Microprocessors and Microsystems 90 (2022) 104461

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

ThreatSurf: A method for automated Threat Surface assessment in automotive cybersecurity engineering

Daniel Zelle [a,*], Christian Plappert [a], Roland Rieke [a], Dirk Scheuermann [a], Christoph Krauß [b]

[a] Fraunhofer Institute for Secure Information Technology, Darmstadt, Germany
[b] Darmstadt University of Applied Sciences, Darmstadt, Germany

ARTICLE INFO

Keywords:
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
UN regulation no. 155
Automated attack path generation

ABSTRACT

Connected smart cars enable new attacks that may have serious consequences. Thus, the development of new cars must follow a cybersecurity engineering process including a Threat Analysis and Risk Assessment (TARA). The attack surface assessment is a central aspect of a TARA. In this paper, we introduce a concrete approach for attack surface assessment following the steps asset identification, threat scenario identification, attack path analysis, and attack feasibility rating of a TARA compliant to ISO/SAE DIS 21434 and an approach to automatize them. We define a generic reference architecture and assets constituting the attack surface, attack building blocks with associated feasibility rating, and a method for automated generation and rating of attack paths using the attack building blocks and attack feasibility. Our exemplary application of the automated attack surface assessment on several threats from the UN regulation no. 155 shows the feasibility of our approach.

1. Introduction

There is currently a major technology change taking place in modern cars. The electrical/electronic (E/E) system within a car, i.e., the in-vehicle network, changes. New technologies such as automotive Ethernet replace or complement legacy systems such as the Controller Area Network (CAN) bus. Several functionalities, which have been realized in the past with dedicated electronic control units (ECUs), are now consolidated on one more powerful ECU. Also, new communication paradigms such as service-oriented communication are introduced with the adaptive platform of Automotive Open System Architecture (AUTOSAR). The previously closed car system is now also connected to other cars, mobile terminals (e.g., smartphones), infrastructure components (e.g., charging stations for electric vehicles), the Internet, and various backend systems (e.g., of the manufacturer or insurance companies). This technology change is required for realizing autonomous driving functions and enabling a large number of value-added services for the so-called *smart cars* [1]. However, this also enables new attack possibilities. The potential consequences of attacks include economic damage (e.g., expensive recalls, loss of reputation), threats to life and limb of road users (e.g., by disabling the brakes), and privacy violations (e.g., the creation of movement profiles).

Efforts are therefore underway to regulate and standardize cybersecurity in a binding manner. For example, the United Nations Economic Commission for Europe (UNECE) has issued regulations 155 "Cyber security and cyber security management system" [2] and 156 "Software update and software update management system" [3] that make cybersecurity relevant for the approval of new vehicle types. A central aspect is the management of vehicle cyber risks which is addressed in the standards SAE J3061 [4] and ISO/SAE DIS 21434 [5] for cybersecurity engineering. This engineering process requires the execution of a comprehensive Threat Analysis and Risk Assessment (TARA) with a calculation of relative values for impact and attack feasibility to derive the associated risk [6].

In this paper, we introduce a concrete approach for attack surface assessment following the steps asset identification, threat scenario identification, attack path analysis, and attack feasibility rating of a TARA compliant to ISO/SAE DIS 21434. Furthermore, we propose an automation for the previously mentioned approach. This paper is based on our own preliminary work presented in [7]. The overall contributions of our work comprise 1. a generic reference architecture that can be mapped to a variety of modern E/E architectures, 2. an extensive set

* Corresponding author.
E-mail addresses: daniel.zelle@sit.fraunhofer.de (D. Zelle), christian.plappert@sit.fraunhofer.de (C. Plappert), roland.rieke@sit.fraunhofer.de (R. Rieke), dirk.scheuermann@sit.fraunhofer.de (D. Scheuermann), christoph.krauss@h-da.de (C. Krauß).

https://doi.org/10.1016/j.micpro.2022.104461
Received 20 May 2021; Received in revised form 7 January 2022; Accepted 21 January 2022
Available online 5 February 2022

of assets in modern cars that constitutes the attack surface, 3. attack building blocks with associated feasibility rating consistent with the requirements of ISO/SAE DIS 21434, 4. an algorithm for automated generation and rating of attack paths using the attack building blocks and attack feasibility, and 5. an exemplary application of the automated attack surface assessment on several threats from the UN regulation no. 155. Out of the scope of our work is the impact rating (and subsequent risk determination), since the impact is heavily manufacturer specific and should be defined by a car manufacturer.

The remainder of the paper is organized as follows: First, we give a brief overview on background and related work in Section 2. In Section 3, we describe the setting we base our analysis on. This comprises the reference architecture and selected use cases. Then, we present our attack surface assessment in Section 4, introduce our approach to automatically determine critical risks in Section 5 and the exemplary application in Section 6. Finally, we conclude the paper in Section 7 and give an outlook on future work.

2. Background and related work

One main focus of security by design approaches in general [8] is to provide a framework for the execution of a TARA to achieve a general cybersecurity concept. The topic of TARA is not new in the automotive domain. Cybersecurity engineering standards were developed in several projects and standards.

In 2009 the EVITA project [9] proposed a method for assessing the security risk [10] for automotive E/E systems that is based on generic approaches such as ISO/IEC 18045:2008 [11]. The EVITA methodology was adopted in many subsequent research projects such as HEAVENS [12] and has also influenced SAE J3061 [4], which uses examples from EVITA in its appendix. In 2016, the HEAVENS researchers stated that the EVITA approach is the pioneering risk assessment approach for the automotive industry [13]. Since then, several other methods have been proposed.

In 2014, the National Highway Traffic Safety Administration (NHTSA) proposed a composite threat model for the automotive industry [14]. Threat analysis steps include identifying critical applications/systems through decomposition, and identifying and analyzing threats.

In 2017 the European Telecommunications Standards Institute (ETSI) published several versions of a Threat, Vulnerability, Risk Analysis (TVRA) [15]. TVRA relies on industry-proven methods and metrics to assess security risk. However, TVRA only focuses on telecommunications threats.

SAE J3061 [4], which was released in 2016, also mentions the risk assessment methods of EVITA [10], an early version of TVRA [15], and HEAVENS [13] as a basis for the respective task. In [16] an application of this SAE method to a communication control unit is shown.

Several research approaches suggested improvements to the standards. The SAHARA method [17], proposed by Macher et al. in 2015, brought together 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. SAHARA uses the STRIDE model to identify threats in a security analysis. In 2016, Macher et al. also reviewed threat analysis methods and the recommendations of SAE J3061 regarding TARA methods [18].

In 2015, Boudguiga et al. proposed a method named RACE [19] that combines EVITA with TVRA. The authors clarified the definition of EVITA attack tree for automotive experts by using automotive functions instead of EVITA attack objectives. They also suggested a risk calculation using the EVITA controllability concept with the TVRA risk assessment.

In 2018, Monteuuis et al. suggested some improvements to existing methods with respect to driver-less vehicles. They proposed a framework named SARA [20] that comprises an improved threat model and a new metric for attack observation for driver assistance systems controllability.

In 2019, Bolovinou et al. suggested a controllability-aware framework named TARA+ [6]. This security analysis framework for automated driving systems combines some features of the above-mentioned SAE and ISO standards.

In 2019, Maple et al. published a reference architecture for attack surface analysis of smart cars in [21]. They demonstrate with case studies how attack trees can be used to understand the attack surface of connected vehicles. Their architecture is more generic when compared to the reference architecture we considered for the work presented here. For example, they under-specify the different internal interaction possibilities and thus hide respective implementation detail. This in turn leads to more abstract attack trees which allow to analyze the attack surface in general but not to the level of different implementation variants. In our work, however, we needed this level of detail in order to be able to compare the advantages and risks of different decisions with respect to the structure of the communication architecture and the concrete selection of communication protocols.

In 2021, the final draft international standard ISO/SAE DIS 21434 [5] for cybersecurity engineering in the automotive domain, which supersedes the SAE J3061 “Cybersecurity Guidebook for Cyber-Physical Vehicle Systems” [4] appeared. The attack surface assessment with a TARA as described in [5] involves several steps which are depicted in Fig. 1. First, relevant assets and corresponding threat scenarios are identified. For each threat scenario, the attacks paths for reaching this scenario are analyzed and the attack feasibility is rated. In parallel, the potential damage caused by the threat scenarios is rated in an impact rating. Impact and attack feasibility are then used to determine the risk. Finally, appropriate risk treatment must be applied to deal with the risks. In the informative annexes the standard also suggests some weighting criteria, categories, and matrices. In this way, the proposed criteria with their value ranges are optional, and slightly modified values as well as complete other criteria may also be used in the context of a risk analysis that corresponds to this ISO standard. In [22], Dantas et al. illustrate by example how an incremental security engineering approach based on ISO/SAE DIS 21434 can support engineers in constructing arguments for item security claims.

In 2021, also two new UN regulations on vehicle approval related to cybersecurity [2] and software update [3] appeared. These are the first internationally binding regulations in the field of cybersecurity in connected and autonomous vehicles. In addition, the interpretation of [2] in [23] provides a table of relationships between the requirements of the regulation and the relevant paragraphs of ISO/SAE DIS 21434.

In our work, we focus on the identification of generic assets of a modern vehicle that form the attack surface and a feasibility assessment for attacks on these assets. The *attack tree* concept has been introduced by Schneier in [24]. A generic notation for unified parametrizable attack trees has been proposed by Wang et al. in [25]. *Attack graphs* have been introduced by Phillips and Swiler in [26]. This concept is closely related to attack trees but attack graphs usually comprise all possible attack paths not just one goal. Abstraction methods for the computation of compact representations of the graph and the inclusion of the liveness analysis have been introduced by Rieke in [27]. A formal model of an *attack surface* is provided by Manadhata and Wing in [28]. The closest related work is the evaluation of the required attack potential identified using the attack trees in EVITA Deliverable 2.3 [10] and the attack surface tables proposed by Petit and Shladover in [29]. However, these tables are rather limited in scope and do not consider some important assets of modern vehicles. To evaluate how our approach can be integrated into the process described in ISO/SAE DIS 21434, we demonstrate on examples of how our method can be used to evaluate specific attack scenarios and alternative security measures. For this purpose, we have selected some threats from the comprehensive list of known threats to vehicles in Annex 5 of UN Regulation No. 155 [2]. Our approach also enables the evaluation of different mitigation strategies for securing vehicle networks in terms of their impact on the feasibility of the overall attack, contributing to the goal of a vehicle security design process.

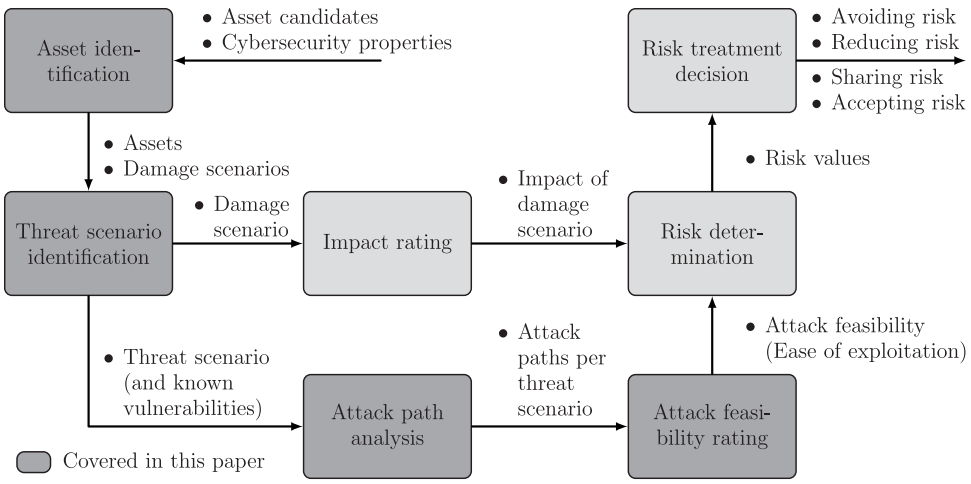

Fig. 1. ISO/SAE DIS 21434 approach.

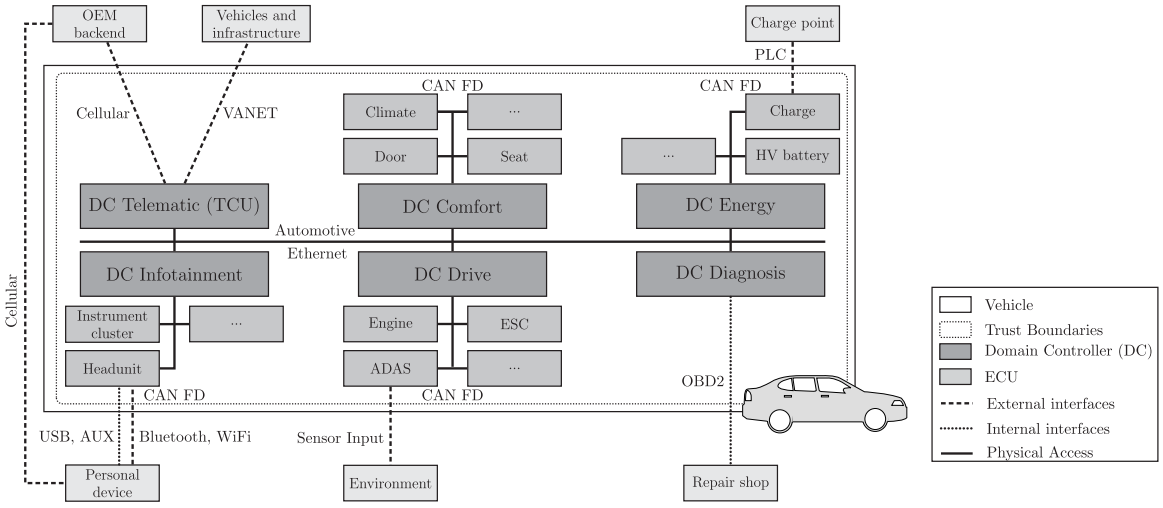

Fig. 2. Generic reference architecture for risk analysis.

3. Setting

In this section, we go into detail about our assumed generic reference architecture and the exemplary use cases on which we base our further analysis.

3.1. Reference architecture

Modern automotive E/E architectures can be structured according to their topology by different design principles. Since traditional topologies with a central gateway are limited regarding the overall network bandwidth, domain-based and centralized E/E architecture concepts have been developed [30].

Fig. 2 shows the generic automotive architecture that we use as reference for further analysis. It is based on a domain-based E/E architecture and abstracted from current architectures of different manufacturers. It 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.

As domain-based E/E architecture the internal network topology is hierarchically structured and features an Ethernet backbone network that connects various Domain Controllers (DCs). The DCs are connected to low bandwidth Controller Area Network Flexible Data-Rate (CAN FD) sub-networks consisting of smaller ECUs.

The domains are roughly structured based on their provided functionality into telematic, comfort, energy, infotainment, drive, and diagnosis domain.

The telematics domain is used to establish communication with the vehicle environment. It is typically represented by the Telematic Control Unit (TCU). The TCU usually establishes communication via long- or short-range communication interfaces such as mobile radio or Vehicular Ad Hoc Networks (VANETs), which are used for V2X communication. Examples of this external communication include the transmission of software updates from OEM backends or, in the case of Vehicle-to-Everything (V2X), the exchange of traffic data with other vehicles or roadside units.

The comfort domain consists of ECUs that control comfort functions, e.g., (semi-)automatic seat adjustment, door control for central locking, and climate control, of the vehicle.

The energy domain is exclusive to electric vehicles and consists of the battery and the associated battery management system as well as the charge controller. The latter is used to establish a connection between the vehicle and a charging station to charge the battery, e.g., via Power-Line Communication (PLC) and a Plug and Charge (PnC) standard such as ISO 15118 [31].

The infotainment domain includes the instrument cluster and multimedia systems such as the radio or CD player as well as the navigation system. Most of the functions of the multimedia system are usually bundled in the headunit ECU. The headunit also enables the connection

of personal devices such as smartphones via Bluetooth, WiFi, USB or AUX.

The drive domain consists of ECUs that enable driving tasks, e.g., Electronic Stability Program (ESC) and motor controller that send acceleration or braking commands to the motor. In addition, the drive domain includes the Advanced Driver-Assistance Systems (ADAS) controller, which is connected to a variety of sensors and supports the driving task, e.g., by sensing the environment with respect to potential obstacles.

Finally, the diagnostic domain provides the On-board Diagnostics (OBD) interface. It is used by workshops to retrieve diagnostic messages from the vehicle network.

3.2. 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 buses of the previously defined reference architecture.

Use case 1: Electric driving. The electric driving process provides the basic major application for the ECUs and buses of the reference architecture. This involves the drive domain with its corresponding ECUs like brake, accelerator and the engine. Attacks performed during driving have different impacts: First of all, they pose a risk to life and limb of road users, i.e., they are highly safety-critical. Furthermore, such attacks also have a financial as well as operational impact if the EV is damaged or prevented from proper operation. In Section 6, we describe an example application that concentrates on especially safety-critical 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 use case of EVs is electric charging of the battery. This involves the energy management system with the battery as well as the external connection to the Charge Point (CP). This external communication is realized using ISO 15118 where a new part for 2nd generation network and application protocol requirements is currently under development [31]. The surrounding processes also include the provision of Original Equipment Manufacturer (OEM) and contract certificates. The charging use case has several different implications, consisting of 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 (e.g., generation of movement profiles). Furthermore, attacks against a proper charging process also have an operational impact since the EV cannot be properly operated if the charging process is not correctly completed. In Section 6, we concentrate on attacks against the proper establishment, use, and termination of the PnC communication between EV and CP. These attacks are dominated by the operational impact.

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. Furthermore, an attacker can manipulate the firmware update such that software errors cannot properly be fixed which results in a further prevention of the vehicle from intended operation. By this way, these attacks may have safety, operational as well as financial implications. In Section 6, we use this use case with an attacker trying to install a manipulated update to unlock features, i.e., to use these features without payment. In this case, the financial impact is the dominating factor.

4. Attack surface assessment

In this section, we describe our attack surface assessment. For the assessment, we follow the approach of ISO/SAE DIS 21434 [5] (cf. Fig. 1 and the example in Annex G). We focus on the identification of typical assets and the attack feasibility rating for single attack building blocks on these assets. We then discuss how the single attack building blocks can be used to determine the overall feasibility of a complete attack path. Finally, we show an approach to map different attacker types to our feasibility rating.

4.1. 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.

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) [32,33] 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 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).

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.

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.

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 (DoS)/ECU disabling, configuration change, flashing of malicious code, and execution of malicious code (possibly with escalated privileges).

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.

4.2. Attack feasibility rating

To determine the attack feasibility of our basic attacks, we utilized the scheme from [34], which is also recommended in ISO/SAE DIS 21434 [5]. It introduces the dimensions *Elapsed Time*, *Specialist Expertise*, *Knowledge of the Item or Component*, *Window of Opportunity*, and *Equipment*. For short notation, these dimensions will be called *time*, *expertise*, *knowledge*, *opportunity* and *equipment* in all subsequent mathematical formulas.

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.

According to the recommended value scale from annex I of ISO/SAE DIS 21434 [5], numerical values are assigned to the previously listed quantitative values of the dimensions.

The combination of all ratings, performed by summing up the numerical values, 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*. A recommended value scale for retransforming the numerical values into these quantitative values is again given in annex I of ISO/SAE DIS 21434 [5]. Our rating of the different basic attacks is listed in Table 1. The table lists all basic attacks on assets with respect to our defined trust boundaries (cf. Fig. 2).

4.3. Discussion on different path calculation methods

To describe complex attacks, the basic attacks introduced in the last section can be used to generate attack paths. Various functions can be used to calculate the feasibility of attacks along the generated paths.

Sum (sum up all values per category along the path). In this approach is that the accumulated values reach the limits of the model more easily, so that a fine-grained differentiation between longer attack paths is no longer possible. In addition, a short but difficult path results in a lower score than a long but simple path.

Average (average per category along the path). Here, attack paths with mostly moderate values are evaluated equally with attack paths with strongly varying values. However, while an attack path with mostly moderate values is actually feasible for an attacker to be executed successfully, an attack path with only one high value, e.g., breaking state-of-the-art cryptography, is very unlikely to happen.

Maximum (maximum values per category along the attack path). This allows a fine-grained differentiation of longer attack paths while at the same time recognizing difficult attacks with low feasibility. However, long medium-scoring paths, as opposed to short high-scoring paths, may have less aggregated feasibility.

Hybrid weighted sum. In this approach, attacks are weighted according to their difficulty to overcome the drawbacks from the *Sum* approach. The sum of each feature is calculated to the power of η, where η is a factor chosen with respect to the expected length of paths. For the attack paths this results in weight $= time^{\eta} + experience^{\eta} + knowledge^{\eta} + opportunity^{\eta} + equipment^{\eta}$ (for variables with assigned values, see explanations in Section 4.2). For example, if the attack graph has paths up to a length of 4, this results in a minimal weight of a feature to be 4. η is chosen so large that a difficult attack path of length 1 is weighted equally or higher than an easy attack path of full length. The short attack path has a weight of 2 so if we choose $\eta = 2$ both attacks have an equal weight so the length of an attack path gets less important with respect to the difficulty of the attack. This allows more difficult attacks to compete better with long paths. To derive the attack feasibility of a path, the maximum values occurring for the 5 dimensions along the path are summed up. The path with the lowest weight value is considered as the most feasible attack path. This approach combines the advantages of the *Maximum* and *Sum* approach.

Our method works with all functions described above, however, because of the drawbacks in some of them, we selected the *Hybrid Weighted Sum* model for the exemplary application that we describe in Section 6.

4.4. Threat agent identification

For our attack surface assessment, we identified the threat agents *Owner*, *Thief*, *Terrorist*, *Organized Crime*, *Mechanic*, *Hacktivist*, and *Foreign Government*. The threat agents with their properties are depicted in Table 2. We based the properties on [35] and adapted them to fit the ISO/SAE DIS 21434 rating categories. In particular, we aligned the rating categories for *Knowledge of the Item or Component* with ISO/SAE DIS 21434 ratings. Additionally, we incorporated the finances category into *Equipment* and made a reassessment of the ratings based on this adaptation.

The threat agents *Thief* and *Owner* are both financially motivated where the *Owner* additionally could be motivated by passion, e.g., illegally tuning his vehicle. Their capabilities regarding *Specialist Expertise*, *Knowledge of the Item or Component*, and *Equipment* are basic. Moreover, the *Owner* has unlimited access to his car while the *Thief* can only access the car as long as the *Owner* does not interrupt him, e.g., informing the police.

The *Terrorist* is motivated by ideology while his capabilities regarding *Specialist Expertise*, *Knowledge of the Item or Component*, and *Equipment* are also basic. However, we rated his *Window of Opportunity* as moderate because we assume that the potential to threaten the owner is higher for the *Terrorist*.

The threat agents *Organized Crime* and *Mechanic* are also financially motivated, but have more sophisticated capabilities regarding *Specialist Expertise*, *Knowledge of the Item or Component*, and *Equipment* than the threat agents described before. The *Window of Opportunity* is rated the same as the *Terrorist* because of the same threat potential. For the *Mechanic*, *Window of Opportunity* is rated easy since he may attack the vehicle undisturbed once it is in his garage for repair or maintenance.

Finally, the *Hacktivist* and *Foreign Government* are motivated by ideology and additionally passion for the *Hacktivist* and financial for the *Foreign Government*. In our model, they have the highest capabilities regarding *Specialist Expertise* and have high capabilities regarding *Equipment* and *Knowledge of the Item or Component*. Regarding the easy *Window of Opportunity*, the *Foreign Government* may (legally) seize a vehicle and may execute their attacks.

The model will allow to assign threat agents to the identified most feasible attack paths and make a more detailed risk assessment based on the threat agent's type, capabilities, and motivation.

5. Automated attack path generation

To extract attack paths, we developed a partially automated process that allows quick adaptation as well as less error-prone and faster results compared to a manual process of attack derivation. The automation to find the most feasible attack path in the vehicle consists of the working steps 1. Directed Weighted Graph Generation, 2. Attack Path Extraction, 3. Threat Path Extraction, and 4. Most Feasible Path Calculation. The process is illustrated in Fig. 3 and the single steps explained in the following.

The input for the automated process is a manually created system model. It is based on the possible attack steps we introduced in Section 4. For every attack step, we define preconditions and the location

Table 1
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
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
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
Wired on-car interfaces and communications							
7.1	Wired communications (corrupt/fake msg and info): USB	< 1 week	Proficient	Public	Easy	Standard	High
7.2	Wired communications (corrupt/fake msg and info): AUX	< 1 week	Proficient	Public	Easy	Specialized	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
8.7	On-car user hardware interfaces (access): USB	< 1 week	Layman	Public	Easy	Standard	High
8.8	On-car user hardware interfaces (access): AUX	< 1 week	Layman	Public	Easy	Standard	High
9.0	On-car communications (disable or DoS): 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

(*continued on next page*)

Table 1 (*continued*).

Id	Asset (attack)	Elapsed time	Specialist expertise	Knowledge of item/component	Window of opportunity	Equipment	Attack feasibility
10.3	Wired com. (intercept, alter, inject, replay): USB	< 1 week	Proficient	Public	Easy	Standard	High
10.4	Wired com. (intercept, alter, inject, replay): AUX	< 1 week	Proficient	Public	Easy	Specialized	High
10.5	Wired com. (intercept, alter, inject, replay): PLC	< 1 week	Proficient	Public	Easy	Specialized	High
10.6	Wired/Wireless com. (spoof): External test and diagnostic equipment	< 1 week	Layman	Public	Unlimited	Specialized	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
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
On-car ECUs							
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 DoS): Resource exhaustion of regular ECU	< 1 week	Proficient	Restricted	Unlimited	Standard	High
16.2	On-car ECU (disable or DoS): Shutdown/Halt	< 1 month	Proficient	Restricted	Unlimited	Standard	High
16.3	On-car ECU (disable or DoS): 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 DoS): Brake pedal position, throttle pedal position, steering angle sensor	< 1 week	Proficient	Restricted	Moderate	Standard	High
24.2	On-car sensors (disable or DoS): Ultrasonic, Lidar, Radar sensor	< 1 week	Layman	Public	Easy	Standard	High
24.3	On-car sensors (disable or DoS): 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

Table 2
Identified threat agents and their properties.
Source: Adapted from [35].

Threat agent	Motivation	Specialized expertise	Knowledge of item/component	Equipment
Owner	Financial, Passion	Layman	Public	Standard
Thief	Financial	Layman	Public	Standard
Terrorist	Ideology	Layman	Public	Standard
Organized Crime	Financial	Proficient	Restricted	Bespoke
Mechanic	Financial	Expert	Strictly Confidential	Specialized
Hacktivist	Ideology, Passion	Multiple experts	Confidential	Bespoke
Foreign Government	Financial, Ideology	Multiple experts	Restricted	Multiple bespoke

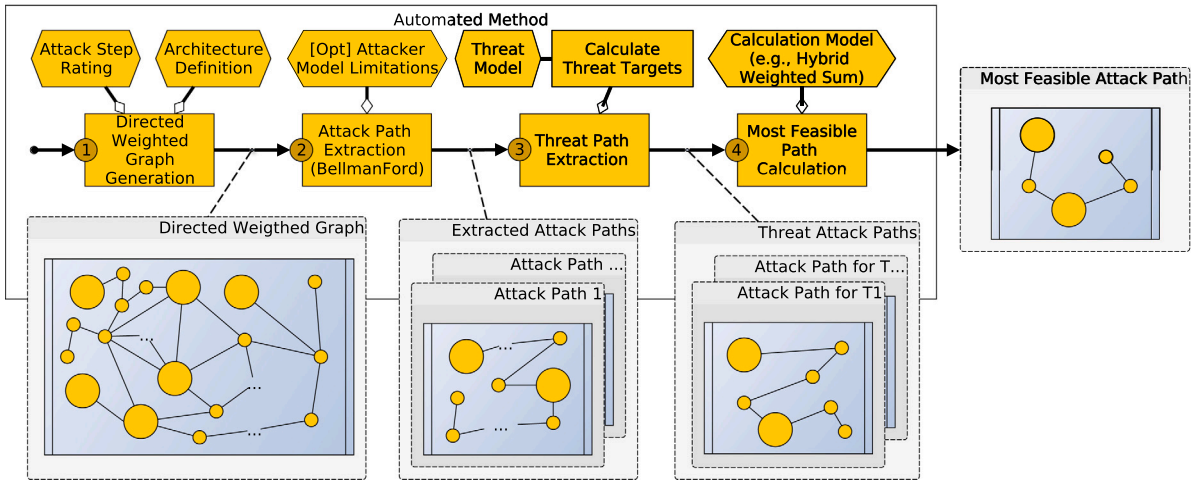

Fig. 3. Automated method.

of the attack based on the vehicle model (Section 3). For example, if an attacker wants to send a CAN-Bus message either a physical connection is necessary or the attacker has compromised an ECU connected to this bus. Every node is based on an attack step (e.g., 17.2 On-car ECU (configuration change): Physical) and a location (e.g., Headunit).

Example We illustrate our following description of our automated process with a very simple example with three attack steps a_1, a_2, and a_3. a_1 is rated (< 1 week, Expert, Public, Standard), a_2 (< 1 year, Expert, Public, Bespoke), and a_3 (< 6 month, Layman, Public, Specialized). An attacker in this model can perform attack a_1 or a_2 before being able to perform attack step a_3. a_3 is also the threat we are analyzing in this example.

1. Directed Weighted graph generation. From the manually defined system model, our automated method creates a directed graph by chaining the single attack steps together according to the defined preconditions and locations. Now, all attacks and the corresponding connections form a directed attack graph that represents all possible attack paths of our vehicle model. In preparation for the application of the BellmanFord algorithm in the next step, every possible entry point accessible to an attacker is connected to a virtual starting node ($vStart$). In the next step all edges, including the new connection starting with $vStart$ are weighted according to the attack feasibility of the attack step which is the destination of the edge. To convert the attack feasibility rating into a single number we use the *hybrid weighted sum* calculation model where η is the length of the longest loop-free path in the graph (cf. Algorithm 1).

2. Attack Path extraction. The attack graph from the last step is used in the following to calculate the shortest loop-free paths with respect to the weight of all attacks. These attacks can be calculated using the BellmanFord algorithm [36,37]. The result are all shortest paths in the directed attack graph starting at $vStart$. At this step, it is possible to restrict an attacker and for example, only consider remote attackers by not connecting physical attacks to $vStart$.

Example In our example the algorithm will connect a_1 and a_2 to $vStart$. The edges from between $vStart$ and a_1 and a_2 gets the weight of the difficulty of the attack steps. Using the BellmanFord algorithm we calculate all shortest paths from $vStart$. In this example these paths are $vStart \longrightarrow a_1$, $vStart \longrightarrow a_2$ and $vStart \longrightarrow a_1 \longrightarrow a_3$.

3. Threat Path extraction. In the next step, predefined threats are mapped to the path nodes that consist of attack step and location. A threat may contain different targets (e.g., an attacker can perform a DoS attack on a bus or shutdown an ECU for the same result). The shortest path to every target of a threat is extracted from the precomputed

Algorithm 1 Prepare shortest path in attack graph

```
G ← initialize AttackGraph
EntryNodes ← {}
vStart ← newNode
for all node ∈ G.nodes do
  if |G^T.edges[node]| = 0 then
    EntryNodes ← EntryNodes ∪ {node}
    G.edges[vStart].append(node, 0)
  end if
end for
G.nodes ← G.nodes ∪ {vStart}
for all edge ∈ G.edges do
  (v,u)=edge
  weight = u.time^η + u.experience^η + u.knowledge^η + u.opportunity^η + u.equipment^η
  edge.weight = weight
end for
shortestPaths ← BellmanFord(G.nodes, G.edges, vStart)
return shortestPaths
```

shortest paths (cf. Algorithm 2). First, the shortest path from the virtual start to the target is calculated. It is compared with the other shortest paths to other targets of the same threat. Finally, the shortest path for a threat is returned.

Algorithm 2 Extract shortest attack path of threat

```
Path ← ∅
for all target ∈ threat do
  T ← target
  Path_target ← target
  while T ≠ vStart do
    print T
    Path_target ← Path_target ∪ shortestPaths.predecessor[T]
    T ← shortestPaths.predecessor[T]
  end while
  if weight(Path_target) < weight(Path) then
    Path = Path_target
  end if
end for
return Path
```

4. Most Feasible path calculation. The set of shortest paths for a threat is finally used to calculate the feasibility of the most feasible attack for a threat. For this purpose, the maximal attack feasibility of every rating category is combined to gain the feasibility of the path.

Example In our example with the three paths $vStart \longrightarrow a_1$, $vStart \longrightarrow a_2$ and $vStart \longrightarrow a_1 \longrightarrow a_3$, there is only one target (a_3). Our method extracts all shortest paths from the graph which leads to a specified attack representing the threat. In this case the path to a_3, which is $vStart \longrightarrow a_1 \longrightarrow a_3$. Thus, the final attack has two attack steps a_1 with (< 1 week, Expert, Public, Standard) and a_2 (< 6 months, Layman, Public, Specialized). The resulting attack path is then rated (< 6 months, Expert, Public, Specialized).

6. Exemplary application on threats derived from the use cases

In this section, we apply our attack feasibility assessment to our use cases. In Section 6.1, we first select significant UNECE threats that are defined in regulation 155 "Cybersecurity and cyber security management system" [2], assign them to an overall threat scenario for each use case and define distinct attack targets. The attack targets represent the final attack step in an attack path in order to fulfill the threat scenario. For selected attack targets we detail attack paths in Section 6.2. Then we apply our feasibility rating and the outcome to the attack paths in Section 6.3. Additionally, Section 6.4 shows how the overall attack feasibility rating can be influenced by introducing dedicated security measures into the system. Finally, Section 6.5 contains a discussion of our approach with regard to other approaches for attack feasibility rating in relation to their eligibility for automotive applications.

6.1. Definition of threat scenarios

We define a distinct threat scenario for each use case and map it to a corresponding impact factor defined in ISO/SAE DIS 21434 that is dominant for this threat scenario. From the UNECE [2] we assign appropriate threats to the threat scenario and map specific attack targets (AT) to the UNECE threats that need to be successfully attacked by an attacker to achieve the threat scenario. The attack targets are mapped to the attack steps defined in Table 1. This is shown in Tables 3–5 and detailed in the following.

As depicted in Table 3, 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 a 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 DC Drive. In particular, the attacker needs to trick the engine into applying the wrong torque parameters. For this to happen, we identified the attack targets AT1–AT8 from the corresponding UNECE threats that are feasible to successfully perform the attack.

Thereby, AT1–AT3 consist of manipulating the sent torque data by either injecting new messages (AT1), replaying old messages (AT2), or altering sent messages (AT3). AT4–AT6 are about DoS attacks either by intercepting corrective torque messages sent from benign controller (AT4), congesting the channel with garbage messages (AT5), or disabling relevant ECUs (AT6). Finally, AT7 consists of taking over the ECU and send seemingly benign messages and AT8 manipulates sensor input.

As depicted in Table 4, 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 DC Energy, the CP, or the PLC communication in order to prevent the charging session from completion. Therefore, the attack targets AT1–AT7 from the corresponding UNECE threats have been defined. AT1–AT4 are about message manipulation, to either intercept messages like a "Charging Stop" (AT1), to replay old messages which normally occur in a charging session that keeps it alive (AT2), to inject new messages (AT3), or to alter already sent messages (AT4). AT5 means to take over an ECU and use it to send legitimate messages or to manipulate its process execution. AT6 and AT7 are about DoS attacks on either an ECU or a communication channel.

As depicted in Table 5, 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 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 targets AT1 and AT2 from the corresponding UNECE threats that are about flashing compromised firmware either remotely or physically.

6.2. Definition of attack paths

In this chapter, we describe exemplary attack paths generated with our algorithm for some of the previously identified attack targets. We chose $\eta = 3$ since an attack path in our model from an external interface to an internal target takes up to eight steps, thus the sum of low attack feasibility is 8. To prevent a physical attack with only one step but higher feasibility we choose a factor four since $2^3 \geq 8$. The amount and complexity of applicable paths depend on the attacker's capabilities and the system properties. For example, the attacker's knowledge about the system, e.g., the knowledge of an insider, may introduce new attack steps while the introduction of security measures may reduce possible attacks. The attack paths for the selected attack targets are described in the following.

Selected attack paths of UC1 are depicted in Table 6. For AT7 and AT8 of use case 1, we show exemplary attack paths AP_{UC1}^{AT7} and AP_{UC1}^{AT8} respectively. AP_{UC1}^{AT7} shows an attack first compromising the Diagnose DC via OBD interface (steps ①–③) and then hijacking the DC Drive to send seemingly benign torque messages on the CAN FD bus that are accepted by the motor controller (steps ④–⑤). In AP_{UC1}^{AT8}, an attacker compromises sensor input from the vehicle's environment to influence the ADAS into sending wrong commands to the engine (step ①).

Selected attack paths of UC2 are depicted in Table 7. For use case 2, we show the exemplary attack paths AP_{UC2}^{AT4} and AP_{UC2}^{AT5}. In AP_{UC2}^{AT4} our implemented automation method generated an attack path where an attacker physically accesses the CAN FD (step ①), brute forces the message authentication code of a CAN message that causes a locking of the charging cable (step ②), and then injecting this message on

Table 3
Threats for the UC1 threat scenario: Manipulate torque (Safety impact).

Threat from UNECE	Attack target (AT) for threat scenario
5. Communication channels used to conduct unauthorized manipulation, deletion or other amendments to vehicle held code/data	• AT1: Inject manipulated torque value (13.0) • AT3: Alter sent torque value (14.1)
6. Communication channels permit untrusted/unreliable messages to be accepted or are vulnerable to session hijacking/replay attacks	• AT2: Replay previously recorded torque value (12.0)
8. DoS attacks via communication channels to disrupt vehicle functions	• AT4: Intercept sent torque value (11.0)
9. An unprivileged user is able to gain privileged access to vehicle systems	• AT7: Takeover ECU and send torque value (21.0)
24. Disruption of systems or operations	• AT5: DoS channel (9.0) • AT6: Disable ECU to prevent sending of torque values (16.1)
32. Physical manipulation of systems can enable an attack	• AT8: Manipulate sensor value (25.2)

Table 4
Threats for the UC2 threat scenario: Immobilize car by never completing charging session (Operational impact).

Threat from UNECE	Attack target (AT) for threat scenario
5. Communication channels used to conduct unauthorized manipulation, deletion or other amendments to vehicle held code/data	• AT3: Inject messages to stay charging session (13.0) • AT4: Alter messages to stay in charging session (10.5)
6. Communication channels permit untrusted/unreliable messages to be accepted or are vulnerable to session hijacking/replay attacks	• AT2: Replay messages from a previously recorded charging session (12.0)
8. DoS attacks via communication channels to disrupt vehicle functions	• AT1: Intercept messages that would complete the charging session (11.0)
9. An unprivileged user is able to gain privileged access to vehicle systems	• AT5: Take over ECU to stay in charging session (21.0)
24. Disruption of systems or operations	• AT6: DoS channel (9.0), • AT7: Disable ECU to prevent from handling the charging session (16.1)

Table 5
Threats for the UC3 threat scenario: Flash compromised firmware to illegally unlock certain features (Financial impact).

Threat from UNECE	Attack target (AT) for threat scenario
12. Misuse or compromise of update procedures	• AT1: Flash compromised firmware via remote access (18.1) • AT2: Flash compromised firmware via physical access (19.1)

Table 6
Attack paths (APs) for selected attack targets (ATs) of UC1.

UC1 Threat Scenario: Manipulate torque (Safety impact)	
Identified ATs	AT1: Inject manipulated torque value (13.0), AT2: Replay previously recorded torque value (12.0), AT3: Alter sent torque value (14.1), AT4: Intercept sent torque value (11.0), AT5: DoS channel (9.0), AT6: Disable ECU to prevent sending of torque values (16.1), **AT7**: Take over ECU and send torque value (21.0), **AT8**: Manipulate incoming sensor values (25.2)
Selected APs	• $AP_{UC1}^{\mathbf{AT7}}$: ① 8.1 On-car interfaces (access — physical tampering): OBD → ② 15.2 On-car ECU (exploit vuln. or impl. error to access ECU): ECU with external interface (wired (OBD/PLC/USB/AUX))@DC Diagnosis → ③ 21.0: On-car ECU (execute Code/Commands)@DC Diagnosis → ④ 13.0: On-car communications (inject): Ethernet → ⑤ 15.3: On-car ECU (exploit vuln. or impl. error to access ECU): ECU with internal interface (wired(CAN/CAN FD/FlexRay/Ethernet))@DC Drive • $AP_{UC1}^{\mathbf{AT8}}$: ① 23.2: On-car Sensors (spoof of sensor Signal): Ultrasonic, Lidar, Radar Sensor@Lidar

the CAN FD bus (step ③). In AP_{UC2}^{AT5}, an attacker again compromises the Diagnose DC via OBD interface (steps ①–③), then compromises the DC Energy via Ethernet (steps ④–⑥), and finally takes over the Charge ECU via CAN FD to prevent the charging session process from completion (steps ⑦–⑨).

Selected attack paths of UC3 are depicted in Table 8. For each attack target of use case 3 we show the exemplary attack paths AP_{UC3}^{AT1} and AP_{UC3}^{AT2}. Attack path AP_{UC3}^{AT1} shows how an attacker could remotely flash new firmware onto the engine ECU to, e.g., increase its horsepower. The attacker compromises the TCU via wireless interface to install a malicious firmware (steps ①–⑤) and to inject messages into the network to update the firmware of the engine (step ⑥). In attack path AP_{UC3}^{AT2}, the attacker uses physical access to flash a compromised firmware via the debug interface of the ECU (steps ①–②).

Besides the examples listed here, the overall results show the majority of attacks used the OBD-Interface or other physical interfaces to perform attacks since the *Window of Opportunity* is weighted less than *Elapsed Time* or *Specialist Expertise* by the example in SO/SAE DIS 21434. If we restrict attacks to remote attacks only most attacks use the WiFi interface to enter the vehicle network since it needs less equipment than other interfaces (e.g., cellular). In both cases, the attack path chosen by our algorithm exploits multiple domain controllers until it reaches its final target. This allows the introduction of specific countermeasures to increase the difficulty for the attacker (cf. Section 6.4).

6.3. Attack path analysis

By applying our calculation model, the attack paths AP_{UC1}^{AT7} and AP_{UC1}^{AT8} of UC1 both achieve an overall feasibility rating of *Medium* (cf. Table 9).

A closer look at the individual ratings reveals that there is a difference in both paths regarding the values *Elapsed Time* and *Window of Opportunity* that even out in the overall rating. In contrast to AP_{UC1}^{AT8}

Table 7
Attack paths (APs) for selected attack targets (ATs) of UC2.

UC2 Threat Scenario: Immobilize car by never completing charging session (Operational impact)	
Identified ATs	AT1: Intercept messages that would complete the charging session (11.0), AT2: Replay messages from a previously recorded charging session (12.0), AT3: Inject messages to stay in charging session (13.0), **AT4**: Alter messages to stay in charging session (10.5), **AT5**: Take over ECU to stay in charging session (21.0), AT6: DoS channel (9.0), AT7: Disable ECU to prevent from handling the charging session (16.1)
Selected APs	• $AP^{\mathbf{AT4}}_{UC2}$: ① 8.3: On-car interfaces (access — physical tampering): CAN FD → ② 1.8:Brute Force SecOC@CAN FD ③ 14.1 On-car communications (alter): CAN FD • $AP^{\mathbf{AT5}}_{UC2}$: ① 8.1 On-car interfaces (access — physical tampering): OBD → ② 15.2 On-car ECU (exploit vuln. or impl. error to access ECU): ECU with external interface (wired (OBD/PLC/USB/AUX))@DC Diagnosis → ③ 21.0: On-car ECU (execute Code/Commands)@DC Diagnosis → ④ 13.0: On-car communications (inject): Ethernet → ⑤ 15.3: On-car ECU (exploit vuln. or impl. error to access ECU): ECU with internal interface (wired(CAN/CAN FD/FlexRay/Ethernet))@DC Energy → ⑥ 21.0: On-car ECU (execute Code/Commands)@DC Energy → ⑦ 13.0 On-car communications (inject): CAN FD → ⑧ 15.3: On-car ECU (exploit vuln. or impl. error to access ECU): ECU with internal interface (wired(CAN/CAN FD/FlexRay/Ethernet))@Charge → ⑨ 21.0: On-car ECU (execute Code/Commands)@Charge

Table 8
Attack paths (APs) for selected attack targets (ATs) of UC3.

UC3 Threat Scenario: Flash compromised firmware to illegally unlock certain features (Financial impact)	
Identified ATs	**AT1**: Flash compromised firmware via remote access (18.1), **AT2**: Flash compromised firmware via physical access (19.1)
Selected APs	• $AP^{\mathbf{AT1}}_{UC3}$: ① 6.3: On-car wireless interfaces (access): WiFi → ② 3.1: Wireless communications (corrupt/fake msg and info): WiFi → 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 • $AP^{\mathbf{AT2}}_{UC3}$: 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

Table 9
Individual and overall resulting rating tuples for the identified attack paths of UC1.

Attack Path AP^{AT7}_{UC1}						
Id	Elapsed time	Specialist expertise	Knowledge of item/component	Window of opportunity	Equipment	Attack feasibility
8.1	(<1 week	Layman	Public	**Easy**	Standard	High)
15.2	(**<6 months**	**Proficient**	**Restricted**	Unlimited	**Specialized**	Medium)
21	(<1 month	Proficient	Public	Unlimited	Standard	High)
13	(<1 week	Proficient	Public	Unlimited	Standard	High)
15.3	(<6 months	Proficient	Restricted	Unlimited	Standard	High)
∑	(<6 months	Proficient	Restricted	Easy	Specialized	**Medium**)
Attack Path AP^{AT8}_{UC1}						
23.2	(**<1 week**	**Proficient**	**Restricted**	**Moderate**	**Specialized**	Medium)
∑	(<1 week	Proficient	Restricted	Moderate	Specialized	**Medium**)

The values responsible for the overall attack path feasibility are typeset in bold.

where the attack can be applied immediately, e.g., by manipulating sensor readings with a laser pointer [38], AP^{AT7}_{UC1} requires a longer preparation time since it involves two privilege escalation attacks where first vulnerabilities in the system need to be found and exploited. However, this is compensated by the easier *Window of Opportunity* in AP^{AT8}_{UC1} where the attack can be persisted once executed successfully, while in AP^{AT8}_{UC1} the attacker needs to be constantly in range of the vehicle to spoof the sensor readings.

Both attack paths require *Specialist Expertise*, *Knowledge of the Item or Component*, and *Equipment* of at least *Proficient*, *Restricted*, and *Specialized*. Thus, the attacks can be executed by our attack agents *Organized Crime*, *Mechanic*, *Hacktivist*, and *Foreign Government*.

In UC2 both attack paths require that the attacker is in range of the vehicle to tamper with the OBD interface. In AP^{AT4}_{UC2}, the attacker then alters valid CAN messages by brute forcing SecOC while in AP^{AT5}_{UC2} the attacker takes over ECUs by exploiting vulnerabilities and then sends valid commands (cf. Table 10).

AP^{AT4}_{UC2} involves breaking cryptographic algorithms and thus requires a *Specialist Expertise* of *Expert* and has a *Window of Opportunity* of *Difficult* since the brute forcing approach cannot be easily persisted. However, no actual preparation time is needed, which results in a low *Elapsed Time* rating. The overall feasibility of AP^{AT4}_{UC2} is *Low* because of the required high expertise to break SecOC and the small *Window of Opportunity*. For AP^{AT5}_{UC2}, the dominant factors that lead to an overall feasibility rating of *Medium* are the lower requirements regarding *Specialist Expertise* and *Window of Opportunity*. The calculation is shown in Table 10.

Applying our threat agent model for AP^{AT4}_{UC2} and AP^{AT5}_{UC2}, at least an attacker of type *Mechanic* is required for AP^{AT4}_{UC2} while AP^{AT5}_{UC2} is also executable by *Organized Crime*.

Finally, in attack paths AP^{AT1}_{UC3} and AP^{AT2}_{UC3} of UC3 the attacker flashes a compromised firmware either remotely or via physical access by using the debug interface of the corresponding ECU (cf. Table 11).

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^{AT2}_{UC3}. In the

Table 10
Individual and overall resulting rating tuples for the identified attack paths of UC2.

Id	Elapsed time	Specialist expertise	Knowledge of item/component	Window of opportunity	Equipment	Attack feasibility
Attack Path AP^{AT4}_{UC2}						
8.3	(**< 1 week**	Proficient	**Restricted**	Moderate	Standard	High)
1.8	(<1 week	Proficient	Restricted	**Difficult**	Standard	Medium)
14.1	(<1 week	**Expert**	Public	Unlimited	**Specialized**	High)
∑	(<1 week	Expert	Restricted	Difficult	Specialized	**Low**)
Attack Path AP^{AT5}_{UC2}						
8.1	(< 1 week	Layman	Public	**Easy**	Standard	High)
15.2	(**<6 months**	**Proficient**	**Restricted**	Unlimited	**Specialized**	Medium)
21.0	(<1 month	Proficient	Public	Unlimited	Standard	High)
13.0	(< 1 week	Proficient	Public	Unlimited	Standard	High)
15.3	(<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)
15.3	(<6 months	Proficient	Restricted	Unlimited	Standard	High)
21.0	(<1 month	Proficient	Public	Unlimited	Standard	High)
∑	(<6 months	Proficient	Restricted	Easy	Specialized	**Medium**)

The values responsible for the overall attack path feasibility are typeset in bold.

Table 11
Individual and overall resulting rating tuples for the identified attack paths of UC3.

Id	Elapsed time	Specialist expertise	Knowledge of item/component	Window of opportunity	Equipment	Attack feasibility
Attack Path AP^{AT1}_{UC3}						
6.3	(<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)
∑	(<6 months	Proficient	Restricted	Easy	Specialized	**Medium**)
Attack Path AP^{AT2}_{UC3}						
8.6	(**<1 month**	**Expert**	**Restricted**	**Difficult**	**Specialized**	Low)
19.1	(<1 week	Proficient	Restricted	Moderate	Specialized	High)
∑	(<1 month	Expert	Restricted	Difficult	Specialized	**Low**)

The values responsible for the overall attack path feasibility are typeset in bold.

overall feasibility rating AP^{AT1}_{UC3} achieves a rating of *Medium* while AP^{AT2}_{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.

The physical attack requires at least a threat agent type of *Mechanic*, while the remote attack can also be executed by *Organized Crime*.

This chapter showed that we can successfully apply our approach to the *Threat Scenario Identification* and *Attack path analysis* steps in ISO/SAE DIS 21434 (cf. Fig. 1). All analyzed attacks need at least capabilities with *Specialist Expertise* of *Proficient*, *Knowledge of the Item or Component* of *Restricted*, and *Equipment* of *Specialized*. Regarding our threat agent model, this corresponds at least to the threat agent type of *Organized Crime*.

6.4. Evaluation of security measures

In this chapter, we show by example how the proposed approach can be used to evaluate the effects of different additionally implemented security technologies on the overall attack feasibility rating.

Table 1 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.

We use the remote malware flash threat of UC3 to show the effects of the introduction of security technologies. Table 12 shows a comparison of the resulting overall feasibility 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 the initial system without additional security measures (AP^{AT1}_{UC3}, cf. Table 11). The following two sections show the adapted attack paths that result from the successive introduction of security technologies. The second section shows the adapted attack path for a system with a security level 1 ($AP^{AT1'}_{UC3}$) and the third section shows the adapted attack path for a system with a security level 2 ($AP^{AT1''}_{UC3}$).

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 TPMs as key storage for the SecOC keys into the system. The TPMs shield the keys against unauthorized access. The attacker is forced to extract the correct SecOC key from a TPM to make the attack persistent even

Table 12
Individual and overall resulting rating tuples for the modified attack path of UC3.

Id	Elapsed time	Specialist expertise	Knowledge of item/component	Window of opportunity	Equipment	Attack feasibility
Attack Path AP^{AT1}_{UC3} (No additional security mechanisms)						
6.3	(<1 week	**Proficient**	Public	**Easy**	Standard	High)
3.1	(<1 week	Proficient	Public	Easy	Standard	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)
∑	(<6 months	Proficient	Restricted	Easy	Specialized	**Medium**)
Attack Path $AP^{AT1'}_{UC3}$ (Security level 1)						
6.3	(<1 week	**Proficient**	Public	**Easy**	Standard	High)
3.1	(<1 week	Proficient	Public	Easy	Standard	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[a]	(**<3 years**	Proficient	Restricted	Unlimited	Standard	Medium)
∑	(<3 years	Proficient	Restricted	Easy	Specialized	**Low**)
Attack Path $AP^{AT1''}_{UC3}$ (Security level 2)						
6.3	(<1 week	Proficient	Public	Easy	Standard	High)
3.1	(<1 week	Proficient	Public	Easy	Standard	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)
1.4[a]	(**>3 years**	**Multiple experts**	**Confidential**	**Difficult**	**Bespoke**	Very Low)
18.2[a]	(<3 years	Proficient	Restricted	Unlimited	Standard	Medium)
∑	(>3 years	Multiple experts	Confidential	Difficult	Bespoke	**Very Low**)

The values responsible for the overall attack path feasibility are typeset in bold.
[a]Attack steps differ from the original attack path and result from the introduction of the security mechanisms.

after a system reboot. Therefore, a successful execution of an additional preceding attack step is necessary where the attacker exploits a physical bug of the TPM (1.4) to obtain the correct SecOC key. This additional attack step increases the difficulty of the overall attack because of the high requirements for *Elapsed Time*, *Specialist Expertise*, *Knowledge of the Item or Component*, and *Equipment*. Due to this, the overall feasibility drops to *Very Low* and since an additional hardware attack on the TPM is necessary, it cannot be done fully remotely any more. Thus, the attack 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* for the initial system to *Very Low* in a system with integrity protection mechanisms and TPMs as secure storage for the SecOC keys. In terms of our threat agent model, the requirements for an attack could be increased to the point where only our most sophisticated threat agent type, the *Hacktivist*, can perform the attack.

This shows that our approach is also applicable to reflect adaptations of the E/E architecture with regard to the attack surface assessment.

6.5. Discussion with respect to other rating approaches

Our proposed attack feasibility rating concept is based on the attack-potential based approach. However, attack-potential based approaches are not the only approaches for attack feasibility rating compliant with ISO/SAE DIS 21434. Other approaches are provided by ratings based on the Common Vulnerability Scoring System (CVSS) and attack vectors. In this section, we briefly discuss the advantages and disadvantages of these different approaches in the automotive context to elaborate, why we chose an attack-potential based approaches in this work.

Attack-potential based approaches. A major advantage of attack-potential based approaches is provided by the high flexibility and the detailed coverage of many different aspects relevant for the evaluation of attacks. On the other hand, even the prescriptions given by ISO/SAE DIS 21434 are very flexible and allow the adaption to individual needs. Furthermore, attack-potential based approaches are a largely accepted strategy in the automotive area. A very well-known method broadly accepted by the automotive industry is provided by the EVITA method [9,10], but many other approaches have been published as well. A disadvantage is provided by the high complexity of estimating all the different parameters. In addition, the mapping of individual conditions to the parameter values is not obvious, and the final results of the rating may depend on the individual expertise of the team performing the risk analysis. Furthermore, the already developed approaches in the automotive area do not completely comply with ISO/SAE DIS 21434.

CVSS based approaches. CVSS already provides a complete standard scheme with a value scale. For the CVSS scheme, the attack feasibility is called exploitability. An ISO/SAE DIS 21434 compliant approach just demands the usage of the base exploit metrics group with the four parameters attack vector, attack complexity, privileges required, and user interaction according to CVSS V 3.1 [39]. This approach provides the advantage that the estimation of the parameters is much easier to handle. Furthermore, the direct mapping to numerical values with always the same scale allows easier comparison of different CVSS based approaches. On the other hand, the CVSS scheme provides the disadvantage of much lower flexibility to be adapted to individual applications. One great problem is provided by the important factor of attack complexity that only has two values low and high. Furthermore, the scheme is not well suited for the automotive area, in particular with respect to the parameters of required privileges and user interaction which are less relevant for this application area. Most attacks in the

automotive area work without any user interaction and different user privileges are only relevant for some specific use cases, e.g., in the area of Diagnostics.

Attack vector-based approaches. Pure attack vector-based approaches are even simpler than CVSS based approaches since they only consider this one parameter is also present in the CVSS scheme. ISO/SAE DIS 21434 only demands the consideration of the predominant attack vector. The usage of the CVSS scale is not required, but already recommended by the informative Annex I of [5]. This means that attack vector-based approaches are even less suited for automotive applications than CVSS based approaches.

Prioritization of our focus. If confronting the advantages and disadvantages of the different types of approaches for attack feasibility rating, the advantages of attack-potential based approaches, for which we set our focus, obviously predominate in the automotive area. The different types of attacks against EVs are very complex and manifold, and in Section 6.1 we have already seen some examples for UNECE threat scenarios with different attack endpoints; hence a flexible consideration of different parameters is very important. For CVSS based approaches, we have the problem that the attack complexity practically combines the parameters of elapsed time, expertise, equipment, and knowledge of item or component, and a rating with just the two different values of high and low must be done. In practical cases, the decision will be difficult whether the implementation of further security technology is considered necessary to reach the value low. The past development of different attack-potential based approaches in the automotive area, in particular the EVITA method, show up the eligibility of this type of approach for this application area. The previously mentioned obstacles of existing solutions can be easily overcome. The little lack of compliance to ISO/SAE DIS 21434 may be eliminated with slight adaptions of the schemes.

7. Conclusion

A TARA is an important part of an automotive cybersecurity engineering process. A central aspect for risk determination is the identification of the attack surface with a comprehensive feasibility assessment of possible attacks for each asset of a modern vehicle, for which we have defined a generic reference architecture. We are introducing a concrete approach for the assessment of the attack surface, comprising the steps of identifying assets, identifying threat scenarios, analyzing the attack path and evaluating the feasibility of attacks by a TARA in accordance with ISO/SAE FDIS 21434. Furthermore, we show a procedure how to automatize this process. Our attack feasibility assessment can be used in a TARA to assess an entire attack path of a threat scenario. Our implemented automation method can automate this step to make this easier, more efficient and less prone to error. A vehicle manufacturer can use the attack feasibility assessments in combination with an impact assessment to determine the risks according to ISO/SAE FDIS 21434. The automated process we have presented provides a less error-prone and faster alternative to the manual process of attack derivation. In addition, this automation enables quick adjustments to compare different architectures and security solutions. Based on our use cases and the UNECE 155 threat categories, we list sample threats for each use case. Our exemplary application for evaluating threat scenarios with corresponding attack paths for three use cases shows the feasibility of our approach and how our automation method contributes to the automation of the process. In addition, with our automation method we have shown how the feasibility of attacks is reduced if certain security mechanisms are introduced into the system. This method for evaluating the attack surface enables the evaluation of various security and damage control technologies with regard to their advantages for a safer vehicle architecture.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research work has been partly funded by the German Federal Ministry of Education and Research (BMBF) and the Hessen State Ministry for Higher Education, Research and the Arts, Germany within their joint support of the National Research Center for Applied Cybersecurity ATHENE and by the BMBF projects VITAF, Germany (ID 16KIS0835) and SAVE, Germany (ID 16KIS1324). 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, Tech. Rep., ENISA, 2016.
[2] UN Regulation No. 155, Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system, 2021, [Online]. https://unece.org/sites/default/files/2021-03/R155e.pdf. (Accessed 30 April 2021).
[3] UN Regulation No. 156, Uniform provisions concerning the approval of vehicles with regards to software update and software updates management system system, 2021, [Online]. https://unece.org/sites/default/files/2021-03/R156e.pdf. (Accessed 30 April 2021).
[4] SAE International, Cybersecurity Guidebook for Cyber-Physical Vehicle Systems, Tech. Rep. J3061, SAE International, 2016.
[5] ISO/IEC, ISO/SAE DIS 21434 — Road Vehicles — Cybersecurity Engineering, International Organization for Standardization, Geneva, CH, 2020.
[6] A. Bolovinou, U. Atmaca, A.T. Sheik, O. Ur-Rehman, G. Wallraf, 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.
[7] C. Plappert, D. Zelle, H. Gadacz, R. Rieke, D. Scheuermann, C. Krauß, Attack surface assessment for cybersecurity engineering in the automotive domain, in: 2021 29th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, PDP, 2021, pp. 266–275.
[8] A. Chattopadhyay, K. Lam, Y. Tavva, Autonomous vehicle: Security by design, IEEE Trans. Intell. Trans. Syst. (2020) 1–15.
[9] The EVITA consortium, EVITA Threat and risk analysis, 2009, https://www.evita-project.org. (Last Accessed 13 January 2020).
[10] 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, G. Pedroza, Security requirements for automotive on-board networks based on dark-side scenarios, in: EVITA project, EVITA Deliverable D2.3, 2009, https://evita-project.org/deliverables.html. (Last Accessed 13 January 2020).
[11] ISO/IEC, ISO/IEC 18045:2008(E): Information technology – Security techniques–Methodology for IT security evaluation, International Organization for Standardization, Geneva, CH, 2008.
[12] HEAVENS consortium, HEAVENS - Healing vulnerabilities to enhance software security and safety, 2016, https://research.chalmers.se/en/project/5809. (Last Accessed 13 January 2020).
[13] A. Lautenbach, M. Islam, HEAVENS - Healing vulnerabilities to enhance software security and safety, 2016, http://autosec.se/wp-content/uploads/2018/03/HEAVENS_D2_v2.0.pdf. (Last Accessed 13 January 2020).
[14] C. McCarthy, K. Harnett, A. Carter, Characterization of Potential Security Threats in Modern Automobiles: A Composite Modeling Approach, National Highway Traffic Safety Administration, 2014.
[15] 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), Tech. Rep., ETSI, 2017.
[16] C. Schmittner, Z. Ma, C. Reyes, O. Dillinger, P. Puschner, Using SAE J3061 for automotive security requirement engineering, in: A. Skavhaug, J. Guiochet, E. Schoitsch, F. Bitsch (Eds.), Computer Safety, Reliability, and Security, Springer, Cham, 2016, pp. 157–170.
[17] G. Macher, H. Sporer, R. Berlach, E. Armengaud, 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.
[18] G. Macher, E. Armengaud, E. Brenner, C. Kreiner, A review of threat analysis and risk assessment methods in the automotive context, in: A. Skavhaug, J. Guiochet, F. Bitsch (Eds.), Computer Safety, Reliability, and Security, Springer International Publishing, Cham, 2016, pp. 130–141.

[19] A. Boudguiga, A. Boulanger, P. Chiron, W. Klaudel, H. Labiod, J. Seguy, RACE: RIsk analysis for cooperative engines, in: 2015 7th International Conference on New Technologies, Mobility and Security, NTMS, 2015, pp. 1–5.

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

[21] C. Maple, M. Bradbury, A.T. Le, K. Ghirardello, A connected and autonomous vehicle reference architecture for attack surface analysis, Appl. Sci. 9 (23) (2019) https://www.mdpi.com/2076-3417/9/23/5101.

[22] Y.G. Dantas, V. Nigam, H. Ruess, Security engineering for ISO 21434, CoRR abs/2012.15080 (2020) [Online]. http://arxiv.org/abs/2012.15080.

[23] Informal working group on cyber security and over-the-air issues and endorsed by the working party on automated/autonomous and connected vehicles (GRVA), in: Proposal for the Interpretation Document for UN Regulation No. [155] on uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system, 2020. [Online] https://unece.org/fileadmin/DAM/trans/doc/2020/wp29/WP29-182-05e.pdf. (Accessed 6 May 2021).

[24] B. Schneier, Attack trees: Modeling security threats, Dr. Dobb's J. (1999).

[25] J. Wang, J.N. Whitley, R.C.-W. Phan, D.J. Parish, Unified parametrizable attack tree, Int. J. Inf. Secur. Res. 1 (2011) 20–26.

[26] C.A. Phillips, L.P. Swiler, A graph-based system for network-vulnerability analysis, in: NSPW '98, Proceedings of the 1998 Workshop on New Security Paradigms, September 22–25, ACM Press, Charlottsville, VA, USA, 1998, pp. 71–79.

[27] R. Rieke, Abstraction-based analysis of known and unknown vulnerabilities of critical information infrastructures, Int. J. Ind. Syst. Eng. 1 (2008) 59–77.

[28] P.K. Manadhata, J.M. Wing, A formal model for a system's attack surface, Mov. Target Def. (2011) 1–28.

[29] J. Petit, S.E. Shladover, Potential cyberattacks on automated vehicles, IEEE Trans. Intell. Trans. Syst. 16 (2) (2015) 546–556.

[30] M. Dibaei, X. Zheng, K. Jiang, R. Abbas, S. Liu, Y. Zhang, Y. Xiang, S. Yu, Attacks and defences on intelligent connected vehicles: A survey, Digit. Commun. Netw. 6 (4) (2020) 399–421.

[31] ISO/IEC, ISO/DIS 15118 — Road Vehicles – Vehicle to Grid Communication Interface – Part 20: 2nd Generation Network and Application Protocol Requirements, Tech. Rep., International Organization for Standardization, Geneva, CH, 2020.

[32] AUTOSAR, Specification of secure onboard communication - CP release 20-11, 2020, https://www.autosar.org/fileadmin/user_upload/standards/classic/20-11/AUTOSAR_SWS_SecureOnboardCommunication.pdf. (Last Accessed 13 January 2020).

[33] AUTOSAR, Specification of secure onboard communication protocol - AP release 20-11, 2020, https://www.autosar.org/fileadmin/user_upload/standards/foundation/20-11/AUTOSAR_PRS_SecOcProtocol.pdf. (Last Accessed 13 January 2020).

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

[35] D. Dominic, S. Chhawri, R.M. Eustice, D. Ma, A. Weimerskirch, Risk assessment for cooperative automated driving, in: Proceedings of the 2nd ACM Workshop on Cyber-Physical Systems Security and Privacy, CPS-SPC '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 47–58, https://doi.org/10.1145/2994487.2994499.

[36] R. Bellman, On a routing problem, Quart. Appl. Math. 16 (1) (1958) 87–90.

[37] L.R. Ford, Network Flow Theory, RAND Corporation, Santa Monica, CA, 1956.

[38] J. Petit, B. Stottelaar, M. Feiri, Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR, in: Black Hat Europe, 2015.

[39] Forum of Incident Response and Security Teams (FIRST), Common vulnerability scoring system version 3.1, 2019.

Daniel Zelle is a research associate at Fraunhofer Institute for Secure Information Technology SIT in the Department Cyber-Physical Systems Security. He received his Master of Science in computer science in 2015 at the University of Paderborn with a focus on embedded systems. Currently, his research is focused on the analysis and design of security protocols in vehicles, especially for in-vehicle and vehicle to infrastructure (e.g. charge station, repair shops or backend services) communication. In this context, he also investigates the protection of personal data generated by a vehicle.

Christian Plappert received his M.Sc. in IT Security from the Technical University of Darmstadt in 2016. Since then he is working as a research associate at the Fraunhofer Institute for Secure Information Technology in the department of Cyber–Physical Systems and Automotive Security in the Trustworthy Platforms research group. His current field of interest is the design and integration of trust anchor-based security solutions, in particular by utilizing the Trusted Platform Module (TPM), to enhance the security of automotive architectures, protocols, and services with Trusted Computing concepts.

Roland Rieke: Dr. rer. nat., Senior Scientist at the Fraunhofer Institute for Secure Information Technology. He has contributed to 46 peer-reviewed publications. His research interests focus on design principles for secure, scalable systems as well as model-based predictive security analysis. He has been involved in several European research projects such as EVITA, EFFECTS+, SecFutur, MASSIF and CITYCoP and is now working in the E-CORRIDOR project on machine learning based intrusion detection technologies for in-vehicle as well as multi-modal transport applications, in particular, behavior conformance tracking, security compliance tracking, and prediction of critical situations.

Dirk Scheuermann, Dr. rer. nat., finished his diploma in Mathematics at the Technical University of Darmstadt in 1994 and his Ph.D. at Justus-Liebig University Gießen in 1998. Both his diploma thesis and his Ph.D. thesis were done in cooperation with Fraunhofer SIT and dealt with cryptography. Since 1998, Dirk Scheuermann is working as a researcher at Fraunhofer SIT. His major interests are cryptography, smart card technology, data formats and protocols. Besides many other research projects, he also participated in the EU project EVITA. Dirk Scheuermann is currently involved in different projects in the area of automotive security and security for railway applications.

Christoph Krauß, Prof. Dr. Christoph Krauß is professor for Network Security at Darmstadt University of Applied Sciences and Head of Automotive Security Research at INCYDE GmbH, which he co-founded. At the National Research Center for Applied Cybersecurity ATHENE, he is Principal Investigator and coordinator of the research area Secure Autonomous Driving. He has over 15 years of experience in IT security. His research and interests include automotive security and privacy, railway security, intelligent energy networks security, trusted computing, network security, efficient and post-quantum cryptography, and security engineering.