Attack Surface Assessment for Cybersecurity Engineering in the Automotive Domain

Christian Plappert, Daniel Zelle, Henry Gadacz, Roland Rieke, Dirk Scheuermann, Christoph Krauß
Fraunhofer Institute for Secure Information Technology
Darmstadt, Germany
{christian.plappert, daniel.zelle, henry.gadacz, roland.rieke, dirk.scheuermann, christoph.krauss}@sit.fraunhofer.de

***Abstract*—Connected smart cars enable new attacks that may have serious consequences. Thus, the development of new cars must follow a cybersecurity engineering process as defined for example in ISO/SAE 21434. A central part of such a process is the threat and risk assessment including an attack feasibility rating. In this paper, we present an attack surface assessment with focus on the attack feasibility rating compliant to ISO/SAE 21434. We introduce a reference architecture with assets constituting the attack surface, the attack feasibility rating for these assets, and the application of this rating on typical use cases. The attack feasibility rating assigns attacks and assets to an evaluation of the attacker dimensions such as the required knowledge and the feasibility of attacks derived from it. Our application of sample use cases shows how this rating can be used to assess the feasibility of an entire attack path. The attack feasibility rating can be used as a building block in a threat and risk assessment according to ISO/SAE 21434.**

***Index Terms*—attack feasibility rating, risk analysis, threat analysis and risk assessment (TARA), automotive security, cybersecurity engineering, road vehicles, ISO/SAE 21434, threat mitigation and resilience, connected car, ISO 15118, AUTOSAR**

I. Introduction

Attacks on so-called *smart cars*, a term coined by ENISA as “systems providing connected, added-value features in order to enhance car users experience or improve car safety” [1], may have serious consequences, even to life and limb of road users. There have already been recalls of millions of vehicles due to security vulnerabilities. Efforts are therefore underway to regulate and standardize cybersecurity in a binding manner. For example, the United Nations Economic Commission for Europe (UNECE) World Forum for Harmonization of Vehicle Regulations (WP.29) adopted two regulations on cybersecurity and software updates that make cybersecurity relevant for the approval of new vehicle types. They will enter into force in January 2021. A central aspect is managing vehicle cyber risks. The standards SAE J3061 [2] and ISO/SAE 21434 [3] propose to consider cybersecurity engineering already in the concept phase of automotive engineering and require the execution of a comprehensive Threat Analysis and Risk Assessment (TARA). The purpose of TARA is the calculation of relative values for impact and attack feasibility to derive the associated risk in a subsequent step [4].

The importance of cybersecurity engineering approaches including TARAs such as ISO/SAE 21434 is reflected in the fact that they are now taken up, e.g., by the Automotive Open System Architecture (AUTOSAR) standards, as crucial precondition for the development process of their standardized automotive software framework [5].

In 2009, the EVITA project [6] has already proposed a security risk rating methodology [7] for automotive electrical / electronic (E/E) systems which is nowadays well-known in the automotive cybersecurity community. However, the EVITA project has evaluated its methodology using basic ratings that have been assessed under the umbrella and technological environment of that time. Now, more than 10 years later, new technologies, architecture components and system configurations have been introduced and the environment has changed considerably. Electric driving including new charging infrastructure and automated driving including new in-vehicle artificial intelligence components as well as new communication networks and distributed computing concepts such as edge computing have evolved. As a result, the EVITA attack feasibility rating, which relates each basic asset to its attack feasibility, is not applicable anymore.

In this paper, we present an attack surface assessment where we define an attack feasibility rating that can be used in a security engineering process compliant to ISO/SAE 21434. The main contributions of this work comprise 1) a generic reference architecture that can be mapped to a variety of modern in-vehicle architectures, 2) the identification of an extensive set of assets in modern vehicles that constitutes the attack surface, 3) a feasibility rating of possible attacks for each considered asset consistent with the requirements of ISO/SAE 21434, 4) a proof of concept evaluation of the attack feasibility rating by selection of several typical automotive use cases and exemplary assessment of risks related to attack paths within the selected use cases, and 5) an example on how to use this approach to evaluate the effects of security and mitigation technologies.

The remainder of the paper is organized as follows: First, we give a brief overview on background and related work in Section II. In Section III, we describe the setting we base our analysis on. This comprises the reference architecture and selected use cases. Then, we present our attack feasibility rating in Section IV and the application of our rating to our use cases in Section V. Finally, we conclude the paper in Section VI and give an outlook on future work.

II. Background and Related Work

For our work, the ISO/SAE 21434 standard for cybersecurity engineering in the automotive domain [3] is the most relevant background as the successor of the SAE J3061 “Cybersecurity Guidebook for Cyber-Physical Vehicle Systems” [2]. One main focus of security by design approaches in general [8] that has been adopted by the ISO/SAE 21434 is to provide a framework for the execution of a TARA to achieve a general cybersecurity concept. The whole process comprises five main steps, namely, item definition, risk analysis, risk treatment decision, derivation of the cybersecurity goal, and a cybersecurity concept. As depicted in Figure 1, for the risk analysis the steps of asset identification, threat scenario identification, impact rating, attack path analysis, attack feasibility rating, risk determination and risk treatment decision are performed for which the standard suggests some weighting criteria, categories, and matrices. These suggestions are given in the informative annexes; by this way, the suggested criteria with their value ranges are optional, and slightly modified values as well as complete other criteria may also be used within a risk analysis compliant to this ISO standard. In our work, we focus on the identification of generic assets and an attack feasibility rating for attacks on these assets.

In order to show how this approach can be integrated into the ISO process, we show exemplary how it can be used to assess specific threat scenarios and alternative security measures. This enables the evaluation of different mitigation strategies securing vehicle networks regarding their impact on the overall attack feasibility and thus contributes to the goal of a security design process for vehicles.

The roots of this cybersecurity engineering standard have been developed in several previous projects and standards.

In 2009, the EVITA project [6] proposed a security risk rating methodology [7] for automotive E/E systems building on generic approaches such as [9]. This methodology has been adopted in many subsequent research projects such as HEAVENS [10] and has also influenced the SAE J3061 [2] which uses examples from EVITA in its appendix. The researchers of HEAVENS state that the EVITA approach is the pioneering risk rating approach for the automotive industry [11].

The National Highway Traffic Safety Administration (NHTSA) suggested a composite threat model [12] relevant for the automotive industry. It comprises the identification of critical applications/systems by decomposition as well as the determination and analysis of threats.

The European Telecommunications Standards Institute (ETSI) published several versions of a Threat, Vulnerability, Risk Analysis (TVRA) [13]. TVRA relies on industry-proven methods and metrics to assess security risk. However, TVRA focuses only on telecommunication threats.

SAE J3061 [2] also mentions the risk assessment methods of EVITA [7], an early version of TVRA [13], and HEAVENS [11] as base for the respective task. In [14] an application of this SAE method to a communication control unit is shown.

Several research approaches suggested improvements to the standards. The SAHARA method [15] merges safety and security analysis in one approach. The safety analysis is based on Hazard Analysis and Risk Assessment (HARA) which identifies and categorizes dangerous events in relation to components of the system under development. A method named RACE [16] combines risk computation using the EVITA controllability concept with the TVRA rating of risk.

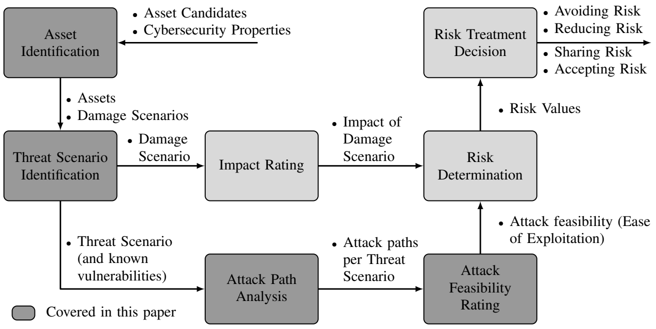

Fig. 1. ISO 21434 Approach

In [17] some improvements to existing methods with respect to driver-less vehicles have been proposed by a framework named SARA. SARA comprises an improved threat model and a new metric for attack observation for driver assistance systems controllability. Moreover, in [4] the TARA+ security analysis framework for automated driving systems combines some features of the above-mentioned SAE and ISO standards.

Our contribution comprises an extensive feasibility rating of possible attacks for each asset of a modern vehicle that constitutes the attack surface. Thus, the closet related work is the evaluation of required attack potential for asset attacks identified from attack trees in the EVITA deliverable 2.3 [7] and the attack surface tables proposed by Petit and Shladover in [18]. However, these tables are rather limited in scope and do not consider some important assets of modern vehicles.

III. Setting

In this section, we describe our assumed generic reference architecture and the example use cases used with our attack feasibility rating.

A. *Reference Architecture*

The topology of a modern automotive E/E system can be structured by different design principles. Traditional gateway-based topology is limited in network bandwidth and thus domain-based as well as centralized E/E architecture concepts have been developed [19]. Figure 2 shows the generic automotive architecture that we use as reference for further analysis. It is abstracted from current architectures of different manufacturers and shows the vehicle’s internal network topology with on-car Electronic Control Units (ECUs), bus systems, and sensors, as well as external entities in the vehicle’s environment with the corresponding communication channels.

The internal network topology is hierarchically structured and features an Ethernet backbone network that connects various Domain Gateways (GWs) that themselves are connected to low bandwidth Controller Area Network Flexible Data-Rate (CAN FD) sub-networks consisting of smaller ECUs.

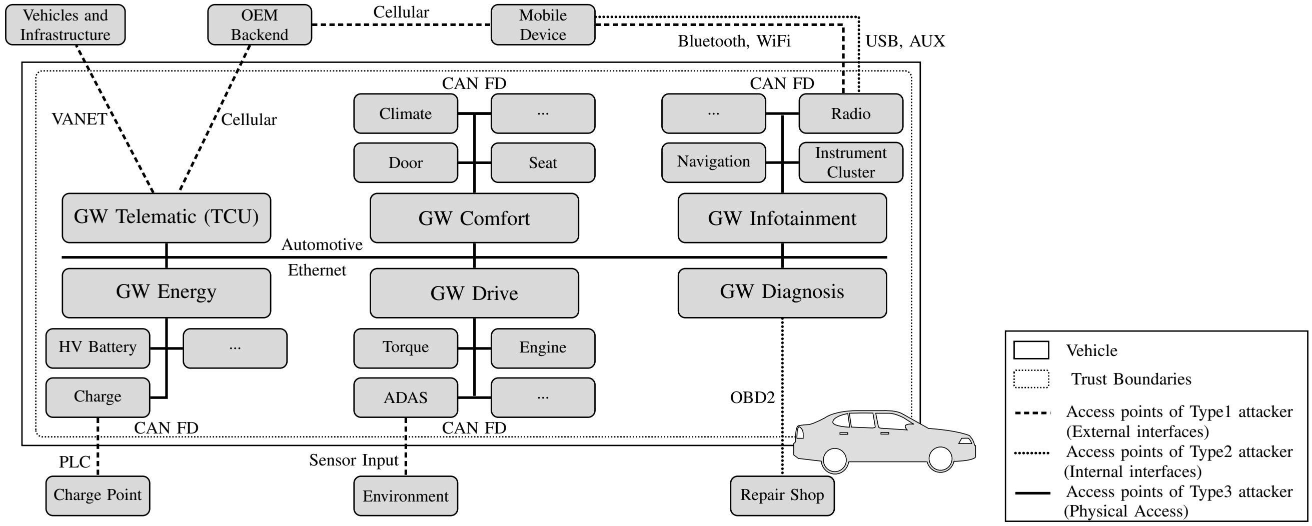

Fig. 2. Generic reference architecture for risk analysis

The domains are roughly structured based on their provided functionality into 1) telematic, 2) comfort, 3) infotainment, 4) energy, 5) drive, and 6) diagnosis domain.

The telematics domain is used to establish communications to the vehicle's environment and is represented by the Telematic Control Unit (TCU). The TCU mostly establishes communications via long or short range communication interfaces like cellular or Vehicular Ad Hoc Networks (VANETs) used for Vehicle-to-Everything (V2X) communication. Examples of this external communication are the transmission of software updates from OEM backends or, in case of V2X, traffic data exchange with other vehicles or roadside units.

The comfort domain consists of ECUs that control comfort features of the car. For example, this domain could implement functionality like (semi-) automatic seat adjustment, door control for the central locking system, and climate control.

The infotainment domain comprises the instrument cluster and multimedia systems like radio or CD player, and the navigation system. Most of the multimedia system functionality is typically bundled in one ECU, the headunit, that also allows the connection of personal consumer devices, like smartphones, via Bluetooth, WiFi, USB, or AUX.

The energy domain is exclusive for electric vehicles and comprises of the battery and its corresponding management system as well as the charging inlet. The latter is used to connect the vehicle via Power-Line Communication (PLC) to a charging station in order to charge the battery via Plug and Charge (PnC) using standards such as ISO 15118 [20].

The drive domain consists of ECUs that enable driving tasks, e.g., torque controller that sends acceleration or deceleration commands to the engine. Additionally, the drive domain includes the Advanced Driver-Assistance Systems (ADAS) controller that is connected to a variety of sensors and supports the driving task, e.g., by sensing the environment for obstacles.

Finally, the diagnosis domain exposes the On-board Diagnostics (OBD) interface. It is used by repair shops to retrieve diagnosis messages from the vehicle network.

Besides the topology of the reference architecture, Figure 2 also shows typical access points of attackers with increasing capabilities roughly based on [21] and the trust boundaries we base our attack rating on. In particular, we define all external entities as trusted but assume the external and on-car network including the ECUs as possibly compromised.

B. *Definition of Use Cases*

To show how our attack feasibility rating can be used to analyze attack paths of certain threat scenarios, we define three typical example use cases addressing three major security and safety critical application areas of a connected Electric Vehicle (EV). Each use case involves specific instances of ECUs and busses of the previously defined reference architecture.

Use Case 1: Electric Driving: The electric driving process provides the basic major application for the ECUs and busses of the reference architecture. This involves the drive domain with its corresponding ECUs like brake, accelerator and the engine. Attacks performed during driving pose a risk to life and limb of road users, i.e., it is highly safety-critical. In the example application described in Section V, we concentrate on threat scenarios where the communication with brake and accelerator is interrupted or manipulated preventing sensors in receiving proper signals.

Use Case 2: Conductive Charging: The second major application of the reference architecture is provided by charging the battery of the EV. This involves the energy management system with the battery as well as the external connection to the Charge Point (CP). The charging use case has safety (e.g., a manipulated CP may damage the vehicle with wrong charging parameters), security (e.g., the compromise of PnC charging credentials), and privacy implications (e.g., generation of movement profiles). In Section V, we concentrate on attacks against the proper establishment, use, and termination of the PnC communication between EV and CP.

Use Case 3: OTA Firmware Update: Over the air (OTA) firmware update is an important use case to fix software errors and vulnerabilities. Software updates are transferred, for example, from the OEM backend via cellular to the vehicle and installed via some diagnostics protocols. The security of this use case is crucial to ensure that no malware is installed or an attacker illegally activates some technical features without payment. In Section V, we use this use case with an attacker trying to install a manipulated update to unlock features.

IV. Attack Surface Assessment

In this section, we describe our attack surface assessment. We follow the approach of ISO/SAE 21434 [3] (cf., Figure 1 and the example in Annex G). We focus on identification of typical assets and the attack feasibility rating for single attack building blocks on these assets. We then show how to determine the overall feasibility of attack paths using the single attack building blocks.

A. Definition of Attack Assets

The defined assets cover the following five categories: 1) Cryptographic keys, 2) Wireless on-car interfaces and communications, 3) Wired on-car interfaces and communications, 4) On-car ECUs, and 5) On-car sensors. Each asset category is then split up into different attack methods and broken down to different technologies used in the vehicle.

1) Cryptographic keys: Cryptographic keys are required for many security mechanisms like secure communication or access control. For example, symmetric keys are used by AUTOSAR's Secure Onboard Communication (SecOC) [22], [23] for securing in-vehicle communication and asymmetric keys are used in the charging credentials for ISO 15118 PnC authentication. In this category, we consider the feasibility to break cryptographic algorithms or to illegally acquire or modify cryptographic keys. For the latter, we distinguish between hardware and software attacks both on keys stored within the ECU memory and within shielded locations, such as Hardware Security Modules (HSMs) or Trusted Platform Modules (TPMs).

2) Wireless on-car interfaces and communications: The wireless interfaces and communication channels enable an attacker to perform remote attacks without physical access to the car. The feasibility to intercept, listen, jam, corrupt, alter, inject, or replay messages via WiFi, Cellular, GPS, and Bluetooth interfaces are considered.

3) Wired on-car interfaces and communications: This category addresses an attacker with physical access to a car. The attacker can access exposed interfaces within the car, such as OBD, debug interfaces like JTAG, USB, and AUX, not directly accessible interfaces to bus systems, such as CAN, CAN FD, FlexRay, and Ethernet, and interfaces to the environment, such as PLC for PnC. These attacks are often used as an entry point into the system to carry out more sophisticated attacks.

4) On-car ECUs: ECUs of a car offer a variety of assets that an attacker might want to corrupt. We consider the following attacks: exploitation of vulnerabilities, denial of service / ECU disabling, configuration change, flashing of malicious code, and execution of malicious code (possibly with escalated privileges).

5) On-car sensors: A car is equipped with several sensors. We consider sensors for pedal position, steering angle, ultrasonic, Lidar, Radar, and cameras. An attack could spoof sensor signals or trick sensors by sending manipulated input.

B. Attack Feasibility Rating

To determine the attack feasibility of our basic attacks, we utilized the scheme from [24], which is also recommended in ISO 21434 [3]. It introduces the dimensions *elapsed time*, *specialist expertise*, *knowledge of the item or component*, *window of opportunity*, and *equipment*.

The dimension *elapsed time* characterizes how much time is needed to prepare and execute an attack and may vary between less than a week and more than three years.

Furthermore, *specialist expertise* describes the abilities of the attacker between layman with no particular expertise, a proficient familiar with the behavior of the target, an expert with deep knowledge of a specific technique (e.g., cryptoanalysis) and multiple experts from different fields of expertise.

Additionally, the required *knowledge of the item or component* indicates the difficulty of the attack. This may vary between public, restricted, confidential, or strictly confidential information necessary to perform the attack.

The *window of opportunity* describes the attacker's window of opportunity to perform an attack. This is mainly limited by the accessibility of the target. Basic attacks only consider the immediate opportunity and no pre-limiting conditions (e.g., sending a message on a bus depends on access to this bus by physical access or via an ECU but the window of opportunity is unlimited once the precondition is true). This dimension has the categories unlimited, easy, moderate, and difficult.

The last dimension of the attack rating is the *equipment* required by an attacker to successfully execute an attack. It has the categories standard, specialized, bespoke, and multiple bespoke.

The combination of all ratings results in the attack feasibility rating of a basic attack. The resulting attack feasibility rating can have the categories *Very Low*, *Low*, *Medium*, and *High*. Our rating of the different basic attacks is listed in Table I. The table lists all basic attacks on assets with respect to our defined trust boundaries (c.f., Figure 2).

C. Discussion on Different Path Calculation Methods

The basic attacks described in the last section will be concatenated to build complete attack paths. The overall feasibility of an attack path is then determined by aggregating the single values along the path. For the aggregation, different calculation methods can be used. We describe three basic approaches and their drawbacks in the following.

a) Sum: An obvious approach is to simply sum up all values per category along the path. A drawback of this approach is that the summed up values will more easily reach the boundaries of the model so that a fine-granular differentiation of longer attack paths is no longer possible.

TABLE I
ATTACK FEASIBILITY RATING

Id	Asset (attack)	Elapsed time	Specialist expertise	Knowledge of item/ component	Window of opportunity	Equipment	Attack feasibility
	Cryptographic Keys						
1.1	Keys (illegal acquisition, modification or breaking): Extract from HSM (Softwarebug)	< 3 years	Expert	Restricted	Unlimited	Specialized	Low
1.2	Keys (illegal acquisition, modification or breaking): Extract from HSM (Hardwareattack)	< 3 years	Multiple experts	Confidential	Difficult	Bespoke	Very Low
1.3	Keys (illegal acquisition, modification or breaking): Extract from TPM (Softwarebug)	> 3 years	Expert	Restricted	Unlimited	Specialized	Very Low
1.4	Keys (illegal acquisition, modification or breaking): Extract from TPM (Hardwareattack)	> 3 years	Multiple experts	Confidential	Difficult	Bespoke	Very Low
1.5	Keys (illegal acquisition, modification or breaking): Extract from Firmware (Software)	< 1 month	Proficient	Confidential	Unlimited	Specialized	Medium
1.6	Keys (illegal acquisition, modification or breaking): Break Cryptographic algorithm (min. AES-128/ RSA 2048/ ECC 256)	> 3 years	Expert	Public	Unlimited	Standard	Very Low
1.7	Keys (illegal acquisition, modification or breaking): Extract from Firmware (Hardware)	< 3 years	Expert	Confidential	Difficult	Bespoke	Very Low
1.8	Keys (forge): Brute Force SecOC	< 1 week	Proficient	Restricted	Difficult	Standard	Medium
	Wireless On-Car Interfaces and Communications						
2.1	Wireless Communications (jamming): GPS	< 1 week	Layman	Public	Easy	Specialized	High
2.2	Wireless Communications (jamming): WiFi (IEEE 802.11p)	< 1 week	Layman	Public	Easy	Standard	High
2.3	Wireless Communications (jamming): Cellular (LTE/5G)	< 1 week	Layman	Public	Easy	Specialized	High
3.1	Wireless Communications (corrupt / fake msg and info): WiFi (IEEE 802.11p)	< 1 week	Proficient	Public	Easy	Standard	High
3.2	Wireless Communications (corrupt / fake msg and info): Cellular (LTE/5G)	< 1 month	Proficient	Public	Easy	Specialized	High
3.3	Wireless Communications (corrupt / fake msg and info): GPS (spoofing)	< 1 month	Proficient	Public	Easy	Specialized	High
3.4	Wireless Communications (corrupt / fake msg and info): Connected Car (via Cellular)	< 1 week	Proficient	Public	Unlimited	Standard	High
3.5	Wired Communications (corrupt / fake msg and info): USB	< 1 week	Proficient	Public	Easy	Standard	High
3.6	Wired Communications (corrupt / fake msg and info): AUX	< 1 week	Proficient	Public	Easy	Specialized	High
4.1	Wireless Com. (listen): WiFi (IEEE 802.11p)	< 1 week	Proficient	Public	Easy	Standard	High
4.2	Wireless Com. (listen): Cellular (LTE/5G)	< 1 month	Proficient	Public	Easy	Specialized	High
4.3	Wireless Com. (listen): Bluetooth (BLE)	< 1 month	Proficient	Public	Easy	Specialized	High
5.1	Wireless Com. (intercept, alter, inject, replay): WiFi (IEEE 802.11p)	< 1 week	Proficient	Public	Easy	Standard	High
5.2	Wireless Com. (intercept, alter, inject, replay): Cellular (LTE/5G)	< 1 month	Proficient	Public	Easy	Specialized	High
5.3	Wireless Com. (intercept, alter, inject, replay): Bluetooth (BLE)	< 1 month	Proficient	Public	Easy	Specialized	High
	Wired On-Car Interfaces and Communications						
5.4	Wired Com. (intercept, alter, inject, replay): USB	< 1 week	Proficient	Public	Easy	Standard	High
5.5	Wired Com. (intercept, alter, inject, replay): AUX	< 1 week	Proficient	Public	Easy	Specialized	High
5.6	Wired Com. (intercept, alter, inject, replay): PLC	< 1 week	Proficient	Public	Easy	Specialized	High
5.7	Wired/Wireless Com. (spoof): External test and diagnostic equipment	< 1 week	Layman	Public	Unlimited	Specialized	High
6.1	On-car wireless interfaces (access): Bluetooth	< 1 week	Proficient	Public	Easy	Standard	High
6.2	On-car wireless interfaces (access): Cellular	< 1 week	Proficient	Public	Unlimited	Specialized	High
6.3	On-car wireless interfaces (access): WiFi	< 1 week	Proficient	Public	Easy	Standard	High
6.4	On-car wireless interfaces (access): GPS	< 1 week	Proficient	Public	Unlimited	Specialized	High
7.1	On-car user hardware interfaces (access): USB	< 1 week	Layman	Public	Easy	Standard	High
7.2	On-car user hardware interfaces (access): Aux	< 1 week	Layman	Public	Easy	Standard	High
8.1	On-car interfaces (access – physical tampering): OBD	< 1 week	Layman	Public	Easy	Standard	High
8.2	On-car interfaces (access – physical tampering): PLC	< 1 month	Proficient	Public	Easy	Specialized	High
8.3	On-car interfaces (access – physical tampering): CAN (FD), Ethernet	< 1 week	Proficient	Restricted	Moderate	Standard	High
8.4	On-car interfaces (access – physical tampering): FlexRay	< 1 week	Proficient	Restricted	Moderate	Specialized	Medium
8.5	On-car interfaces (access – physical tampering): Debug interfaces (e.g. JTAG) (for easy to access components)	< 1 month	Expert	Restricted	Moderate	Specialized	Medium
8.6	On-car interfaces (access – physical tampering): Debug interfaces (e.g. JTAG) (for components with difficult access e.g. HV Battery)	< 1 month	Expert	Restricted	Difficult	Specialized	Low
9.0	On-car Communications (disable or Denial of Service): CAN (FD), FlexRay, Ethernet	< 1 week	Proficient	Public	Unlimited	Standard	High
10.1	On-car Communications (listen): CAN (FD), FlexRay, Ethernet	< 1 month	Proficient	Public	Unlimited	Standard	High
10.2	On-car Communications (listen + understand): PLC	< 1 month	Proficient	Public	Easy	Standard	High
11.0	On-car Communications (intercept): CAN (FD), FlexRay, Ethernet	< 1 week	Proficient	Public	Unlimited	Standard	High
12.0	On-car Communications (replay): CAN (FD), FlexRay, Ethernet	< 1 week	Proficient	Public	Unlimited	Standard	High
13.0	On-car Communications (inject): CAN (FD), FlexRay, Ethernet	< 1 week	Proficient	Public	Unlimited	Standard	High

Continued on the next page

TABLE I
ATTACK FEASIBILITY RATING (CONTINUED)

Id	Asset (attack)	Elapsed time	Specialist expertise	Knowledge of item/ component	Window of opportunity	Equipment	Attack feasibility
	On-Car ECUs						
14.1	On-car Communications (alter): CAN (FD), FlexRay	< 1 week	Expert	Public	Unlimited	Specialized	High
14.2	On-car Communications (alter): Ethernet	< 1 week	Proficient	Public	Unlimited	Standard	High
15.1	On-car ECU (exploit vuln. or impl. error to access ECU): ECU with external interface (wireless (Cellular/BLE/Wifi))	< 6 months	Proficient	Restricted	Unlimited	Specialized	Medium
15.2	On-car ECU (exploit vuln. or impl. error to access ECU): ECU with external interface (wired (OBD/PLC/USB/AUX))	< 6 months	Proficient	Restricted	Unlimited	Specialized	Medium
15.3	On-car ECU (exploit vuln. or impl. error to access ECU): ECU with internal interface (wired (CAN (FD)/FlexRay/Ethernet))	< 6 months	Proficient	Restricted	Unlimited	Standard	High
15.4	On-car ECU (exploit vuln. or impl. error to access ECU): ECU with debug interface (wired (UART/JTAG/...))	< 6 months	Proficient	Restricted	Unlimited	Specialized	Medium
15.5	On-car ECU (exploit vuln. or impl. error to access ECU): XCP (via CAN (FD))	< 6 months	Proficient	Restricted	Unlimited	Standard	High
16.1	On-car ECU (disable or Denial of Service): Resource exhaustion of regular ECU	< 1 week	Proficient	Restricted	Unlimited	Standard	High
16.2	On-car ECU (disable or Denial of Service): Shutdown/Halt	< 1 month	Proficient	Restricted	Unlimited	Standard	High
16.3	On-car ECU (disable or Denial of Service): Resource exhaustion of High Performance ECU	< 1 week	Expert	Restricted	Unlimited	Specialized	High
17.1	On-car ECU (configuration change): Remote	< 1 month	Proficient	Restricted	Unlimited	Standard	High
17.2	On-car ECU (configuration change): Physical	< 1 month	Proficient	Restricted	Moderate	Specialized	Medium
18.1	On-car ECU (remote malware flash): No integrity measures	< 1 week	Proficient	Restricted	Unlimited	Standard	High
18.2	On-car ECU (remote malware flash): With integrity measures	< 3 years	Proficient	Restricted	Unlimited	Standard	Medium
19.1	On-car ECU (flash via physical access): ECU without integrity measures external flash	< 1 week	Proficient	Restricted	Moderate	Specialized	High
19.2	On-car ECU (flash via physical access): ECU with integrity measures (e.g., secure boot or measured boot)	< 3 years	Proficient	Restricted	Moderate	Specialized	Low
19.3	On-car ECU (flash via physical access): ECU without integrity measures with embedded flash	< 6 months	Expert	Restricted	Moderate	Bespoke	Low
20.0	On-car ECU (exploit for priv. Escalation)	< 6 months	Proficient	Restricted	Unlimited	Standard	High
21.0	On-car ECU (execute Code/Commands)	< 1 month	Proficient	Public	Unlimited	Standard	High
22.0	On-car ECU: Access to Replacement Parts	< 1 week	Layman	Public	Unlimited	Standard	High
	On-Car Sensors						
23.1	On-car Sensors (spoof of sensor Signal): Brake pedal position, Throttle pedal position, Steering angle sensor	< 1 week	Proficient	Restricted	Moderate	Specialized	Medium
23.2	On-car Sensors (spoof of sensor Signal): Ultrasonic, Lidar, Radar Sensor	< 1 week	Proficient	Restricted	Moderate	Specialized	Medium
23.3	On-car Sensors (spoof of sensor Signal): Rear view camera, Stereo front camera	< 1 week	Proficient	Restricted	Moderate	Specialized	Medium
24.1	On-car Sensors (disable or Denial of Service): Brake pedal position, Throttle pedal position, Steering angle sensor	< 1 week	Proficient	Restricted	Moderate	Standard	High
24.2	On-car Sensors (disable or Denial of Service): Ultrasonic, Lidar, Radar Sensor	< 1 week	Layman	Public	Easy	Standard	High
24.3	On-car Sensors (disable or Denial of Service): Rear view camera, Stereo front camera	< 1 week	Layman	Public	Easy	Standard	High
25.1	On-car Sensors (external manipulation of sensor input): Brake pedal position, Throttle pedal position, Steering angle sensor	< 1 week	Proficient	Restricted	Moderate	Specialized	Medium
25.2	On-car Sensors (external manipulation of sensor input): Ultrasonic, Lidar, Radar Sensor	< 1 week	Proficient	Restricted	Moderate	Specialized	Medium
25.3	On-car Sensors (external manipulation of sensor input): Rear view camera, Stereo front camera	< 1 week	Proficient	Restricted	Moderate	Specialized	Medium

b) Average: A different option is to calculate the average per category along the path. While this solves this problem of the *Sum* approach, it introduces another drawback. Extreme values towards the low or high end will be equalized in the overall feasibility rating. This means that attack paths with mostly moderate values are then equal to attack paths with strongly varying values. While an attack path with mostly moderate values may be indeed feasible for an attacker to successfully execute, an attack path with only one highly rated value, e.g., breaking state-of-the-art cryptography, is very unlikely to happen. However, this calculation model may rate them as equally possible which is not desired.

c) Maximum: Another approach would be to select the maximum values per category along the attack path. This allows a fine-granular differentiation of longer attack paths while acknowledging difficult attacks with a low feasibility. A drawback is that long medium-rated paths may have a lower aggregated feasibility in contrast to a short high-rated path.

Section V shows an exemplary application of the *Maximum* value calculation model. However, our model is also capable to support more sophisticated approaches, e.g., hybrid approaches where each category is calculated differently.

V. Exemplary Application on Threats derived from the Use Cases

In this section, we apply our attack feasibility assessment to our use cases. We first define some exemplary threat scenarios with corresponding attack paths in Section V-A. Then we describe the application of our feasibility rating and the outcome in Section V-B. Additionally, Section V-C shows how the overall attack feasibility rating can be influenced by introducing dedicated security measures into the system.

A. Definition of Threat Scenarios and Attack Paths

Table II shows a distinct threat scenario for each use case. Each threat scenario is mapped to a corresponding impact factor defined in ISO/SAE 21434 that is dominant in this threat. Additionally, we show multiple end points that need to be successfully attacked to achieve the threat scenario and describe exemplary attack paths for some of the end points (typeset in bold). The amount and complexity of applicable paths depends on the attacker capabilities and the system properties. For example, the attacker's knowledge about the system, e.g. insider, may introduce new attack steps while the introduction of security measures may reduce possible attacks.

For Use Case 1, we select a threat where the attacker manipulates the torque of the vehicle so that the vehicle will abruptly accelerate or brake to cause an accident injuring the driver or use the car as projectile to injure other road users. The dominating impact is the safety of the driver or road users. For a successful attack, the attacker needs to compromise the CAN FD sub-net of the GW Drive. In particular, the attacker needs to trick the engine into applying the wrong torque parameters. For this to happen, we identified the attack endpoints E1–E8 that are feasible to successfully perform the attack. Thereby, E1–E3 consist of manipulating the sent torque data by either injecting new messages (E1), replaying old messages (E2), or altering sent messages (E3). E4–E6 are about Denial of Service (DoS) attacks either by intercepting corrective torque messages sent from benign controller (E4), congesting the channel with garbage messages (E5), or disabling relevant ECUs (E6). Finally, E7 consists of taking over the ECU and send seemingly benign messages and E8 manipulates sensor input. For E1 and E8, we show exemplary attack paths $\text{AP}^{\text{E1}}_{\text{UC1}}$ and $\text{AP}^{\text{E8}}_{\text{UC1}}$ respectively. $\text{AP}^{\text{E1}}_{\text{UC1}}$ shows an attack first compromising the TCU via cellular (steps ① – ⑤) and then hijacking the GW Drive to send seemingly benign torque messages on the CAN FD bus that are accepted by the motor controller (steps ⑥ – ⑨). In $\text{AP}^{\text{E8}}_{\text{UC1}}$, an attacker compromises sensor input from the vehicle's environment to influence the ADAS into sending wrong commands to the engine (step ①).

For Use Case 2, we select a threat where the vehicle is immobilized by keeping it attached to the charging station. The dominating impact is operational since the driving functionality is degraded. To achieve this, an attack must compromise the CAN FD sub-net of the GW Energy, the CP, or the PLC communication in order to prevent the charging session from completion. Therefore, the attack endpoints E1-E7 (as shown in Table II) have been defined. E1-E4 are about message manipulation, to either intercept messages like a "Charging Stop" (E1), to replay old messages which normally occur in a charging session that keeps it alive (E2), to inject new messages (E3), or to alter already sent messages (E4). E5 means to take over an ECU and use it to send legitimate messages or to manipulate its process execution. E6 and E7 are about DoS attacks on either an ECU or a communication channel. For this use case, we show the exemplary attack paths $\text{AP}^{\text{E2}}_{\text{UC2}}$ and $\text{AP}^{\text{E5}}_{\text{UC2}}$. In $\text{AP}^{\text{E2}}_{\text{UC2}}$ we show an attack path where an attacker compromises the PLC communication between vehicle and CP to replay charging continue messages (steps ① – ②). In $\text{AP}^{\text{E5}}_{\text{UC2}}$, an attacker compromises the Radio via Bluetooth (steps ① – ⑤), then compromises GW Infotainment, then GW Energy (steps ⑥ – ⑧), and finally the Charge ECU where the execution of the charging session process is manipulated to prevent it from completion (steps ⑨ – Ⓝ).

For Use Case 3, we select a threat where an attacker flashes a compromised firmware to illegally unlock certain features. The dominating impact in this scenario is financial as the attacker saves money not buying them from the Original Equipment Manufacturer (OEM). For a successful attack, the attacker must modify the firmware of the relevant controller so that the desired feature gets unlocked. Thus, we identified the two attack endpoints E1 and E2 that are about flashing compromised firmware either remotely or physically. For each attack endpoint we show the exemplary attack paths $\text{AP}^{\text{E1}}_{\text{UC3}}$ and $\text{AP}^{\text{E2}}_{\text{UC3}}$. Attack path $\text{AP}^{\text{E1}}_{\text{UC3}}$ shows how an attacker could remotely flash a new firmware onto the engine ECU to, e.g., increase its horsepower. The attacker compromises the TCU via Cellular to accept the malicious firmware (steps ① – ⑤) and to inject messages into the network to update the firmware of the engine (step ⑥). In attack path $\text{AP}^{\text{E2}}_{\text{UC3}}$, the attacker uses physical access to flash a compromised firmware via the debug interface of the ECU (steps ① – ②).

B. Attack Path Analysis

By applying our calculation model, the attack paths $\text{AP}^{\text{E1}}_{\text{UC1}}$ and $\text{AP}^{\text{E8}}_{\text{UC1}}$ for T_{UC1} both achieve an overall feasibility rating of *Medium* (c.f., Table III). A closer look at the individual ratings reveals that there is a difference between both paths regarding the values *elapsed time* and *window of opportunity* that even out in the overall rating. These deviations are caused by the fact that the preparation time of $\text{AP}^{\text{E1}}_{\text{UC1}}$ is higher because it is a remote attack where first multiple vulnerabilities in the system needs to be found and exploited. However, the remote attack can then be executed any time over a cellular connection independently of the actual position of the vehicle, while in $\text{AP}^{\text{E8}}_{\text{UC1}}$ the attacker needs to be in range of the vehicle to spoof the sensor readings.

TABLE II
Threat scenarios with identified attack endpoints (EPs) and selected attack paths (APs) derived from our use cases

UC1	Threat	T_{UC1}: Manipulate torque (Safety impact)
	Identified EPs	**E1**: Inject manipulated torque value (13.0), E2: Replay previously recorded torque value (12.0), E3: Alter sent torque value (14.1), E4: Intercept sent torque value (11.0), E5: DoS channel (9.0), E6: Disable ECU to prevent sending of torque values (16.1), E7: Take over ECU and send torque value (21.0), **E8**: Manipulate incoming sensor values
	Selected APs	• $AP^{\mathbf{E1}}_{UC1}$: ① 6.2: On-car wireless interfaces (access): Cellular → ② 3.2: Wireless Communications (corrupt / fake msg and info): Cellular (LTE/5G) → ③ 15.1: On-car ECU (exploit vuln. or impl. error to access ECU): ECU with external interface (wireless (Cellular/BLE/Wifi))@TCU → ④ 20.0: On-car ECU (exploit for priv. Escalation)@TCU → ⑤ 21.0: On-car ECU (execute Code/Commands)@TCU → ⑥ 15.3: On-car ECU (exploit vuln. or impl. error to access ECU): ECU with internal interface (wired(CAN/CAN FD/FlexRay/Ethernet))@GW Drive → ⑦ 20.0: On-car ECU (exploit for priv. Escalation)@GW Drive → ⑧ 21.0: On-car ECU (execute Code/Commands)@GW Drive → ⑨ 13.0: On-car Communications (inject): CAN FD@GW Drive • $AP^{\mathbf{E8}}_{UC1}$: ① 23.2: On-car Sensors (spoof of sensor Signal): Ultrasonic, Lidar, Radar Sensor@Lidar
UC2	Threat	T_{UC2}: Immobilize car by never completing the charging session (Operational impact)
	Identified EPs	E1: Intercept messages that would complete the charging session (11.0), **E2**: Replay messages from a previously recorded charging session (12.0), E3: Inject messages to stay in charging session (13.0), E4: Alter messages to stay in charging session (5.6), **E5**: Take over ECU to stay in charging session (21.0), E6: DoS channel (9.0), E7: Disable ECU to prevent from handling the charging session (16.1)
	Selected APs	• $AP^{\mathbf{E2}}_{UC2}$: ① 8.2: On-car interfaces (access – physical tampering): Powerline (PLC)@PLC → ② 5.6: Wired Com. (intercept, alter, inject, replay): PLC@PLC • $AP^{\mathbf{E5}}_{UC2}$: ① 6.1: On-car wireless interfaces (access): Bluetooth → ② 5.3: Wireless Com. (intercept, alter, inject, replay) Bluetooth → ③ 15.1:On-car ECU (exploit vuln. or impl. error to access ECU): ECU with external interface (wireless (Cellular/BLE/Wifi)@Radio → ④ 20.0: On-car ECU (exploit for priv. Escalation)@Radio → ⑤ 21.0: On-car ECU (execute Code/Commands)@Radio → ⑥ 15.3: On-car ECU (exploit vuln. or impl. error to access ECU): ECU with internal interface (wired(CAN/CAN FD/FlexRay/Ethernet))@GW Infotainment → ⑦ 20.0: On-car ECU (exploit for priv. Escalation)@GW Infotainment → ⑧ 21.0: On-car ECU (execute Code/Commands)@GW Infotainment → ⑨ – Ⓝ …*Repeat attack sequence 15.3, 20.0, and 21.0 to continue hijacking GW Energy and finally the Charge Controller*
UC3	Threat	T_{UC3}: Flash compromised firmware to illegally unlock certain features (Financial impact)
	Identified EPs	**E1**: Flash compromised firmware via remote access (18.1), **E2**: Flash compromised firmware via physical access (19.1)
	Selected APs	• $AP^{\mathbf{E1}}_{UC3}$: ① 6.2: On-car wireless interfaces (access): Cellular → ② 3.2: Wireless Communications (corrupt / fake msg and info): Cellular (LTE/5G) → ③ 15.1: On-car ECU (exploit vuln. or impl. error to access ECU): ECU with external interface (wireless (Cellular/BLE/Wifi)@TCU → ④ 20.0: On-car ECU (exploit for priv. Escalation)@TCU → ⑤ 21.0: On-car ECU (execute Code/Commands)@TCU → ⑥ 18.1: On-car ECU (remote malware flash): ECU without integrity measures@Engine • $AP^{\mathbf{E2}}_{UC3}$: ① 8.6: On-car interfaces (access – physical tampering): Debug interfaces (e.g. JTAG) (for components with difficult access e.g. HV Battery)@Engine → ② 19.1: On-car ECU (flash via physical access): ECU without integrity measures external flash@Engine

In T_{UC2} both attack paths require that the attacker is in range of the vehicle to either compromise the PLC (AP^{E2}_{UC2}) or Bluetooth (AP^{E5}_{UC2}) communication. Differences are again caused by *elapsed time* for exploiting vulnerabilities and also *knowledge of the item or component*. The knowledge needed to compromise the PLC communication is rated easier since the whole charging protocol is standardized, e.g., in ISO 15118, and, in contrast to in-vehicle internals that may be IP-protected by the OEMs, publicly available. This differences are major and thus result in a different overall feasibility rating where the feasibility of AP^{E2}_{UC2} is rated *High* and AP^{E5}_{UC2} is rated *Medium*.

Finally, in attack paths AP^{E1}_{UC3} and AP^{E2}_{UC3} of T_{UC3} the attacker flashes a compromised firmware either remotely or via physical access by using the debug interface of the corresponding ECU. While the differences in *elapsed time* and *window of opportunity* are again introduced by the remote attack, the attacker needs to have expert expertise for the physical attack, e.g., to circumvent physical protection mechanisms to first access the debugging interfaces. This causes a high value regarding *specialist expertise* for AP^{E2}_{UC3}. In the overall feasibility rating AP^{E1}_{UC3} achieves a rating of *Medium* while AP^{E2}_{UC3} achieves a rating of *Low*. This seems reasonable to us because of the high level of expertise needed in comparison to the small window of opportunity.

This chapter showed that we could successfully apply our approach to the *Threat Scenario Identification* and *Attack Path Analysis* steps in ISO 21434 (c.f., Figure 1).

C. Evaluation of Security Measures

In the previous chapters, we based our analysis on a system where no dedicated security mechanisms are implemented. Now, we show by example how the proposed approach can be used to evaluate effects of different additionally implemented security technologies on the overall attack feasibility rating.

Table I already shows how the rating of single attack steps varies depending on the system's security properties. For example, the feasibility for the illegal acquisition of cryptographic keys (1.1–1.7) gets lower if shielded locations, e.g., HSMs or TPMs, are used to store the keys. Also flashing malicious software (18.1–18.2) gets more difficult if software integrity verification measures are implemented. An example is the remote malware flash threat of UC3. Table IV shows a comparison of the resulting overall feasibilities in a system in which the security measures are successively expanded.

The first section of the table shows the original attack path that originates from a system without dedicated security measures (AP^{E1}_{UC3}, c.f., Table III). The following two sections show the adapted attack paths that result from the successive introduction of security technologies. The second section

TABLE III
INDIVIDUAL RATING TUPLES FOR THE IDENTIFIED ATTACK PATHS AND THE RESULTING OVERALL ATTACK PATH FEASIBILITY

The values for the resulting overall attack path feasibility are selected according to Section IV-C and typeset in bold.

Attack Path AP^{E1}_{UC1}						
6.2	(<1 week,	**Proficient**,	Public,	Unlimited,	**Specialized**,	High)
3.2	(<1 month,	Proficient,	Public,	**Easy**,	Specialized,	High)
15.1	(**<6 months**,	Proficient,	**Restricted**,	Unlimited,	Specialized,	Medium)
20.0	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
15.3	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
20.0	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
13.0	(<1 week,	Proficient,	Public,	Unlimited,	Standard,	High)
$\sum$	(<6 months,	Proficient,	Restricted,	Easy,	Specialized,	**Medium**)
Attack Path AP^{E8}_{UC1}						
23.2	(**<1 week**,	**Proficient**,	**Restricted**,	**Moderate**,	**Specialized**,	Medium)
$\sum$	(<1 week,	Proficient,	Restricted,	Moderate,	Specialized,	**Medium**)
Attack Path AP^{E2}_{UC2}						
8.2	(**<1 month**,	**Proficient**,	**Public**,	**Easy**,	**Specialized**,	High)
5.6	(<1 week,	Proficient,	Public,	Easy,	Specialized,	High)
$\sum$	(<1 month,	Proficient,	Public,	Easy,	Specialized,	**High**)
Attack Path AP^{E5}_{UC2}						
6.1	(< 1 week,	**Proficient**,	Public,	**Easy**,	Standard,	High)
5.3	(< 1 month,	Proficient,	Public,	Easy,	**Specialized**,	High)
15.1	(**<6 months**,	Proficient,	**Restricted**,	Unlimited,	Specialized,	Medium)
20.0	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
15.3	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
20.0	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
…Repeat attack sequence 15.3, 20.0, and 21.0 to continue hijacking GW Energy and finally the Charge Controller …						
$\sum$	(<6 months,	Proficient,	Restricted,	Easy,	Specialized,	**Medium**)
Attack Path AP^{E1}_{UC3}						
6.2	(<1 week,	**Proficient**,	Public,	Unlimited,	**Specialized**,	High)
3.2	(<1 month,	Proficient,	Public,	**Easy**,	Specialized,	High)
15.1	(**<6 months**,	Proficient,	**Restricted**,	Unlimited,	Specialized,	Medium)
20.0	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
18.1	(<1 week,	Proficient,	Restricted,	Unlimited,	Standard,	High)
$\sum$	(<6 months,	Proficient,	Restricted,	Easy,	Specialized,	**Medium**)
Attack Path AP^{E2}_{UC3}						
8.6	(**<1 month**,	**Expert**,	**Restricted**,	**Difficult**,	**Specialized**,	Low)
19.1	(<1 week	Proficient	Restricted	Moderate	Specialized	High)
$\sum$	(<1 month,	Expert,	Restricted,	Difficult,	Specialized,	**Low**)

shows the adapted attack path for a system with a security level 1 ($AP^{E1'}_{UC3}$) and the third section shows the adapted attack path for a system with a security level 2 ($AP^{E1''}_{UC3}$).

The system with security level 1 implements integrity protection mechanisms like secure and measured boot making it more difficult for an attacker to flash arbitrary software. To overcome these mechanisms, the attacker first needs an image to be signed with the correct image signing key or needs to replace the verification key. The attacker also needs to replace the integrity reference values to successfully flash the modified/old image. These additional steps increase the *elapsed time* parameter in this attack step. The introduction of these security mechanisms reduces the overall feasibility rating to *Low*.

The system with security level 2 additionally adds AUTOSAR's SecOC to the on-car bus systems. The corresponding keys are assumed to be stored in a TPM that shields them against unauthorized access. This makes the attack even more difficult since an attacker cannot send arbitrary data without knowing the corresponding message authentication key. The attacker is forced to acquire the correct SecOC key to make the attack persistent even after a system reboot. Therefore, a successful execution of an additional preceding attack path is necessary where the attacker hijacks the relevant ECU and exploits a software bug of the TPM interface to obtain the SecOC key. Especially, attack step 1.3 (exploiting a software bug of the TPM) increases the difficulty of the overall attack because of the high requirements for *elapsed time* and *specialist expertise*. Due to this, the overall feasibility drops to *Very Low* and thus is very unlikely to be successful.

Through the introduction of two different security technologies we could successively decrease the overall feasibility for a specific attack from *Medium* in a system with no security technologies to *Very Low* in a system with integrity protection mechanisms and secured channels via SecOC. This shows that our approach is also applicable to reflect adaptations of the E/E architecture with regard to the attack surface assessment.

VI. Conclusion and Future Work

Threat and risk analyses are an important part of an automotive cybersecurity engineering process. A central aspect for determining risks is the identification of the attack surface with an extensive feasibility rating of possible attacks for each asset of a modern vehicle for which we defined a generic reference architecture. Our rating can be used in a TARA to rate an entire attack path of a threat scenario. A vehicle manufacturer can use these ratings in combination with an impact rating to determine the risks according to ISO/SAE 21434. As an example, we used our attack feasibility rating to rate threat scenarios with corresponding attack paths of three use cases. Furthermore, we showed how the attack feasibility rating decreases if certain security mechanisms are introduced into the system. This attack surface assessment method enables the evaluation of different security and mitigation technologies regarding their benefits towards a more secure vehicle architecture.

As future work, we will further analyze the attack surface by identifying additional attacks and attack paths. Since a manual identification of all attack paths is not possible (because of permutations of attack building blocks), we plan to develop a tool for automated attack path generation and rating.

Acknowledgment

This research work has been partly funded by the German Federal Ministry of Education and Research (BMBF) and the

TABLE IV
INDIVIDUAL RATING TUPLES FOR THE MODIFIED ATTACK PATH AND THE RESULTING OVERALL ATTACK PATH FEASIBILITY

The values for the resulting overall attack path feasibility are selected according to Section IV-C and typeset in bold. Attack steps marked with '' differ from the original attack path and result from the introduction of the security mechanisms.*

ATTACK PATH $\mathrm{AP}_{\mathrm{UC3}}^{\mathrm{E1}}$ (No additional security mechanisms)						
6.2	(<1 week,	**Proficient**,	Public,	Unlimited,	**Specialized**,	High)
3.2	(<1 month,	Proficient,	Public,	**Easy**,	Specialized,	High)
15.1	(**<6 months**,	Proficient,	**Restricted**,	Unlimited,	Specialized,	Medium)
20.0	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
18.1	(<1 week,	Proficient,	Restricted,	Unlimited,	Standard,	High)
$\sum$	(<6 months,	Proficient,	Restricted,	Easy,	Specialized,	**Medium**)
ATTACK PATH $\mathrm{AP}_{\mathrm{UC3}}^{\mathrm{E1'}}$ (Security Level 1)						
6.2	(<1 week,	**Proficient**,	Public,	Unlimited,	**Specialized**,	High)
3.2	(<1 month,	Proficient,	Public,	**Easy**,	Specialized,	High)
15.1	(<6 months,	Proficient,	**Restricted**,	Unlimited,	Specialized,	Medium)
20.0	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
18.2*	(**<3 years**,	Proficient,	Restricted,	Unlimited,	Standard,	Medium)
$\sum$	(<3 years,	Proficient,	Restricted,	Easy,	Specialized,	**Low**)
ATTACK PATH $\mathrm{AP}_{\mathrm{UC3}}^{\mathrm{E1''}}$ (Security Level 2)						
6.2	(<1 week,	Proficient,	Public,	Unlimited,	**Specialized**,	High)
3.2	(<1 month,	Proficient,	Public,	**Easy**,	Specialized,	High)
15.1	(<6 months,	Proficient,	**Restricted**,	Unlimited,	Specialized,	Medium)
20.0	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
13.0*	(<1 week,	Proficient,	Public,	Unlimited,	Standard,	High)
20.0*	(<6 months,	Proficient,	Restricted,	Unlimited,	Standard,	High)
21.0*	(<1 month,	Proficient,	Public,	Unlimited,	Standard,	High)
1.3*	(**>3 years**,	**Expert**,	Restricted,	Unlimited,	Specialized,	Very Low)
18.2*	(<3 years,	Proficient,	Restricted,	Unlimited,	Standard,	Medium)
$\sum$	(>3 years,	Expert,	Restricted,	Easy,	Specialized,	**Very Low**)

Hessen State Ministry for Higher Education, Research and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE and by the BMBF project VITAF (ID 16KIS0835). Additionally, the project leading to this application has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 883135.

REFERENCES

[1] ENISA, "Cyber security and resilience of smart cars," ENISA, Tech. Rep., 2016.

[2] SAE International, "Cybersecurity guidebook for cyber-physical vehicle systems," SAE International, Tech. Rep. J3061, 2016.

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

[4] A. Bolovinou, U. Atmaca, A. T. Sheik, O. Ur-Rehman, G. Wallraf, and A. Amditis, "Tara+: Controllability-aware threat analysis and risk assessment for l3 automated driving systems," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 8–13.

[5] AUTOSAR, "Explanation of safety overview," https://www.autosar.org/fileadmin/user_upload/standards/adaptive/20-11/AUTOSAR_EXP_SafetyOverview.pdf, 2020, last accessed 2020-01-13.

[6] The EVITA consortium, "EVITA Threat and Risk Analysis," https://www.evita-project.org, 12 2009, last accessed 2020-01-13.

[7] A. Ruddle, D. Ward, B. Weyl, S. Idrees, Y. Roudier, M. Friedewald, T. Leimbach, A. Fuchs, S. Gürgens, O. Henniger, R. Rieke, M. Ritscher, H. Broberg, L. Apvrille, R. Pacalet, and G. Pedroza, "Security requirements for automotive on-board networks based on dark-side scenarios," EVITA project, EVITA Deliverable D2.3, 2009, last accessed 2020-01-13. [Online]. Available: https://evita-project.org/deliverables.html

[8] A. Chattopadhyay, K. Lam, and Y. Tavva, "Autonomous vehicle: Security by design," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–15, 2020.

[9] ISO/IEC, "ISO/IEC 18045:2008(E): Information technology – Security techniques – Methodology for IT security evaluation," International Organization for Standardization, Geneva, CH, Standard, 2008.

[10] HEAVENS consortium, "HEAVENS - HEAling Vulnerabilities to Enhance Software Security and Safety," https://research.chalmers.se/en/project/5809, 2016, last accessed 2020-01-13.

[11] A. Lautenbach and M. Islam, "HEAVENS - HEAling Vulnerabilities to Enhance Software Security and Safety," http://autosec.se/wp-content/uploads/2018/03/HEAVENS_D2_v2.0.pdf, 03 2016, last accessed 2020-01-13.

[12] C. McCarthy, K. Harnett, and A. Carter, "Characterization of potential security threats in modern automobiles: A composite modeling approach," National Highway Traffic Safety Administration, 10 2014.

[13] ETSI, "ETSI TS 102 165-1 V5.2.3 (2017-10) – CYBER; Methods and protocols; Part 1: Method and pro forma for Threat, Vulnerability, Risk Analysis (TVRA)," ETSI, Tech. Rep., 2017.

[14] C. Schmittner, Z. Ma, C. Reyes, O. Dillinger, and P. Puschner, "Using sae j3061 for automotive security requirement engineering," in *Computer Safety, Reliability, and Security*, A. Skavhaug, J. Guiochet, E. Schoitsch, and F. Bitsch, Eds. Cham: Springer, 2016, pp. 157–170.

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

[16] A. Boudguiga, A. Boulanger, P. Chiron, W. Klaudel, H. Labiod, and J. Seguy, "Race: Risk analysis for cooperative engines," in *2015 7th International Conference on New Technologies, Mobility and Security (NTMS)*, 2015, pp. 1–5.

[17] J. P. Monteuuis, A. Boudguiga, J. Zhang, H. Labiod, A. Servel, and P. Urien, "SARA: security automotive risk analysis method," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security, CPSS@AsiaCCS 2018, Incheon, Republic of Korea, June 04-08, 2018*, D. Gollmann and J. Zhou, Eds. ACM, 2018, pp. 3–14.

[18] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2015.

[19] M. Dibaei, X. Zheng, K. Jiang, R. Abbas, S. Liu, Y. Zhang, Y. Xiang, and S. Yu, "Attacks and defences on intelligent connected vehicles: a survey," *Digital Communications and Networks*, vol. 6, no. 4, pp. 399 – 421, 2020.

[20] ISO/IEC, "ISO/DIS 15118 — Road vehicles – Vehicle to grid communication interface – Part 20: 2nd Generation network and application protocol requirements," International Organization for Standardization, Geneva, CH, Standard, 2020.

[21] D. Zelle, R. Rieke, C. Plappert, C. Krauß, D. Levshun, and A. Chechulin, "Sepad – security evaluation platform for autonomous driving," in *2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, 2020, pp. 413–420.

[22] AUTOSAR, "Specification of secure onboard communication - CP release 20-11," https://www.autosar.org/fileadmin/user_upload/standards/classic/20-11/AUTOSAR_SWS_SecureOnboardCommunication.pdf, 2020, last accessed 2020-01-13.

[23] ——, "Specification of secure onboard communication protocol - AP release 20-11," https://www.autosar.org/fileadmin/user_upload/standards/foundation/20-11/AUTOSAR_PRS_SecOcProtocol.pdf, 2020, last accessed 2020-01-13.

[24] O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle, and B. Weyl, "Security requirements for automotive on-board networks," in *2009 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST)*, 2009, pp. 641–646.