in case all packets get compromised instead of only one. The other problem is how to assign the path for each frame without overloading the network.

C. VLAN Segmentation

Lin and Yu discuss VLAN segmentation to increase security in automotive networks [35]. VLANs separate physical networks into logical networks. It is used to make nodes inaccessible from another VLAN. This makes it more difficult for an attacker to reach a device in a different VLAN. In addition, separation reduces broadcast domains, which reduces network load. Moreover, in VLANs priorities are used for preferring traffic. This reduces the impact of cross-traffic on the latency of higher priority traffic.

D. TCP/IP Security Protocols

Lastinec and Hudec evaluate performance characteristics of different security protocols from the TCP/IP stack [36]. Thereby they focus on communication and processing delays, and response times. They consider topologies with an Ethernet backbone and lower CAN sub-networks as CAN is the most widely-used in-vehicle network to date. Due to the limited bandwidth and maximum payload size in CAN they aim to take advantage of the increased bandwidth and secure the traffic on the Ethernet backbone network. So they leave the lower sub-networks untouched. They conclude Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) might be suitable for non-realtime traffic instead of critical network traffic like control traffic. All other security protocols comply with their requirements. The best performance is given with Internet Protocol Security (IPsec) secured UDP traffic. The authors used less computationally intensive cryptography algorithms for IPsec.

V. Software-Defined Networking Concept

SDN extracts the control plane from all network devices into a central and independent control entity called SDN controller. Network devices still form the data plane. A well-defined Application Programming Interface (API) allows the control plane to manage the data plane.

Figure 5 shows a simplified design view of SDN. One can think of the control plane as the control flow in programming whereas the data plane is comparable to the data flow of a program. A Network Operating System (NOS) as the network controller is used as a centralized intelligent component in the control plane. Network applications are used for custom requirements which use the provided API of the network controller. They are implemented in the management plane of the controller.

In an SDN network the network devices are simple forwarding elements. Forwarding decisions are flow based. Network devices use flow tables. Forwarding decisions for packets are based on flow table entries. Packets have to match with a matching rule of an entry. A set of packet field values is used as match criteria. If it matches, the packet is forwarded to the corresponding out port defined in the entry. Else the packet

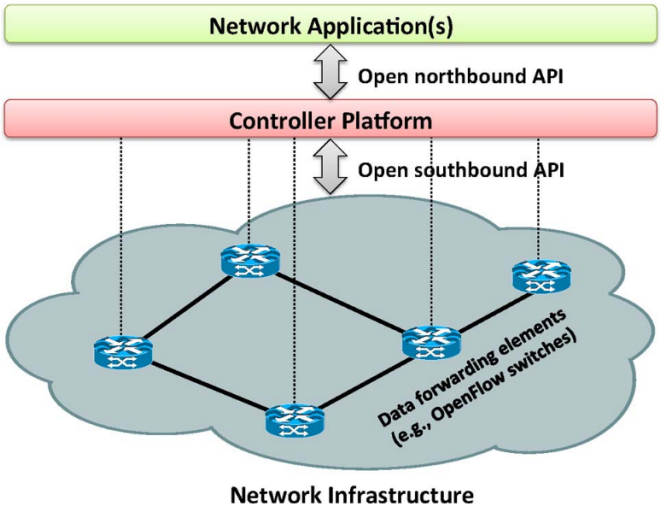

Fig. 5. Simplfied view of an SDN architecture [7]

is dropped or sent to the controller. Network applications can process the packet if the packet is not dropped.

Figure 5 shows the *northbound* and *southbound* API. The *northbound* API is the provided API for network applications by the NOS. It abstracts low-level instructions of the *southbound* API to program forwarding devices. The *southbound* API is defined by the used NOS. It defines how the control plane interacts with forwarding devices. A NOS or controller manages forwarding devices. There are two different types of controllers. (1) Centralized controllers have scaling limitations and are designed for small networks. (2) Distributed controllers can meet the requirements from small to large networks. A distributed controller can be a centralized cluster of nodes or a physically distributed set of elements. The first can offer high throughput for very dense data centers. The latter can be more resilient to different kinds of logical and physical failures. Updates on distributed controllers may not be applied immediately. That means that there will be a period of time not updated controller nodes read old values. Distributed solutions like ONOS provide a strong consistency model with an impact on the system performance. So all controller nodes will read the most recent value after a write operation. Distributed controllers communicate via east/westbound APIs (Figure 6). The functions of these interfaces include sharing data between controllers, algorithms for data consistency, network traffic monitoring and notification capabilities (e.g., check if a controller is up or notify a take over on a set of forwarding devices).

VI. The OpenFlow Protocol

The component that updates flow tables of forwarding devices is OpenFlow. OpenFlow is a southbound API for Software-Defined Networking (SDN). It was proposed by McKeown et al. [37]. To work with OpenFlow all forwarding devices must be enabled for it. The consortium website contains the OpenFlow Switch Specification [38]. Figure

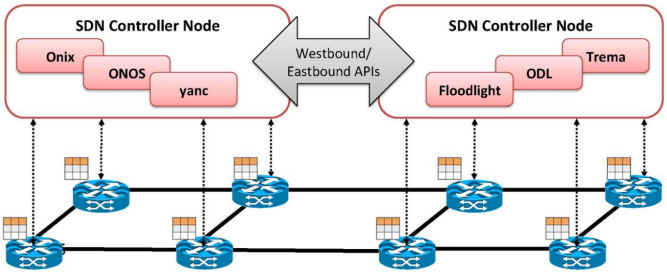

Fig. 6. Distributed controllers: east/westbound APIs [7]

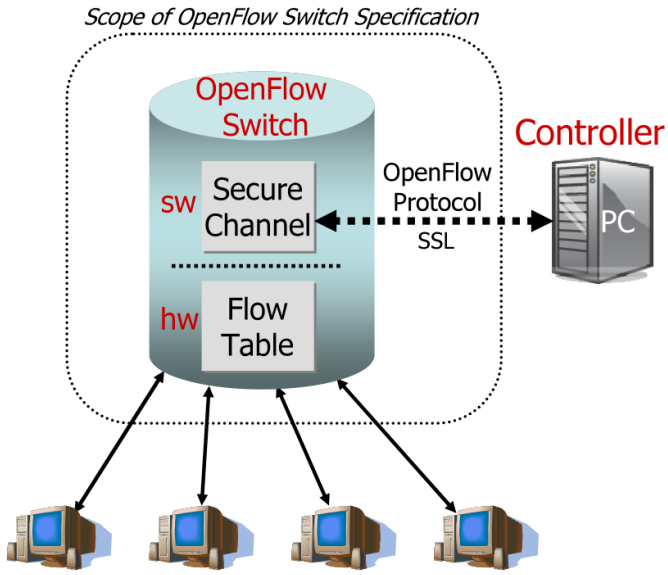

Fig. 7. Idealized OpenFlow Switch. The Flow Table is controlled by a remote controller via the Secure Channel [7]

7 shows an example. An OpenFlow Switch consists of at least three parts: (1) A Flow Table (2) A Secure Channel that connects the switch to a remote control process (called the controller), allowing commands and packets to be sent between a controller and the switch using (3) The OpenFlow Protocol, which provides an open and standard way for a controller to communicate with a switch [37]. There are two categories of OpenFlow switches. One category are dedicated OpenFlow switches that do not support Layer 2 and Layer 3. The other are OpenFlow-enabled general purpose commercial Ethernet switches and routers. Dedicated OpenFlow switches are just dumb datapath elements. They just forward packets between ports as defined by the controller. The flow can now be specified with rules looking for the packets header, such as the packets source MAC address, IP address, and the packets VLAN tag, and all packets from the same switch port. For each rule or flow-entry an action is associated with it.

VII. SDN Security Concepts in LANs

SDN enables the development and use of custom applications in the management plane. Further, this makes it possible to implement security services, traffic monitoring, access

control etc. to facilitate the network management. But the management plane coupled with the separation of the control and data plane in SDN requires to treat attacks differently. For example, in contrast to a traditional network, attacks can propagate alongside the data flow path through the controller and take down the controller. With that, forwarding devices can still forward traffic, but their data flow tables can no longer be modified.

Besides that, custom applications form another attack surface. Malicious applications or not secured applications can lead to undesirable behavior. The deployment of applications can follow two models which differ in security and flexibility. A strict model would only allow developers to make changes in applications which provides less flexibility but more security. With a relaxed model end-users can also make modifications by deploying custom applications which provides more flexibility but less security due to the higher probability of malicious code and misconfigurations [39] [40].

Xu and Hu introduce a Software-Defined Security (SDS) scheme where the data plane includes security devices along network devices [41]. They cite firewalls, intrusion detection systems, etc as security devices. In the control plane a Security Controller (SC) exists along the SDN controller. The SC collects security intelligence information from the security devices and uploads them to the security apps in the management plane. It also interacts with a SC agent in the SDN controller through the westbound interface to issue flow commands which the SDN controller performs on network devices. Additionally it has a global flow information of the current network and monitors the flows by interacting with the SC agent. This approach has two methods of detecting suspicious flows. On the one hand, unknown flows are forwarded by network devices to the SDN controller and to the SC. In this case the SC sends a warning log to the management plane. On the other hand, security devices can also detect suspicious behavior and push warning logs to the management plane through the SC. Security apps provide a view of the capabilities the security devices have and orchestrate security services. For example, security apps can issue other security strategies when abnormal behavior in traffic is detected by SC or security devices. To have reasonable distribution of the utilization rate across the security devices appropriate scheduling policies are needed. For that the authors list algorithms which are applicable for SDN networks.

Al-Zewairi et al. propose an SDN controller enhanced with security functionalities to detect and prevent IP and MAC spoofing attacks [39]. The authors denote it also as Software-Defined Security (SDSec) controller. It has two in-memory tables, which are the Switches Table and the Hosts Table. The Switches Table contains information about trusted network switches including the switch name, IP address, MAC address and available interfaces. The Hosts Table holds information about network hosts including the host name, IP address, MAC address, to which switch and on what interface it is connected in addition to its authentication status and the action to be taken to its traffic. Every host or switch must perform an authentication process with the SDSec controller before it can communicate on the network. When the new device is discovered, the controller checks the IP address and MAC address of the device against both tables. If there is no match in either tables, the device is marked as authenticated and is allowed to communicate on the network. Otherwise it is marked as not authenticated and the information of the new device is removed from both tables to allow it joining the network again in the future.

Krishnan and Oliver show a Distributed Denial of Service (DDoS) mitigation mechanism by blocking or redirecting the attack with flow rules [42]. The authors make use of monitoring the traffic in the network. The monitoring process takes place in the SDN controller. An attack is detected if the datarate at the source of a switch exceeds a determined threshold of bits per second. In this case, a corresponding flow rule is added to all switches connected to the data path from which the DDoS attack originates to block the corresponding ports. In SDN, DDoS attacks are possible at the data and control planes. The impact on the data plane behaves as in a traditional network, while a control plane failure brings down the controller and disables the control over all forwarding devices.

VIII. SDN Concepts in Cars

Since automotive Ethernet has emerged to fulfill the higher bandwidth requirements, the use of SDN could provide benefits regarding safety, robustness, security, cost efficiency, and future-readiness with easily updatable network devices [43]. However, real-time is crucial for automotive networks. The fail-safe operation of safety-critical systems must be provided. Besides that, emerging attack surfaces must be secured.

Häckel et al. propose a Time-Sensitive Networking (TSN) capable SDN approach called Time-Sensitive Software-Defined Networking (TSSDN) [43]. The static nature of in-vehicular networks favors their manageability. The SDN controller provides global knowledge of the network and all active flows. That enables control over flows and their timing. The authors implemented a controller application managing a Stream Reservation (SR) table. With an own SR table which manages TSN streams of the whole network, a global knowledge of all streams in the network is provided. Scheduling and transmission selection of network flows remain in the switches to guarantee timing. TSN streams are matched by the fields *Listener Group*, *Talker Address*, *Ingress Port*, *VLAN ID* and *Stream Priority*. The *Talker Address* is the source MAC address from which a TSN stream originates, and the *Ingress Port* is the switch port where the stream arrives. The *Listener Group* is the MAC multicast address to which the stream is forwarded to reach the subscribers. For using a stream multicast group in multiple VLANs the *VLAN ID* and VLAN *Stream Priority* is used. For flows for which there is no corresponding flow rule, the packets are forwarded to the SDN controller. To setup a new stream, the corresponding source node reports it to the controller. The controller registers the stream in its SR table and informs the whole network

about the new stream. The subscribing nodes of that stream update their SR table too and inform the controller about the subscription. Subsequently the SDN controller performs the corresponding flow rules on all switches along the path and informs the source of the stream about the subscription. A case study in their work shows that all deadlines are met without a delay penalty for the TSN traffic.

Haeberle et al. introduce an SDN architecture for automotive networks considering requirements for real-time and infotainment [44]. To meet real-time requirements TSN is selected. The data plane includes schedulers, rate limiters, firewalls, fail-safe mechanisms and redundant links. Besides a network controller a management system authenticates components and applications, manages TSN configurations, controls the automotive network, and keeps an inventory of components and applications and their permissions. In addition, the authors suggest that the northbound interface of the controller and the discovery mechanism are well-defined and the definitions need to be accessible by all potential manufacturers of data plane devices for interoperability between devices of different manufacturers. The authors discuss the following points for further explanation of their architecture sketch:

- *Data Plane:* Redundant links between switches are used. During normal operation redundant links are bonded using link aggregation. Scheduling variants can be used for load-balancing. Another option is 1+1 protection, where the sender sends the traffic over both links. If one link fails, the receiver selects the traffic from the functioning link. In case of insufficient bandwidth over the single link, best-effort traffic can be dropped. Sensor rates could be reduced to the minimal safe rate to reduce total traffic.
  Network traffic is classified into hard real-time traffic, soft real-time traffic, best-effort traffic and network configuration traffic. For real-time guarantees soft real-time traffic can operate in a degraded state. Rate limiters prevent faulty components from flooding the network. Flow rules isolate non-related traffic from each other. MACsec (IEEE 802.1AE) or AUTOSAR SecOC [45]. A provided Internet uplink is secured using a firewall.
- *Control Plane:* The network controller configures the network. In-band signaling reduces cabling and therefore also cost and weight. Access control lists and different permission levels regulate the access of applications to the network controller.
- *TSN Configuration:* Connecting new safety-critical devices requiring TSN, requires updates for routing and re-calculation of the TSN schedule. A hybrid approach is proposed to ensure a safe operation of the TSN reconfiguration. First the network controller updates the flow tables for the new TSN stream without changing the existing flows. This may not be optimal and can even require to disable less critical systems. As soon as an Internet connectivity is available, the calculation is triggered in a cloud service to re-calculate an optimal TSN schedule, if possible.
- *Device and Application Discovery:* New devices and applications require to authenticate themselves to the SDN controller. The authentication is performed by sending a signed manifest to the controller. The information included in the valid manifest is stored in the local device inventory and the network is re-configured to meet the requirements.
- *Failover Scenarios:* As mentioned earlier, if one of the redundant links in the backbone fails, all traffic is redirected through the other link. To ensure safe operation a precalculated outage schedule for TSN traffic is applied. If the bandwidth of the single remaining link is insufficient for critical traffic, non-critical traffic is restricted or even stopped. In case a switch or both links of the redundant links fail, it has to be ensured that safety-critical systems can still operate or even stop the car if necessary. The authors mention to connect such systems via a back-up network like a bus system. In case the controller fails backup flows and a backup TSN schedule configured by the controller on the switches are proposed as a precaution. Similar to backbone link fails, communication of non-critical systems may be restricted or stopped.
- *Security of Devices and Applications:* New devices have only access to the network for discovery purposes. Applications on devices have no network access by default. Further access is granted, if the device can provide a signed manifest by a trusted manufacturer. The device must also provide a signed manifest to get applications granted for further network access. For verification of the manifests a certification authority (CA) store containing the certificates of all providers is suggested. The car could query the CA store or keep a local copy of it. Furthermore, the car must refresh its local CA regularly. Thus, revoked certificates are noticed due to compromise or loss of trust. Certificate revocation could also be applied to existing and new devices and applications to deny them. Additional integrity checks of applications are proposed to prevent threats from altered applications by attackers. Isolation of applications and monitoring of their resource usage is needed.
- *Network Security:* Flows are derived from the requirements stated in the manifest of devices and applications. It must be ensured that devices and applications do not exhaust the resources of the network. Flows of devices or applications communicating with the outside world are forwarded through a firewall. For integrity of the transmitted data MACsec is proposed.

Meyer et al. introduce a network anomaly detection approach in cars extracting traffic characteristics using SDN and TSN [46]. Flow rules in SDN already define the behavior of every flow. As the safety-critical communication flows of automotive networks are specified in network designs, their behavior is determined. Therefore a rigorous configuration can enforce the expected network traffic behavior. The authors'

Network Anomaly Detection System (NADS) is used in the control plane as a controller application. Switches collect statistics of critical values and forward them to the SDN controller. The NADS inspects these values and can detect for example suspicious frame drops because a frame violated a flow meter configuration. Flow meter configurations are part of TSN which determine whether a frame is forwarded or dropped. For example, a flow meter would drop a frame if too little time has passed since the previous frame arrived. In case of a detection the NADS could report suspicious behavior to higher instances (e.g., a cloud defense center) or perform countermeasures by reconfiguring flow rules and TSN settings. In their case study they inspect different attack scenarios using different types of ingress control. The results show that network anomalies including DDoS is detected without generating falsely positive alarms.

IX. Relevant Conferences

Research findings on topics automotive networks, Software-Defined Networking and security are presented at technical conferences. The following listing shows some relevant conferences in no particular order:

- IEEE Vehicular Technology Conference (IEEE VTC) [3]
- IEEE Vehicular Networking Conference (IEEE VNC) [4]
- IEEE NetSoft [5]
- IEEE Local Computer Networks (IEEE LCN) [6]
- USENIX Security [7]
- IEEE Security & Privacy (IEEE S&P) [8]
- IEEE Communications and Network Security (IEEE CNS) [9]

X. Conclusion and Outlook

This work presented an initial overview about the the state-of-the-art of SDN-based security in automotive networks. Automotive Ethernet enables more bandwidth and in combination with SDN, and with Time-Sensitive Networking (TSN) real-time requirements can be met. SDN provides centralized network intelligence using a network controller. At the same time it opens up new attack surfaces due to loss of control over all forwarding devices in case of controller failure or faulty and malicious network applications.

Flow rules can isolate traffic. For example, critical safety systems can be made inaccessible by the infotainment system. Security solutions like TLS, MACsec, authentication and Anomaly Detection Systems (ADSs), to name a few, can prevent attackers from gaining access to the network.

However, further research is needed since open challenges remain.

[3]IEEE VTC: https://vtsociety.org/events/ (Accessed 14.02.2021)

[4]IEEE VNC: https://ieee-vnc.org/ (Accessed 14.02.2021)

[5]IEEE NetSoft: https://netsoft2021.ieee-netsoft.org/ (Accessed 14.02.2021)

[6]IEEE LCN: https://www.ieeelcn.org/ (Accessed 14.02.2021)

[7]USENIX Security: https://www.usenix.org/conferences (Accessed 14.02.2021)

[8]IEEE S&P: https://www.ieee-security.org/TC/SP-Index.html (Accessed 14.02.2021)

[9]IEEE CNS: https://ieee-cns.org (Accessed 14.02.2021)

References

[1] C. Buckl, A. Camek, G. Kainz, C. Simon, L. Mercep, H. Stähle, and A. Knoll, "The Software Car: Building ICT Architectures for Future Electric Vehicles," in *2012 IEEE International Electric Vehicle Conference*, Mar. 2012, pp. 1–8.

[2] fortiss GmbH, "The Software Car: Information and Communication Technology (ICT) as an Engine for the Electromobility of the Future," fortiss GmbH, Tech. Rep., Mar. 2011.

[3] K. Matheus and T. Königseder, *Automotive Ethernet*. Cambridge, United Kingdom: Cambridge University Press, Jan. 2015.

[4] IEEE 802.1 Working Group, "IEEE Standard for Local and Metropolitan Area Network–Bridges and Bridged Networks," IEEE, Standard Std 802.1Q-2018 (Revision of IEEE Std 802.1Q-2014), Jul. 2018.

[5] S. Checkoway, D. Mccoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive Experimental Analyses of Automotive Attack Surfaces," in *Proceedings of the 20th USENIX Security Symposium*, vol. 4. USENIX Association, Aug. 2011, pp. 77–92. [Online]. Available: http://www.autosec.org/pubs/cars-usenixsec2011.pdf

[6] C. Miller and C. Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle," *Black Hat USA*, vol. 2015, p. 91, 2015. [Online]. Available: https://ericberthomier.fr/IMG/pdf/remote_car_hacking.pdf

[7] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[8] W. Zimmermann and R. Schmidgall, *Bussysteme in der Fahrzeugtechnik*. Springer Fachmedien Wiesbaden, 2014.

[9] K. Reif, *Automobilelektronik*. Springer Fachmedien Wiesbaden, 2014.

[10] T. Steinbach, *Ethernet-basierte Fahrzeugnetzwerkarchitekturen für zukünftige Echtzeitsysteme im Automobil*. Wiesbaden: Springer Vieweg, Oct. 2018.

[11] J.-L. Scharbarg, M. Boyer, and C. Fraboul, "CAN-ethernet architectures for real-time applications," in *2005 IEEE Conference on Emerging Technologies and Factory Automation*. IEEE Press.

[12] A. Kern, D. Reinhard, T. Streichert, and J. Teich, "Gateway strategies for embedding of automotive CAN-frames into ethernet-packets and vice versa," in *Architecture of Computing Systems - ARCS 2011*. Springer Berlin Heidelberg, 2011, pp. 259–270.

[13] "IEEE Draft Standard for Ethernet Amendment 10: Media Access Control Parameters, Physical Layers and Management Parameters for 200 Gb/s and 400 Gb/s Operation," *IEEE P802.3bs/D3.3, July 2017*, pp. 1–393, 2017.

[14] A. Pretschner, M. Broy, I. H. Krüger, and T. Stauner, "Software Engineering for Automotive Systems: A Roadmap," in *2007 Future of Software Engineering*, ser. FOSE '07. Washington, DC, USA: IEEE Computer Society, May 2007, pp. 55–71. [Online]. Available: http://dx.doi.org/10.1109/FOSE.2007.22

[15] M. Cakir, T. Häckel, S. Reider, P. Meyer, F. Korf, and T. C. Schmidt, "A QoS Aware Approach to Service-Oriented Communication in Future Automotive Networks," in *2019 IEEE Vehicular Networking Conference (VNC)*. Piscataway, NJ, USA: IEEE Press, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/9062794

[16] International Organization for Standardization, "Road vehicles – Functional safety –(Part 1–10)," ISO, Standard ISO 26262, 2011.

[17] SAE J3061:2016-01-14, *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*, SAE, Aug. 2018. [Online]. Available: https://www.sae.org/standards/content/j3061/

[18] International Organization for Standardization, "Road vehicles – Cybersecurity engineering," ISO, Geneva, CH, Standard ISO/SAE DIS 21434, 2020.

[19] L. Schnieder and R. S. Hosse, "Entwurf angriffssicherer Systeme," in *Leitfaden Automotive Cybersecurity Engineering*. Springer Fachmedien Wiesbaden, 2018, pp. 13–24.

[20] B. Jelacic, D. Rosic, I. Lendak, M. Stanojevic, and S. Stoja, "STRIDE to a Secure Smart Grid in a Hybrid Cloud," in *Computer Security*. Springer International Publishing, dec 2017, pp. 77–90.

[21] DIN IEC 62443-3-3:2015-06, *Industrielle Kommunikationsnetze – IT-Sicherheit für Netze und Systeme – Teil 3-3: Systemanforderungen zur IT-Sicherheit und Security-Level*, DIN IEC, 2014.

[22] C. Schmittner, Z. Ma, C. Reyes, O. Dillinger, and P. Puschner, "Using SAE J3061 for Automotive Security Requirement Engineering," in

Lecture Notes in Computer Science. Springer International Publishing, 2016, pp. 157–170.

[23] M. Ihle and B. Glas, "Impact of demonstrated remote attacks on security of connected vehicles," in *Fahrerassistenzsysteme 2016*. Springer Fachmedien Wiesbaden, 2018, pp. 101–117.

[24] P. Krüger, *Architektur Intelligenter Verkehrssysteme (IVS)*. Springer Fachmedien Wiesbaden, 2015.

[25] J. Krimmling, *Ampelsteuerung*. Springer Fachmedien Wiesbaden, 2017.

[26] E. Schoitsch, C. Schmittner, Z. Ma, and T. Gruber, "The need for safety and cyber-security co-engineering and standardization for highly automated automotive vehicles," in *Advanced Microsystems for Automotive Applications 2015*. Springer International Publishing, jun 2015, pp. 251–261.

[27] L. Schnieder, *Schutz Kritischer Infrastrukturen im Verkehr*. Springer Fachmedien Wiesbaden, 2018.

[28] ISO/SAE DIS 21434:2020(E), ISO/SAE International, 2020.

[29] C. Corbett, E. Schoch, F. Kargl, and F. Preussner, "Automotive Ethernet: Security Opportunity or Challenge?" in *Sicherheit 2016 - Sicherheit, Schutz und Zuverlässigkeit*, M. Meier, D. Reinhardt, and S. Wendzel, Eds. Bonn: Gesellschaft für Informatik e.V., 2016, pp. 45–54.

[30] H. Ju, B. Jeon, D. Kim, B. Jung, and K. Jung, "Security Considerations for In-Vehicle Secure Communication," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*. IEEE Press, 2019, pp. 1404–1406.

[31] Q. Hu and F. Luo, "Review of Secure Communication Approaches for In-Vehicle Network," *International Journal of Automotive Technology*, vol. 19, no. 5, pp. 879–894, Sep. 2018.

[32] H. Mun, K. Han, and D. H. Lee, "Ensuring safety and security in can-based automotive embedded systems: A combination of design optimization and secure communication," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7078–7091, 2020.

[33] V. L. L. Thing and J. Wu, "Autonomous Vehicle Security: A Taxonomy of Attacks and Defences," in *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. Piscataway, NJ, USA: IEEE Press, Dec. 2016.

[34] H. Kwon, S. Lee, J. Choi, and B. Chung, "Mitigation mechanism against in-vehicle network intrusion by reconfiguring ecu and disabling attack packet," in *2018 International Conference on Information Technology (InCIT)*. IEEE Press, 2018, pp. 1–5.

[35] C. Lin and H. Yu, "Invited: Cooperation or Competition? Coexistence of Safety and Security in Next-Generation Ethernet-based Automotive Networks," in *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE Press, 2016, pp. 1–6.

[36] J. Lastinec and L. Hudec, "Comparative Analysis of TCP/IP Security Protocols for Use in Vehicle Communication," in *2016 17th International Carpathian Control Conference (ICCC)*. IEEE Press, 2016, pp. 429–433.

[37] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[38] Open Networking Foundation. [Online]. Available: https://www.opennetworking.org/

[39] M. Al-Zewairi, D. Suleiman, and S. Almajali, "An Experimental Software Defined Security Controller for Software Defined Network," in *2017 Fourth International Conference on Software Defined Systems (SDS)*. IEEE Press, 2017, pp. 32–36.

[40] I. Ahmad, S. Namal, M. Ylianttila, and A. Gurtov, "Security in Software Defined Networks: A Survey," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2317–2346, 2015.

[41] X. Xu and L. Hu, "A Software Defined Security Scheme Based on SDN Environment," in *2017 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*. IEEE Press, 2017, pp. 504–512.

[42] S. Krishnan and J. J. E. Oliver, "Mitigating DDoS Attacks in Software Defined Networks," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE Press, 2019, pp. 960–963.

[43] T. Häckel, P. Meyer, F. Korf, and T. C. Schmidt, "Software-Defined Networks Supporting Time-Sensitive In-Vehicular Communication," in *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. Piscataway, NJ, USA: IEEE Press, Apr. 2019, pp. 1–5.

[44] M. Haeberle, F. Heimgaertner, H. Loehr, N. Nayak, D. Grewe, S. Schildt, and M. Menth, "Softwarization of Automotive E/E Architectures: A Software-Defined Networking Approach," in *2020 IEEE Vehicular Networking Conference (VNC)*, 2020, pp. 1–8.

[45] AUTOSAR, "Specification of Secure Onboard Communication," AUTOSAR, Tech. Rep. 654, Dec. 2017.

[46] P. Meyer, T. Häckel, F. Korf, and T. C. Schmidt, "Network Anomaly Detection in Cars based on Time-Sensitive Ingress Control," in *2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall)*. Piscataway, NJ, USA: IEEE Press, Nov. 2020.