SEPAD – Security Evaluation Platform for Autonomous Driving

Daniel Zelle*, Roland Rieke*†, Christian Plappert*, Christoph Krauß*, Dmitry Levshun‡, and Andrey Chechulin‡
*Fraunhofer SIT, Darmstadt, Germany
†ITMO University, St. Petersburg, Russia
‡SPIIRAS, St. Petersburg, Russia

***Abstract*—The development and evaluation of security solutions for autonomous vehicles is a challenging task. Many researchers have no access to real vehicles to implement and test their solutions. In addition, vehicle E/E architectures of different brands or even model series of one car manufacturer differ significantly. Also, vehicles may be the source of physical hazards, e.g., an exploding airbag. To enable researchers to develop, implement, and evaluate new security solutions for autonomous vehicles, we propose a new security evaluation platform called SEPAD and a dedicated development process for testing security mechanisms with it. SEPAD allows to model realistic E/E architectures where the developed security solutions can be integrated and evaluated without causing safety risks for the researcher or other road users.**

***Index Terms*—automotive security, evaluation platform, autonomous driving, intrusion detection, trusted computing, secure in-vehicle protocols**

I. Introduction

Modern vehicles make intensive use of information and communication technologies (ICT) and are also connected to the outside world. On the one hand, ICT enables sophisticated Advanced Driver Assistance Systems (ADAS), infotainment services, and much more and is the enabler for autonomous driving. On the other hand, new possibilities for attacks emerge. Thus, appropriate security mechanisms must be deployed. However, the development and evaluation of security solutions for autonomous vehicles faces several challenges. First of all, a vehicle is expensive and requires special development tools and test equipment which also cost a lot of money. Second, vehicle electrical/electronic (E/E) architectures and many software components on Electronic Control Units (ECUs) are unknown (to the researcher) and may require intensive reverse engineering. Third, a vehicle represents only one single E/E architecture which is not sufficient to evaluate whether a security solutions is suitable for different E/E architectures. Forth, the vehicle may not support new technologies or even if the technology is present, it is not (publicly) documented and difficult to modify. Fifth, the modification of parts of the vehicle, either by integrating new security solutions or by performing attacks, may damage the vehicle. And finally, vehicles may be the source of physical hazards, e.g., high voltage, or airbag explosives, which may harm the life and limb of the security researcher. Thus, many security researchers use only theoretical models and estimations which may not resemble realistic autonomous vehicles.

To address this issue, a development and evaluation platform for autonomous driving is required. Vehicular autonomy has been categorized by the Society of Automotive Engineers (SAE) in six levels from Level 0 (no automation), via Levels 1–4 (increasing driver support by automation) to Level 5 (fully autonomous). Thus, the security evaluation platform should resemble currently available real vehicles as well as possible future vehicles with increasing levels of automation. Such a platform can be used for evaluating novel security mechanisms, e.g., hardware security solutions, security protocols, or mechanisms such as Intrusion Detection and Prevention Systems (IDPS).

In this paper we introduce SEPAD, a security evaluation platform for autonomous driving. Realistic E/E architectures can be modeled comprising typical components and protocols which are used in modern and upcoming autonomous vehicles. This includes, e.g., ECUs for charging electric vehicles using ISO 15118 [1], ADAS or Vehicle2X (V2X) connectivity but also modern communication technologies such as Automotive Ethernet. SEPAD is integrated in a scaled down model car. Newly developed security mechanisms can be easily integrated and evaluated in a realistic environment without any safety risks for the researcher. Also architectural design decisions can be modeled and evaluated in terms of security. In addition, we describe a dedicated development process for security mechanisms by using SEPAD.

The paper is organized as follows: First, we give a brief overview on related work in Section II. In Section III, we describe the requirements for SEPAD. These are derived from an analysis of current attacks, promising approaches for security solutions, and envisions automotive technologies and protocols. Then, we describe the modular architecture, components, and supported technologies and protocols of SEPAD in Section IV. We describe our proposed security development process in Section V. Finally, we conclude our paper in Section V and give an outlook on our future work.

II. Related Work

For research in autonomous driving, vehicles are often extended with specific sensors and actors. For example, in [2] or [3] the cars are extended with lidar sensors and cameras as

well as control units for steering wheel, gas and brake paddle. However, for security research in this area, in particular when attacking the system in order to evaluate IDPS solutions, the use of a real vehicle is too dangerous. It has been shown that model cars can be used as alternative platforms to evaluate autonomous driving algorithms. In [4] such a model car is described which uses a core computational component directly connected to all relevant sensors. Technology of autonomous driving is also tested in competitions like Carolo-Cup [5], where automotive data- and time-triggered frameworks or the robot operating system [6] are used as base platforms. These development frameworks are used for prototyping but not in real vehicles.

For cyber security research, [7] proposed a remote controlled car with one central component. This platform does not address complex in-vehicle networks which are typically used in autonomous vehicles. An architecture for autonomous cars combining classic automotive bus systems like Controller Area Network (CAN) and FlexRay as well as Ethernet communication is described in [8].

The Toyota InfoTechnology Center presents the platform PASTA (Portable Automotive Security Testbed with Adaptability) [9] based on non-proprietary hardware as a platform for research, education, and information sharing of vehicle cyber security. The platform allows to perform attacks on typical physical attack surfaces, however, it does not include sensors or interactions with the physical world. Thus, influences and impacts to the real world cannot be observed.

Similarly Fowler et al. [10] introduced the concept of a test platform for CAN and On-board diagnostics (OBD) based security testing in a hardware-in-the-loop test environment. Again, no sensors are influenced by the environment and also there are no physical impacts. Moreover, the testbed concentrates only on CAN bus vulnerabilities.

Contrary to the hardware-in-the-loop approach, Mundhenk et al. [11] presented a software simulation to test security protocols in vehicle networks. An early sketch of our concept has been presented as an extended abstract in [12].

III. Requirements Analysis

In this section, we identify the requirements for our security evaluation platform for autonomous driving. In Section III-A, we describe the attacker model which must be supported by our platform. In Sections III-B, III-C, and III-D, we describe the different types of security mechanisms which can be implemented and evaluated by our platform. We distinguish between security solutions which prevent attacks, enable the detection of attacks, and mechanisms to respond to attacks. Finally, we discuss the functional requirements which must be supported by the platform in Section III-E.

A. Attacker Model

In this section, we describe the attacker model which must be supported by our platform. This includes the different types and levels of attackers as well as the typical attack surface. Similar to [13], we propose to distinguish attackers by their type and level. The type (in the range between 0 and 4) describes the type of access an attacker has to the vehicle (see Table I) and the level (in the range between 1 and 3) describes the capabilities and resources an attacker has (see Table II).

TABLE I: Types of Attackers

Type	Description
0	No access to autonomous vehicle elements and network, only indirect action (examples: social engineering methods, usage of environmental analysis interfaces to deceive the control system)
1	Indirect access to autonomous vehicle elements and network (example: cellular communication)
2	Indirect access to autonomous vehicle elements and network, while being within a certain proximity of the vehicle (examples: WLAN and Bluetooth)
3	Direct physical access to autonomous vehicle elements and network (examples: OBD-2 port, USB)
4	Full access to autonomous vehicle elements and network (examples: compromised ECUs, direct access to the CAN bus)

TABLE II: Levels of Attackers

Level	Description
1	Attacker has insufficient knowledge about autonomous vehicle elements and network and can use only wide-spread software tools and exploits only well-known vulnerabilities (examples: eavesdropping, Denial of Service (DoS) Attacks, exploiting known software vulnerabilities)
2	Attacker has detailed information about autonomous vehicle elements and network and can use specialized attacking tools and exploit unknown vulnerabilities (examples: injection of specific CAN messages, manipulation of an ECU's firmware, direct attacks on sensors (e.g., lidar))
3	Level 2 attackers with almost unlimited resources (examples: readout of confidential data (e.g., cryptographic keys) maybe even using side-channel attacks, attacks on used (weak) cryptographic algorithms, exchange of ECUs)

In our model, the structure of types and levels are hierarchical. This means that an attacker with a certain type is able to perform any attack which is possible for an attacker of the same type but lower level. It also means that an attacker of higher type is able to perform any attack which is possible for an attacker of lower type but the same or lower level.

In Table III, we list the attack surface on modern vehicles with attacks points, possible attacks and the associated attacker type and level. In addition, we will give some examples of security mechanisms to counter the attacks. Please note that we show only attacks for the highest attacker level and type and do not consider the hierarchical structure by including all other inherited attacks.

B. Prevention

Prevention mechanisms are carried out to prevent the successful execution of an attack. In the following, we describe two important aspects which must be supported by SEPAD: trust anchor technologies, security protocols, and mitigation approaches.

TABLE III: Attack Surface of Modern Vehicles

Attacker Type	Attacker Level	Attack Point	Attack	Possible Security Mechanism
0	1	Sensor Lidar	Relay/Spoof signal attack [14]	Redundancy, random probing
0	2	Sensor Camera	AI manipulation [15]	Robust image recognition systems, sensor fusion
1	1	Cellular(Internet connection)	Distributed Denial of Service	Intrusion detection system
1	2		Vulnerability exploitation	Intrusion prevention system
1	2	Cellular	Spoofing [16]	End to end authentication
1	2		SIP registration hijacking [16]	End to end authentication
1	2		Eavesdropping [16]	End to end security
1	2	Cellular (system update)	Man in the Middle	Improve protocols
1	3		Cryptanalysis of encrypted protocols	Usage of reliable protocols
2	1	WLAN	Eavesdropping	Encrypted communication
2	1	DAB	Arbitrary Code execution [17]	Secure Coding, isolation mechanisms
2	1	FM/AM	Inject arbitrary RDS messages to re-route car [18]	Authentication and integrity-protection of messages
2	1	GPS	Location Spoofing [19]	Plausibility checks (signal strength, direction)
2	1	RKE (RFID/NFC)	Relay Attack [20]	Distance Bounding Technologies
2	1		Protocol Flaw [21]	Improve protocol
2	2	TPMS	spoofing attacks [22]	Plausibility/consistency checks, IDS
2	2		Battery drain attacks [22]	Plausibility/consistency checks, IDS
2	2		Vehicle tracking [22]	Pseudonymization of identifiers
2	2	V2X	Denial of Service	Intrusion detection
2	3		Authentication Attacks	Efficient revocation, secure key storage
2	2		Tracking	Pseudonym changing scheme
2	2	Bluetooth	Injection attacks due to flawed protocol stack implementation [23]	Secure Coding, isolation mechanisms
2	1	PLC	Eavesdropping [24]	Improve protocol (e.g., secure end-to-end channels, pseudonym identifiers)
3	2		Protocol flaws [1]	Improve protocol (e.g., secure end-to-end channels, time synchronization, instantiation of trust anchors)
3	1	RFID/NFC key	Cloning the UID	Multi-factor authentication
3	2	USB / SD-Card / CD / Other Storage Media	Flash compromised firmware through USB/CD [23], [25]	Secure update mechanism (Authenticated and integrity-protected binaries, downgrade protection)
4	2	LIN	Inject false headers/responses [26]	Authenticated messages [26]
4	2	FlexRay	Denial of Service [27]	Intrusion detection
4	2		Injection Attack [27]	Authenticated messages
4	2	Most	Denial of Service [27]	Intrusion detection
4	2	Automotive Ethernet	Man in the Middle attacks [28]	End-2-end communication channels
4	2		Replay Attack [28]	Introducing freshness values
4	2		Denial of Service [28]	isolation mechanisms, redundancy,
4	2		Sniffing Attack [28]	Encipherment, Gateways
4	2	CAN/CAN-FD(OBD)	Denial of Service [29]	Detect anomalies [30]–[34]
4	3		ECU impersonation [29]	Physical fingerprinting [35], [36]

1) Trust Anchor Technologies: Attacks targeting to read out or manipulate flash memory in order to extract keys or circumvent software-based security mechanisms (c.f., Table III, e.g., [23], [25]) are typically mitigated by introducing different kinds of hardware trust anchors into the system. These trust anchors allow to instantiate a trusted domain isolated from the probably compromised rest of the system.

In the trusted computing context three different roots of trust are defined that may be combined to instantiate a trusted domain with certain security guarantees. These roots are the root of trust for storage, measurement and reporting. A root of trust for storage establishes a secure environment, a so-called shielded location, to store and process security sensitive information. A root of trust for measurement allows to trustworthy measure the platform's state, e.g., by calculating hash values over the going to be executed code. Finally, a root of trust for reporting establishes a mechanism to provide a remote party with authentic and integrity-protected measurement values. This mechanism is called remote attestation and requires both the root of trust for measurement to obtain trustworthy measurements as well as the root of trust for storage to aggregate and store all measurement values throughout the boot process.

Instantiations of these trust anchors come in a variety of different forms with different security capabilities. This reaches from instantiations only requiring a small immutable and unreadable code area to store a secret value like Device Identifier Composition Engine (DICE) [37] over processor extensions like ARM's TrustZone [38] or Intel's Software Guard Extensions (SGX) [39] to dedicated chips like the Trusted Platform Module (TPM) [40] or various Hardware Security Modules (HSMs).

Especially in the automotive domain, HSMs like Secure Hardware Extension (SHE) are currently utilized. Despite SHE offers only a root of trust for storage to instantiate a shielded location, it successfully mitigates offline attacks where the goal is to extract keys from the flash memory since SHE securely stores cryptographic keys and executes cryptographic

operations like encryption and Message Authentication Code (MAC) calculations isolated from the host system.

Next to SHE also other trust anchor technologies are considered by Original Equipment Manufacturers (OEMs) and Tier1 suppliers to be included into future E/E architectures like DICE and TPM. While DICE is especially suited for small ECUs and is not dependent on additional hardware, it provides different security and performance guarantees than SHE or a even a full-fledged TPM.

SEPAD must support the evaluation of the interaction and impact of these different trust anchor technologies and corresponding protocols in a realistic heterogeneous E/E architecture without causing physical harm or high expenses while allowing easy adaptations to new requirements. Hardware can be easily plugged in and out and resulting impacts on the architecture, e.g., in regard to additional introduced delays, can be examined.

2) Security Protocols: SEPAD must provide support for implementing and evaluating different security protocols in realistic E/E architectures in close to real conditions. The effectiveness against different types of attacks (cf. Section III-A) and efficiency (e.g., in terms of introduced additional communication overhead) of these protocols must be analyzable. Especially, the analysis of the impact of integrating security protocols in different automotive bus systems as well as the use of different hardware trust anchors should be analyzable.

Typical security protocols include MAC-based authentication schemes such as AUTOSAR's Secure Onboard Communication (SecOC) but also more sophisticated protocols such as Internet Protocol Security (IPsec), which has been proposed to use in automotive Ethernet networks. SEPAD also supports the communication with external entities, e.g., OEM backend servers or charge points for electric vehicles. Here, typically Transport Layer Security (TLS) is used to secure the communication.

In addition, SEPAD must also support the analysis of unprotected E/E architectures to analyze the impact of successful attacks on different types of E/E architectures. It is expected that also modern and autonomous vehicles will still use legacy technologies such as CAN or LIN. Thus, it must be possible to evaluate approaches to separate (sub)networks securely from each other where some (sub)networks do not implement any security mechanisms.

3) Mitigation: While mitigation approaches do not prevent the attack itself, they prevent the possible impact of an attack. SEPAD must be designed to support the integration and evaluation of different mitigation approaches. This includes especially separation approaches. For example, an E/E architecture can be designed to tolerate attacks on non-critical parts, e.g., infotainment, but still preventing successful attacks on safety-critical parts such as the braking system. Separation approaches are also applicable on host-systems. For example, modern vehicles consolidate (safety-critical and not safety-critical) functionalities on one ECU which have previously been realized with multiple different ECUs. By separating these functionalities in different compartments using technologies such as separation kernel, virtualization etc., attacks could still be executed within one compartment but their impact is limited to only this compartment and cannot spread to other compartments with safety-critical functionalities.

C. Detection

In addition to evaluating security mechanisms which try to prevent attacks, SEPAD must also support different detection mechanisms. In the following, we describe three types of detection mechanisms which must be supported by SEPAD.

1) Intrusion Detection Systems: Intrusion detection systems (IDS) which monitor and analyse host systems or network traffic are well know and used in classical networks, e.g., corporate networks. In the last years, many researches developed IDS solutions for analysing the communication within vehicles. A major challenge for these approaches are the resource constraints of the ECUs in a vehicle. Thus, SEPAD must support the integration of IDS approaches to evaluate their effectiveness and efficiency in typical automotive E/E architectures.

Automotive IDS research did not yet consider the novel designs of E/E architectures and how it affects the different detection techniques or the placement of the IDS. For example, an IDS placed at a gateway as a correlation sensor as proposed in [41] would allow to observe and correlate sensor values from different networks. However, a recent review of work in this area [42] shows that the majority of papers do not analyze multiple data sources for attack detection. Most papers describe solutions targeting specific well-known types of attacks such as man-in-middle or denial-of-service. Because new attack techniques are discovered, and attacks are deployed long after a vehicle has been sold, it is imperative to detect new threats and address vulnerabilities affecting released vehicles [43]. For example, bus-off attacks [44] were not known until recently. Such novel attacks can only be found by realistic evaluation models of novel E/E architectures which must be supported by SEPAD.

SEPAD can be used to address specific challenges in autonomous vehicles. For example, suitable detection techniques can be developed which make use of the higher resources of certain ECUs (e.g., artificial intelligence based computations in edge computing nodes) and utilize the changes of in-vehicle networks by retrieving data from multiple data sources. This challenge can be addressed by configuration options of SEPAD.

2) Attack Detection by Physical Measurements: In contrast to traditional networks, in the automotive domain there are specific physical properties of communication media that could be used by detection mechanisms for side channel analysis. The work on detection of ECU impersonating attacks such as [35], [36] in most cases uses some kind of physical fingerprinting by voltage or timing analysis with specific hardware. This work seeks to mitigate the general problems of missing authenticity measures in CAN bus design and thus is complementary to intrusion detection by analyzing anomalies

in payload or message frequency. SEPAD should also support such approaches.

3) Attestation: The trust anchor technologies described in Section III-B1 may be used to instantiate various attestation mechanisms and protocols that detect the integrity of the vehicle [45], [46]. SEPAD must support such approaches.

D. Response

SEPAD must support the evaluation of response mechanisms which are usually executed after detecting an attack.

1) Reaction: After detecting an attack, an appropriate reaction must be performed. In classical computer networks, an incident is reported to the user together with suggestions about possible actions to take. However, in a vehicle the driver is unlikely to have the technical knowledge to react appropriately within a very short time [41]. If an autonomous vehicle is attacked, the impact will be much higher as the driver will not be able to take control in time, if at all [43]. Therefore, study and design of appropriate alerting methods and security strategies [47] adapted to automotive systems and dynamic risk management response systems similar to such systems in critical infrastructures [48] is a further research direction that requires investigation.

2) Recovery: Recovery is a special instance of reaction which enables the system to ”heal” from the attack. In case crucial components are not responding probably anymore because they hang up or are compromised by an attacker, mechanisms of recovery must be instantiated to change their status back to the operating condition. Since the components are not responding to any sent commands from management entities, they need an intrinsic secure and fail-safe mechanism to boot in a state where recovery is possible again.

One promising solution to solve this issue are watchdog timers that are able to reset the platform back to an operating state where it is again responding to management commands. The device is protected by an “execution latch” (authenticated watchdog timer) where execution is periodically granted by authenticated commands of an management service [49]. As soon as these deferral tickets are absent for a defined time frame, the platform will reset itself to an recoverable state.

E. Functional Requirements

SEPAD must support different E/E architectures for autonomous vehicles. These architectures include both automotive hardware (ECUs, sensors, actors, gateways, bus networks like CAN, Local Interconnect Network (LIN), and Automotive Ethernet) and software (middleware like AUTOSAR or protocols like Diagnostics over Internet Protocol (DoIP)). A modular approach is required to enable the analysis of security concepts in different architectures. SEPAD must also be easily configurable and extendable, e.g., by using open source software projects. As a central aspect, SEPAD must enable the integration and evaluation of security solutions without causing safety risks for the researcher or other road users. Thus, all functionalities which may be a threat to life and limb must be exchanged, e.g., with a simulation environment which enables a reliable statement about the real consequences.

IV. Architecture

SEPAD provides a basic set of components, communication technologies and protocols integrated into a model car. The components can be arranged in different E/E architectures, e.g., by using gateways to group several ECUs into domains.

A. Components

Core components of the system enable autonomous driving by monitoring of the environment and execution of actions. All (currently supported) components are shown in Figure 1. Steering (steering unit), speed (engine control unit), and lights (light unit) are examples for actors in the autonomous driving domain control. Ultrasonic and lidar distance sensors and cameras observe the environment. Gyroscope, an accelerometer, and a wheel rotation sensor monitor the state of the vehicle.

Furthermore, our model car is an electric vehicle which is equipped with a battery, a battery management system, a charging controller, and a metering system.

The infotainment domain consists of the instrument cluster and the head unit as interfaces to driver and passengers. A sound system allows to play music and the telematic unit allowing communication of the vehicle with the Internet, other vehicles, or mobile devices (e.g., smartphones). For the sake of simplicity, we also integrated seat and door control into this domain.

Components of the evaluation domain are connected to the networks of all other domains and thus can be used to monitor and influence data flows between and within these domains. The logger component of the evaluation domain is used to monitor and store data flows for reproducing test results or gather training data for an IDPS. With the attack simulation component data flows can be influenced to simulate various attacks, e.g., by modifying, replaying, relaying, delaying or intercepting messages. The impact of these and other definable attacks can be analyzed to develop sophisticated detection and mitigation techniques.

B. Communication Technologies

As depicted in Figure 2, SEPAD supports various automotive communication technologies both in the in-vehicle network as well as to external entities. In particular, SEPAD utilizes Automotive Ethernet which is expected to be the predominant technology in autonomous vehicles as well as CAN and CAN FD to support also legacy ECUs. Powerline communication is used to support electric charging via ISO15118 with a charge point. For external communication, SEPAD implements Bluetooth, WLAN and cellular communication to connect to various backend systems and external devices like smartphones.

C. Communication Protocols

Various communication protocols are implemented in modern vehicles. In the first iteration of our platform, we integrate the publicly available protocols shown in Table IV. In particular, we use DoIP [50] for error diagnose of vehicles

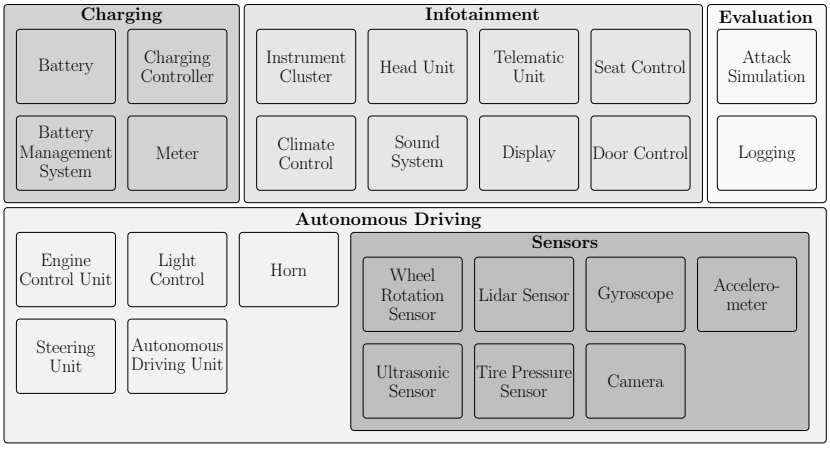

Fig. 1: Component Architecture

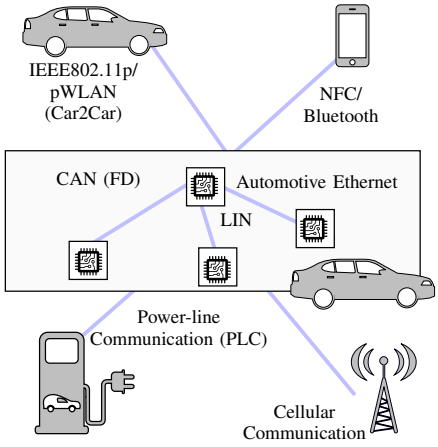

Fig. 2: Communication Technologies

and SOME/IP (-SD) [51], [52] which implement publish-subscribe services for the in-vehicle networks. Moreover, we integrate the audio video bridging protocol (AVB) [53] which is commonly used in the automotive domain to transmit audio and video streams with real time constraints. For the external communication we integrate remote vehicle interaction protocol (RVI) [54] developed by Jaguar Land Rover and IEEE 802.11p vehicular adhoc network for Vehicle2Vehicle (V2V) communication. Electric charging use cases like authentication and payment are covered by integrating ISO 15118.

TABLE IV: Automotive Protocols

Protocol	Application
SOME/IP	Service oriented internal communication
DoIP	Vehicle diagnose for repair and maintenance
AVB	Audio/Video Data Transfer
RVI	Connection to backend and mobile devices
ISO 15118	Charging Protocol
IEEE 802.11p	Vehicular Ad Hoc Network (VANET)

Fig. 3: Development Cycle for Security Components

V. Development Process for Security Mechanisms

SEPAD can be used to enhance a secure development process which can result in innovative security mechanisms. The process is divided into five steps which are shown in Figure 3: *Simulate Attacks*, *Analyze Impact*, *Evaluate Risk*, *Adapt Security Strategy*, and *Evaluate in Operation*.

Simulate Attacks uses the attack simulation functionality of SEPAD to simulate the relevant attacks of the assumed attacker (cf. Section III-A). Next, *Analyze Impact* is used to evaluate the impact of the attack. Since SEPAD is integrated in a model car without any hazardous functionalities (e.g., exploding airbags), the life and limb of the researchers is not at stake. In a basic analysis, a researcher can simply observe the behaviour of the model car. For example, the researcher can observe whether the model car slows down, stops, or makes a turn after injecting messages or performing a Denial of Service (DoS) attack. SEPAD supports support also more sophisticated analyses by logging all events. The researcher can use the log files of the vehicle to identify the messages which were resulted in the impact. This enables a deep understanding of an attack and its impact on every part of the vehicle. Using this information from SEPAD, the researcher can evaluate the risk for the vehicle, the passengers, external persons, and the environment in the *Evaluate Risk* step. In addition to the possible impact, SEPAD gives information about the likelihood of an attack by providing information about the preconditions for an attack or how easy it is to execute in realistic environments. For example, it can be analyzed whether an attacker can be executed remotely or only via physical access. Obviously, the first results in a much higher risk. Using this information, the researcher can use for example the method described in [55] to determine the risk in an automotive scenario. Based on the identified risk, the security strategy can be adapted in the *Adapt Security Strategy* step by adapting the security architecture and using appropriate security mechanisms (cf. Table III). In this phase, the strategies are designed, theoretically analyzed,

and implemented in the SEPAD model car. In the *Evaluate in Operation* step, an evaluation of the operating car is performed without implementing attacks to identify any possible issues introduced by the security mechanisms. For example, functional problems which are the result of a high latency due to added cryptographic calculations can be discovered. After successfully finishing this step, the process starts over again to evaluate whether the adapted security strategy is suitable to prevent the risk of attacks.

VI. Conclusion and Future Work

In this paper we present a novel security evaluation platform for autonomous driving (SEPAD) and a dedicated development process for testing security mechanisms with it. SEPAD allows to model realistic E/E architectures and tackles shortcomings of current and future security development processes for such architectures. In specific, it provides a close to real testing environment for security researchers and developers lacking access to real cars and serves as a sandbox to test security technologies without risking expensive automotive hardware or causing physical harm to researchers and developers and other road users.

SEPAD enables testing and evaluation of security concepts for autonomous vehicles as well as possible impacts of successful attacks on cyber-physical systems of the vehicle or its environment. The evaluation platform is not limited to specific E/E architectures of a specific car manufacturer and can be easily modified or extended.

SEPAD's design and architectural features were derived from a comprehensive requirements analysis. First, we analyzed the current automotive attack surface and defined an attacker model that must be supported by our platform. Based on this model and distinct threats, we examine different promising types of security mechanisms that target the prevention, detection, or response to the described attacks. In addition, we analyzed technologies that are currently deployed in the automotive environment and derived functional requirements that our platform needs to support. This includes hardware (e.g., ECUs, sensors and actors) and software components (e.g., implementations of operating systems, automotive middlewares and automotive protocols). We plan to make the architectural design and software components publicly available in order to enable the automotive industry and other researchers to reproduce the platform and results.

Accompanying our platform, we describe a dedicated security development process for SEPAD which allows to iteratively improve security strategies by consistently adapting, testing and evaluating new approaches.

As ongoing and future work, we will intensively use SEPAD to develop, implement, evaluate security mechanisms for autonomous vehicles. In particular, we will develop detection techniques using an IDS. IDS approaches for autonomous vehicular systems may differ from traditional solutions because, for example, they can exploit specific properties of autonomous vehicles, e.g., special architectures of the E/E system but also possibly higher resources of specific ECUs, which could be used for more sophisticated machine learning approaches. Also, the best strategy to respond to a running attack in a vehicle will be examined. Impacts when improving adaptive risk mitigation and response solutions, e.g., extended stopping distances or changed steering behavior, need to be considered and continuously tested with SEPAD.

Acknowledgment

This research work has been funded by the German Federal Ministry of Education and Research and the Hessen State Ministry for Higher Education, Research and the Arts within their support of the National Research Center for Applied Cybersecurity ATHENE, by the German Federal Ministry of Education and Research in the context of the project VITAF (ID 16KIS0835) and partially by RFBR project number 19-29-06099.

References

[1] K. Bao, H. Valev, M. Wagner, and H. Schmeck, "A threat analysis of the vehicle-to-grid charging protocol iso 15118," *Computer Science-Research and Development*, vol. 33, no. 1-2, pp. 3–12, 2018.

[2] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*, June 2011, pp. 163–168.

[3] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi, "Towards a viable autonomous driving research platform," *Proceedings of the 2013 IEEE Intelligent Vehicles Symposium, pp. 763-770*, pp. 763–770, June 2013.

[4] F. Bormann, E. Braune, and M. Spitzner, "The c2000 autonomous model car," in *4th European Education and Research Conference (EDERC 2010)*, Dec 2010, pp. 200–204.

[5] M. Nolte, T. Form, S. Ernst, R. Graubohm, and M. Maurer, "The carolocup student competition: Involving students with automated driving," in *12th European Workshop on Microelectronics Education, EWME 2018, Braunschweig, Germany, September 24-26, 2018*, 2018, pp. 95–99.

[6] A. Hellmund, S. Wirges, Ö. S. Tas, C. Bandera, and N. O. Salscheider, "Robot operating system: A modular software framework for automated driving," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2016, pp. 1564–1570.

[7] J. Axelsson, A. Kobetski, Z. Ni, S. Zhang, and E. Johansson, "Moped: A mobile open platform for experimental design of cyber-physical systems," in *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*. IEEE, 2014, pp. 423–430.

[8] B. Zheng, H. Liang, Q. Zhu, H. Yu, and C. Lin, "Next generation automotive architecture modeling and exploration for autonomous driving," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, July 2016, pp. 53–58.

[9] T. Toyama, T. Yoshida, H. Oguma, and T. Matsumoto, "PASTA: Portable Automotive Security Testbed with Adaptability," Black Hat Europ, Tech. Rep., 2018, https://i.blackhat.com/eu-18/Wed-Dec-5/eu-18-Toyama-PASTA-Portable-Automotive-Security-Testbed-with-Adaptability[1].pdf.

[10] A. Tomlinson, J. Bryans, and S. A. Shaikh, "Towards viable intrusion detection methods for the automotive controller area network," in *2nd ACM Computer Science in Cars Symposium*, 2018.

[11] P. Mundhenk, A. Mrowca, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Open source model and simulator for real-time performance analysis of automotive network security," *Acm Sigbed Review*, vol. 13, no. 3, pp. 8–13, 2016.

[12] D. Zelle, R. Rieke, and C. Krauß, "Security test platform for autonomous driving," 3. ACM COMPUTER SCIENCE IN CARS SYMPOSIUM (CSCS 2019), Tech. Rep., 2019.

[13] D. Levshun, I. Kotenko, and A. Chechulin, "The integrated model of secure cyber-physical systems for their design and verification," in *Intelligent Distributed Computing XIII*. Springer International Publishing, 2020, pp. 333–343.

[14] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[15] C. Sitawarin, A. N. Bhagoji, A. Mosenia, M. Chiang, and P. Mittal, "DARTS: deceiving autonomous cars with toxic signs," *CoRR*, vol. abs/1802.06430, 2018.

[16] Y. Park and T. Park, "A survey of security threats on 4g networks," in *2007 IEEE Globecom workshops*. IEEE, 2007, pp. 1–6.

[17] A. Davis, "Broadcasting your attack DAB security," Black Hat USA, Tech. Rep., 2015, https://www.blackhat.com/us-15/briefings.html#broadcasting-your-attack-security-testing-dab-radio-in-cars.

[18] A. Barisani and B. Daniele, "Unusual car navigation tricks: Injecting rds-tmc traffic information signals," in *Proceedings of the CanSecWest Conference*, 2007.

[19] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful gps spoofing attacks," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 75–86.

[20] A. Francillon, B. Danev, and S. Capkun, "Relay attacks on passive keyless entry and start systems in modern cars," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2011.

[21] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès, "Lock it and still lose iton the (in) security of automotive remote keyless entry systems," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016.

[22] R. M. Ishtiaq Roufa, H. Mustafaa, S. O. Travis Taylora, W. Xua, M. Gruteserb, W. Trappeb, and I. Seskarb, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *19th USENIX Security Symposium, Washington DC*, 2010, pp. 11–13.

[23] 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 Conference on Security*, ser. SEC'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 6–6.

[24] R. Baker and I. Martinovic, "Losing the car keys: Wireless phy-layer insecurity in {EV} charging," in *28th {USENIX} Security Symposium ({USENIX} Security 19)*, 2019, pp. 407–424.

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

[26] J. Takahashi, Y. Aragane, T. Miyazawa, H. Fuji, H. Yamashita, K. Hayakawa, S. Ukai, and H. Hayakawa, "Automotive attacks and countermeasures on lin-bus," *Journal of Information Processing*, vol. 25, pp. 220–228, 2017.

[27] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*. Bochum, 2004.

[28] T. Kiravuo, M. Sarela, and J. Manner, "A survey of ethernet lan security," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1477–1491, 2013.

[29] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*. IEEE, 2010, pp. 447–462.

[30] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*, Dec 2015, pp. 45–49.

[31] H. Song, H. Kim, and H. Kim, *Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network*. IEEE Computer Society, 3 2016, vol. 2016-March, pp. 63–68.

[32] Z. Wei, Y. Yang, Y. Rehana, Y. Wu, J. Weng, and R. H. Deng, *IoVShield: An Efficient Vehicular Intrusion Detection System for Self-driving (Short Paper)*. Cham: Springer International Publishing, 2017, pp. 638–647.

[33] I. Berger, R. Rieke, M. Kolomeets, A. Chechulin, and I. Kotenko, "Comparative study of machine learning methods for in-vehicle intrusion detection," in *Computer Security. ESORICS 2018 International Workshops, CyberICPS 2018 and SECPRE 2018, Revised Selected Papers*, ser. Lecture Notes in Computer Science, vol. 11387. Cham: Springer, 2019, pp. 85–101.

[34] Y. Chevalier, R. Rieke, F. Fenzl, A. Chechulin, and I. V. Kotenko, "Ecu-secure: Characteristic functions for in-vehicle intrusion detection," in *Intelligent Distributed Computing XIII*, ser. Studies in Computational Intelligence, vol. 868. Cham: Springer, 2020, pp. 495–504.

[35] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "Voltageids: Low-level communication characteristics for automotive intrusion detection system," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2114–2129, Aug 2018.

[36] K. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, T. Holz and S. Savage, Eds. USENIX Association, 2016, pp. 911–927.

[37] Trusted Computing Group, "Hardware Requirements for a Device Identifier Composition Engine," https://trustedcomputinggroup.org/wp-content/uploads/Hardware-Requirements-for-Device-Identifier-Composition-Engine-r78_For-Publication.pdf, 03 2018.

[38] Arm Limited, "Introducing Arm TrustZone," https://developer.arm.com/technologies/trustzone, 03 2019.

[39] Intel Corporation, "Intel Software Guards Extensions," https://software.intel.com/en-us/SGX, 10 2019.

[40] Trusted Computing Group, "TPM 2.0 Library Specification," https://trustedcomputinggroup.org/resource/tpm-library-specification/, 09 2016.

[41] M. Müter, A. Groll, and F. C. Freiling, "A structured approach to anomaly detection for in-vehicle networks," *2010 Sixth International Conference on Information Assurance and Security*, pp. 92–98, 2010.

[42] 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. 21 266–21 289, 2019.

[43] Daimler, Aptiv, Audi, Baidu, BMW, Continental, Fiat Chrysler Automobiles, HERE, Infineon, Intel, and Volkswagen. (2019) Safety first for automated driving. [Online]. Available: https://www.daimler.com/dokumente/innovation/sonstiges/safety-first-for-automated-driving.pdf

[44] W. Choi, K. Joo, H. Jo, M. Park, and D. Lee, "Voltageids: Low-level communication characteristics for automotive intrusion detection system," *IEEE Transactions on Information Forensics and Security*, 3 2018.

[45] M. Khodari, A. Rawat, M. Asplund, and A. Gurtov, "Decentralized firmware attestation for in-vehicle networks," in *Proceedings of the 5th on Cyber-Physical System Security Workshop*, ser. CPSS '19. New York, NY, USA: ACM, 2019, pp. 47–56.

[46] H. Oguma, A. Yoshioka, M. Nishikawa, R. Shigetomi, A. Otsuka, and H. Imai, "New attestation based security architecture for in-vehicle communication," in *IEEE GLOBECOM 2008 - 2008 IEEE Global Telecommunications Conference*, Nov 2008, pp. 1–6.

[47] R. Rieke, J. Schütte, and A. Hutchison, "Architecting a security strategy measurement and management system," in *Proceedings of the Workshop on Model-Driven Security*, ser. MDsec '12. New York, NY, USA: ACM, 2012, pp. 2:1–2:6.

[48] G. Granadillo, S. Dubus, A. Motzek, J. Garcia-Alfaro, E. Alvarez, M. Merialdo, S. PAPILLON, and H. Debar, "A dynamic risk management response system to handle threats in scada systems," *Future Generation Computer Systems*, 06 2017.

[49] P. England, R. Aigner, A. Marochko, D. Mattoon, R. Spiger, and S. Thom, "Cyber-resilient platform requirements," Microsoft, Tech. Rep., August 2017. [Online]. Available: https://www.microsoft.com/en-us/research/publication/cyber-resilient-platform-requirements/

[50] AUTOSAR. (2018) Specification of diagnostic over ip - classic platform 4.4.0. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_DiagnosticOverIP.pdf

[51] ——. (2018) Specification on SOME/IP transport protocol - classic platform 4.4.0. [Online]. Available: https://www.autosar.org/fileadmin/Releases_TEMP/Classic_Platform_4.4.0/Communication.zip

[52] ——. (2018) Specification of service discovery - classic platform 4.4.0. [Online]. Available: https://www.autosar.org/fileadmin/Releases_TEMP/Classic_Platform_4.4.0/Communication.zip

[53] AVnu Automotive Technical Working Group. (2016) Automotive ethernet avb functional and interoperability specification revision 1.5. [Online]. Available: https://avnu.org/wp-content/uploads/2014/05/Automotive-Ethernet-AVB-Func-Interop-Spec-v1.5-Public.pdf

[54] Jaguar Land Rover. (2014) Remote vehicle interaction (rvi). [Online]. Available: https://github.com/GENIVI/rvi_core

[55] G. Macher, H. Sporer, R. Berlach, E. Armengaud, and C. Kreiner, "Sahara: A security-aware hazard and risk analysis method," *2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 621–624, 2015.