Formal Modeling: A Step Forward to Cyber Secure Connected Car Systems

Branka Stojanović, Katharina Hofer-Schmitz, Kai Nahrgang, Heribert Vallant, and Christian Derler

1 Introduction

The paradigm *communicate with anyone anywhere at anytime* nowadays spans to cyber-physical systems in general and impacts many fields, including different types of industry (e.g., transportation, manufacturing, IT, etc.), health, and mobility [1]. An increase in connectivity demands, including a built-in connectivity, is reflected in a great deal by vehicle manufacturing industry. The established path and goal for automotive industry include connected cars and autonomous driving [2].

Existing and potential services in the connected cars industry should increase the road safety, bring more comfort to all passengers, and add more efficiency in traffic flows. Different sensors and services inside connected cars communicate and synchronize in order to enhance drivers' experience and make processes smoother. In addition to in-vehicle communication, connected cars communicate and interact with their environment, including other vehicles, roadside users, and external infrastructure and devices and even share processing efforts between other entities.

An increase in vehicles' connectivity demands influences in a great deal a rise of security issues. The ***security-by-design*** frameworks, including threat modeling and formal methods, have potential to respond to these challenges.

This chapter covers two main objectives – (i) a comprehensive overview of the *connected cars' communication architecture* and most important communication protocols under the V2X umbrella, with a special focus on the security perspective and (ii) *security-by-design* frameworks application within this domain, *threat modeling* state of the art methodologies and the ability to adapt those for the

B. Stojanović (✉) · K. Hofer-Schmitz · K. Nahrgang · H. Vallant · C. Derler
Joanneum Research Forschungsgesellschaft mbH, Graz, Austria
e-mail: Branka.Stojanovic@joanneum.at; Katharina.Hofer-Schmitz@joanneum.at; Kai.Nahrgang@joanneum.at; Heribert.Vallant@joanneum.at; Christian.Derler@joanneum.at

U. Z. A. Hamid, F. Al-Turjman (eds.), *Towards Connected and Autonomous Vehicle Highways*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-030-66042-0_6

automotive industry and *formal verification* tools and their applications in V2X protocols space. Additionally, it discusses challenges and future research directions, as research and development path within this industry. In Fig. 1, an illustration of a security-by-design procedure for in-vehicle communication is presented as research guideline. It includes threat modeling and formal verification based on inputs from previous steps and standards/specifications and their redesign according to findings.

The chapter is structured as follows: In Sect. 2, a basic introduction into the communication architecture and the protocols in use for connected cars is given. Section 3 focuses on threat modeling and describes different modeling methodologies and their possible applications, benefits, and limitations to model connected cars. Section 4 focuses on formal methods and describes used tools and considered protocols. In Sect. 5, key points are summarized, and open challenges for future work are stated.

2 Connected Cars Communication Architecture and Protocols

This section focuses on the communication architecture and most important automotive communication protocols from the security perspective.

The term vehicle-to-everything, commonly known by abbreviation V2X, encompasses all types of communications in the automotive domain, involving different types of communication entities, like vehicles, infrastructure units, motorcycles, cycles, pedestrians, etc. The heterogeneous connected car network consists of two main subnetworks [3] – intra-vehicle network, which covers a communication between in-vehicle devices, and inter-vehicle network, including the communication between the vehicle and surrounding.

2.1 Connected Car Network

Intelligent transportation system usually refers to the connected car system. It encompasses diverse entities and technologies, like vehicles, infrastructure units, and roadside users, and then data processing, communication and sensor technologies, etc. The heterogeneous network of such a system that connects different types of entities using different types of communication technologies consists of two main subnetworks [4]:

- *Intra-vehicle* network – covers a communication between in-vehicle devices, including controlling units, sensors, and actuators.
- *Inter-vehicle* network – commonly refers to the communication between vehicles; in this paper this term will be extended to all communication types among the vehicle and surrounding devices: on-board unit in-vehicle and external

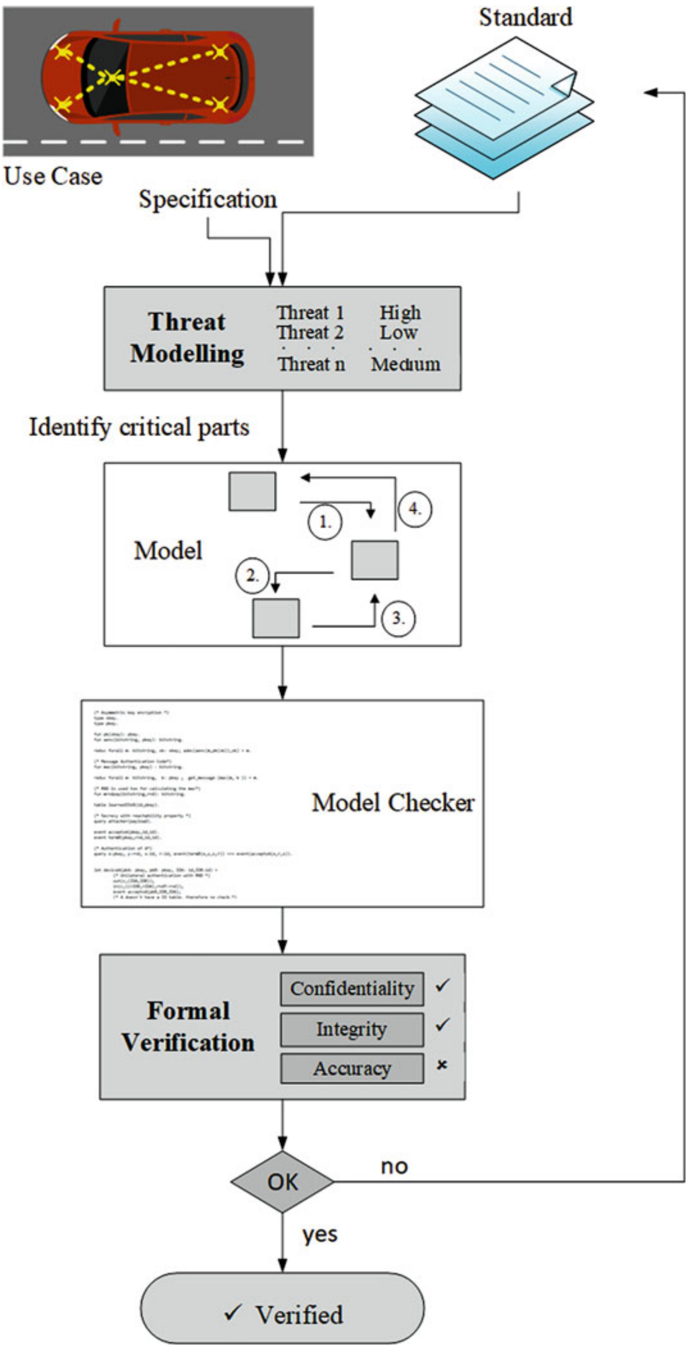


Fig. 1 Illustration of security-by-design framework within connected cars industry

entities, like roadside users (pedestrian, motorcyclists, etc.), infrastructure units, and central processing units (central/cloud server).

V2X on the other hand supports a unified connectivity platform for all connected end points and allows road entities to transmit information such as their current speed, position, and direction to the neighboring entities. It includes both *intra*- and *inter*-vehicle networks and can be categorized in different types of communication (Fig. 2):

- *In-vehicle* communication – represents the communication between entities in *intra*-vehicle subnetwork;
- *Vehicle-to-vehicle* (*V2V*) – covers the communication between vehicles, for example, the vehicle can broadcast the message of a pedestrian crossing the road to other vehicles, or the vehicle learns of another vehicle ahead braking suddenly and communicates this alert with other vehicles.
- *Vehicle-to-infrastructure* (*V2I*) – represents the communication between road entities and infrastructure units, for example, the vehicle can communicate with the traffic lights to know the speed at which he can drive to get green at the next traffic light, etc.
- *Vehicle-to-grid* (*V2G*) – supports the communication between vehicles and the electric grid, for example, plug-in electric vehicles communicate with the power grid to sell services on a return basis either by returning electricity to the grid or by throttling their charging rate;
- *Vehicle-to-pedestrian* (*V2P*) – provides the connection between the vehicle and vulnerable road users (VRU), including pedestrians, cyclists, and motorized two-wheeler operators; a typical *V2P* crash prevention system involves periodic exchange of safety messages among vehicles and VRUs [5].

This book chapter focuses on the security and safety perspectives of the most important automotive communication protocols, including in-vehicle communication and V2V and V2I protocols, because they address safety applications that are crucial for a rapid, robust, and timely performance, where any delay in message delivery could lead to a potentially fatal collision. Safety applications include various warnings (e.g., red light violations, curve speeds, reduced speed/work zones, emergency electronic brake lights, forward collisions, etc.) that are sent from their place of occurrence, picked by the closest vehicle, and then further propagated to the surrounding vehicles.

2.2 Intra-vehicle Communication

The interaction between various sensors and controlling units inside the vehicle requires an information exchange using specific communication protocols [2, 6].

Intra-vehicle communication usually involves LIN (Local Interconnect Network), CAN (Controller Area Network), FlexRay, MOST (Media Oriented System

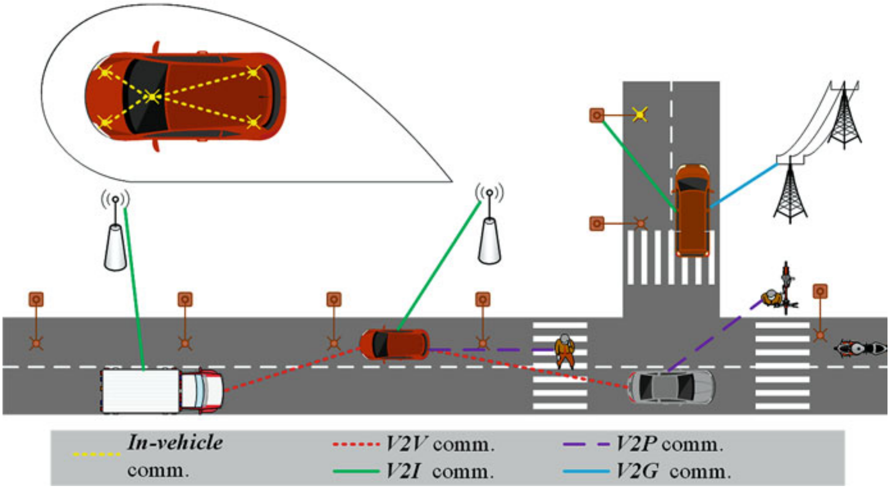


Fig. 2 Connected vehicle communication illustration

Table 1 *Intra*-vehicle protocols

Protocol	Data rate	Medium	Standard	Alliance	Year
CAN	1 Mbps	Twisted pair	ISO 11898	ISO	1991
MOST	150 Mbps	Optical fiber	Proprietary	MOST Coop. consortium	2001
LVDS	655 Mbps	Twisted pair	TIA/EIA-644	TIA	2001
LIN	19.2 kbps	Single wire	ISO 17987	LIN consortium	2002
FlexRay	20 Mbps	Twisted pair/optical fiber	ISO 17458	FlexRay consortium	2005
Automotive Ethernet	10 Mbps	Single twisted pair	IEEE802.3cg-2019	OPEN alliance	2019
	<10 Gbps	Single twisted pair	IEEE P802.3ch	OPEN alliance	tba

Transport), LVSD (Low-voltage differential signaling), or Automotive Ethernet. An overview of intra-vehicle protocols is presented in Table 1.

2.2.1 CAN: Controller Area Network

CAN [7] is a robust automotive-specific bus standard. It defines the functionality of the first two layers of the Open Systems Interconnection (OSI) network model – Layer-1 and Layer-2. CAN's design allows communication between different devices inside vehicles, including microcontrollers, or ECUs (electronic control units) [6]. CAN was first developed and released in 1986 by Robert Bosch GmbH. I1991 is the year of production of the first vehicle featuring this protocol [8].

CAN standard ISO 11898 was released in 1993 by the ISO – International Organization for Standardization. It was later restructured into two parts, with a third part released afterward. The most recent versions of those parts of ISO 11898 standard are as follows: (i) ISO 11898-1:2015[1] covering the data link layer and physical signaling; (ii) ISO 11898-2:2016[2] covering CAN, high-speed medium access units; (iii) ISO 11898-3:2006[3] covering CAN, the low-speed, fault-tolerant, medium-dependent interface.

Typical applications include the communication between ECUs controlling engine, power transmission, gearbox, antilock braking/ABS, electric power steering, etc. Beside passenger vehicles, it is used in trucks and buses, agricultural equipment, electronic equipment for aviation and navigation, building automation, medical equipment, industrial automation, etc. CAN bus is used in the on-board diagnostics (OBD)-II [9] vehicle diagnostics standard, as one of five supported protocols. CAN nodes are connected through a twisted pair bus, and data rates supported are up to 1 Mbps.

CAN is a low-level protocol and contains no direct support for security features. The implementations do not contain an encryption standard, and it leaves networks using CAN protocol open to cyber attacks, like man-in-the-middle frame interception and inserting messages on the bus. Security mechanisms are customized and usually implemented on the application and manufacturer level.

2.2.2 MOST: Media Oriented Systems Transport

MOST [10] is an automotive-specific high-speed multimedia network technology. It defines the physical and the data link layer as well as other layers of the ISO/OSI model of data communication. It was first introduced in 2001 by the MOST Cooperation consortium and has been implemented in ten vehicle models in the same year. The technology is nowadays used in almost every car brand, including Audi, General Motors, BMW, Hyundai, Honda, Lancia, Jaguar, Porsche, Mercedes-Benz, Land Rover, Toyota, Saab, Volkswagen, SKODA, Volvo, and SEAT. It is used to transport data signals, video, and audio inside vehicles. MOST nodes are connected via plastic optical fiber (POF) (MOST25, MOST150) or electrical conductor (MOST50, MOST150) physical layers, and it supports data rates up to 150 Mbps (MOST150).

[1] https://www.iso.org/standard/63648.html

[2] https://www.iso.org/standard/67244.html

[3] https://www.iso.org/standard/36055.html

The MOST protocol is secured by an automatically generated CRC sum (4 bytes) and ACK/NAK mechanism with automatic retry. There is no automatic retransmission in case of an error, and it has to be handled by the higher layers [10].

2.2.3 LVDS: Low-Voltage Differential Signaling

LVDS [11] is a technical standard that specifies high-speed signaling, using a differential, serial communication protocol. It specifies only the physical layer, while different data communication standards and applications that are built on top of it specify a data link layer of the ISO/OSI model.

The LVDS standard was defined in 2001, as ANSI/TIA/EIA-644-A standard.[4] It is used for high-speed video, graphics, video camera data transfers, and general-purpose computer buses. Although LVDS was not specifically developed for the automotive industry, its high-speed bandwidth of 655 Mbps over twisted-pair copper cable made it the top choice for automotive camera manufacturers. Besides automotive infotainment system, it is used in LCD-TVs, industrial cameras and machine vision, notebooks, tablets, etc.

LVDS protocol, as Layer-1 protocol, does not define any security mechanisms.

2.2.4 LIN: Local Interconnect Network

LIN [12] is an automotive-specific bus standard, defined as a cheaper alternative to CAN for less important components of the in-vehicle network [6], like the seats and steering wheel adjustment. It is a broadcast master-slave serial network protocol, which supports a data rate up to 19.2 kbps, via a single wire. Similar to CAN, it specifies the first two layers of OSI model.

The first fully implemented version of LIN protocol was specified in 2002, by the LIN Consortium, founded by five car manufacturers – BMW, Volkswagen Group, Audi, Volvo Cars, and Mercedes-Benz. It is standardized in the ISO 17987 series, where ISO/AWI 17987-8[5] is the standard defined for LIN over DC power line (DC-LIN).

LIN supports only error detection and checksums and faces similar risk exposures as CAN.

[4] https://www.ti.com/lit/an/slla038b/slla038b.pdf

[5] https://www.iso.org/standard/71044.html

2.2.5 FlexRay

FlexRay is an automotive-specific bus standard. Its advantages over CAN are higher reliability and speed, while disadvantage is additional cost overhead. Similar to the previously described bus standards, it specifies the first two layers of the OSI model – the physical layer and the data link layer.

The FlexRay Consortium developed it in 2009, mainly for high-performance onboard automotive computing applications, including drive electronics and safety (e.g., proximity control, active suspension, drive-by-wire, etc.). It comes with a bandwidth from up to 10 Mbps and uses unshielded cable pairs. The consortium, which later disbanded, included BMW, Volkswagen, Daimler, and General Motors (GM). FlexRay is specified in ISO 17458-1[6], 17458-2[7], 17458-3[8], 17458-4[9], and 17458-5[10] standards.

FlexRay, like other previously described bus protocols, was engineered in the absence of any security concerns. Therefore, it is highly vulnerable to adversarial attacks [13].

2.2.6 IEEE 802.3: Automotive Ethernet

Ethernet standard, commonly utilized as communication bus, is introduced to automotive industry as automotive Ethernet. The driving force for Ethernet usage in the automotive industry was primarily the high bandwidth. Additionally, the usage of UTP (unshielded twisted single-pair) cabling, its size, flexibility, and cost, also contributed to Ethernet applicability in vehicles. UTP cabling reduces network complexity and cabling costs and also contributes to free space and less weight of cars [6].

There are several revisions to the IEEE 802.3 standard that were made to fully meet the automotive requirements:

- IEEE 802.3bw[11]: 100BASE-T1 – 100 Mbps Ethernet over a single twisted pair for automotive applications, released 2015, superseded;
- IEEE 802.3bp[12]: 1000BASE-T1 – 1 Gbps Ethernet over a single twisted pair, automotive and industrial environments, released 2016, superseded;

[6] https://www.iso.org/standard/59804.html

[7] https://www.iso.org/standard/59806.html

[8] https://www.iso.org/standard/59807.html

[9] https://www.iso.org/standard/59808.html

[10] https://www.iso.org/standard/59809.html

[11] https://standards.ieee.org/standard/802_3bw-2015.html

[12] https://standards.ieee.org/standard/802_3bp-2016.html

- IEEE 802.3bv[13]: 1000BASE-RHx – 1000 Mbps Ethernet over plastic optical fiber (POF), intended for home, industrial, and automotive use, released 2017, superseded;
- IEEE 802.3cg[14]: 10BASE-T1 – 10 Mbps Ethernet over a single twisted pair, intended for automotive and industrial applications, released 2019, active;
- IEEE P802.3ch[15]: IEEE draft standard for multi-Gig automotive Ethernet (2.5, 5, 10 Gbps) over 15 m, release date tba, active.

Comparing to previously described bus standards, Automotive Ethernet, with high bandwidth gives leeway to better authentication or encryption mechanisms (e.g., Media Access Control (MAC)), and due to point-to-point characteristics of the Ethernet, a stricter separation into and within functional domains can be achieved, using Virtual Local Area Network (VLAN), Quality of Service (QoS), and firewall concepts [14].

2.3 **Inter-*vehicle Communication***

The interaction between vehicles and surrounding devices, including other vehicles and road side users, usually includes discussions about two types of protocols – WiFi based, often referred to as IEEE 802.11p from the name of the first standard designed to this scope, and cellular technologies including LTE-V2X and recently 5G, as part of the fourth generation of Third Generation Partnership Project (3GPP) standards and under the broader umbrella of the C-V2X (Cellular-V2X) [1]. Inter-vehicle protocols overview is presented in Table 2.

2.3.1 IEEE 802.11p

IEEE 802.11p[16] is the name of the first WiFi-based standard designed for V2X communication, released in 2010. Later, IEEE 802.11p was included in the IEEE 802.11-2012, which is afterward superseded by the IEEE 802.11-2016[17]. IEEE 802.11p defines the layer-1 (PHY) and layer-2 (MAC) layer protocols. A number of other standards have been defined above IEEE 802.11p standard, creating two different pillars – one in the USA, known as DSRC (dedicated short-range communication) and WAVE (wireless access in vehicular environments), and one in Europe, known as ETSI-ITS-G5 [1].

[13] https://standards.ieee.org/standard/802_3bv-2017.html

[14] https://standards.ieee.org/standard/802_3cg-2019.html

[15] https://standards.ieee.org/project/802_3ch.html

[16] https://standards.ieee.org/standard/802_11p-2010.html

[17] https://standards.ieee.org/standard/802_11-2016.html

Table 2 *Inter*-vehicle protocols

Protocol	Standard	Description	Status	Alliance	Year
IEEE 802.11p	IEEE 802.11p	Amendment 6: Wireless Access in Vehicular Environments	Supers.	IEEE	2010
IEEE 802.11p	IEEE 802.11	802.11-2016 – includes IEEE 802.11p functionalities	Active	IEEE	2016
C-V2X	3GPP Release 14	Mission Critical (MC) enhancements, LTE support for V2X services, IoT, voice and multimedia-related items, location and positioning items, etc.	Frozen	3GPP	2016
C-V2X	3GPP Release 15	New Radio (5G), 5G Phase 1, massive IoT, V2X Phase 2, MC networking with legacy systems, LTE improvements, etc.	Frozen	3GPP	2018
IEEE 802.11p	P802.11bd	Amendment: Enhancements for Next Generation V2X	Draft	IEEE	2018
C-V2X	3GPP Release 16	5G Phase 1, industrial IoT, V2X Phase 3, etc.	Frozen	3GPP	2019
IEEE 802.11p	IEEE 1609.12	1609.12-2019 – IEEE Standard for Wireless Access in Vehicular Environments (WAVE) Identifiers	Active	IEEE	2019
IEEE 802.11p	ETSI EN 302 663	ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency bands	Active	ETSI	2020
C-V2X	ETSI EN 303 613	LTE-V2X Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band	Active	ETSI	2020
C-V2X	3GPP Release 17	NR enhancements, enhanced V2X, unmanned aerial systems, etc.	Sched.	3GPP	2021

In the USA, IEEE 1609 standards define protocols below the application layer as wireless access in vehicular environments (WAVE), with IEEE 1609.12-2019[18] as the active version. In Europe, IEEE 802.11p was adopted by ETSI under the

[18] https://standards.ieee.org/standard/1609_12-2019.html

Cooperative Intelligent Transport Systems (C-ITS[19]) as ITS-G5[20], together with a large number of other documents dealing with all layers above it, dedicated to automotive ITS and Road Transport and Traffic Telematics (RTTT).

In May 2018, IEEE announced a new study group focused on the evolution of 802.11 technology for next-generation V2X communications. Their work resulted in publishing the amendment IEEE 802.11bd[21] later the same year. The ability to communicate for relative vehicle speeds of 250 kmph is a key feature of 802.11p. It operates in the licensed ITS band of 5.9 GHz with 10 MHz channel. IEEE 802.11p typically supports the range of 150–300 m. Its data rate is typically 6–27 Mbps, and it uses mesh network topology.

2.3.2 Cellular V2X

In parallel with IEEE 802.11p development, cellular technologies have been evaluated as long-range alternative. In 2016, 3GPP created the so-called C-V2X within Long-Term Evolution (LTE) Release 14[22]. It included a short-range interface that can be used also outside the cellular coverage and that poses an alternative to IEEE 802.11p [1].

In general, LTE is a wireless broadband communication standard designed for data terminals and mobile devices. It is based on the 2G/2.5G GSM/EDGE and 3G UMTS/HSPA technologies. LTE is specified in the 3GPP Release 8 and 9 document series, where Release 9 defines minor enhancements. It is also known as 4G LTE, Advance 4G, and 3.95G, since it does not meet the technical criteria of a 4G wireless service (defined in the 3GPP Rel. 8 and 9). In the beginning of 2020, ETSI published LTE-V2X[23] standard – LTE-V2X Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band.

LTE-V2X advantages include easy implementation – it can use existing cellular infrastructure. It supports relative speeds of up to 500 kmph [15]. It provides rates of 300 Mbps for downlink and 75 Mbps for uplink. A transmission range depends on application mode and can be up to 100 km in the radio network, while in Direct C-V2X applications, it is greater than 450 m. It provides a longer reaction time for driver, than in 802.11p communications [16].

[19] https://www.etsi.org/technologies/automotive-intelligent-transport

[20] https://www.etsi.org/deliver/etsi_en/302600_302699/302663/01.03.01_60/en_302663v010301p.pdf

[21] https://standards.ieee.org/project/802_11bd.html

[22] https://www.3gpp.org/release-14

[23] https://www.etsi.org/deliver/etsi_en/303600_303699/303613/01.01.01_30/en_303613v010101v.pdf

In 2018, 3GPP published the Release 15[24] that describes 5G NR (5G New Radio), including vehicle-to-everything communications (V2X) Phase 2. 5G is the successor of GSM (2G), UMTS (3G), and LTE and LTE Advanced Pro (4G).
The International Telecommunication Union Radiocommunication Sector (ITU-R) has lists following main uses for 5G:

- eMBB – Enhanced Mobile Broadband: an enhancement of 4G LTE mobile broadband services that includes more capacity, higher throughput, and faster connections;
- URLLC – Ultra-Reliable Low-Latency Communications: includes support for applications that requires uninterrupted and robust data exchange, like mission critical applications (deployment expected after 2021);
- mMTC – Massive Machine Type Communications: connects a large number of low-power and low-cost devices in a wide area; it should have increased battery lifetime and high scalability (deployment expected after 2021).

The three key frequency ranges for 5G spectrum, necessary to deliver widespread coverage and support all use cases, are:

- <1 GHz, which supports IoT services and provides widespread coverage across urban, suburban, and rural areas
- 1–6 GHz, expected to operate within the 3.3–3.8 GHz and to provide a good mixture of coverage and capacity benefits
- >6 GHz, expected to operate in 26 GHz and/or 28 GHz band, needed to meet the ultrahigh broadband speeds envisioned for 5G.

The targeted air latency in 5G is 1–4 ms. 5G should operate with throughput up to 10 Gbps, a hundred times faster throughput than 4G (LTE) speed of 100 Mbps. 5G Phase 2 is announced in 3GPP Release 16[25], with final submission planned for June 2020. It includes V2X Phase 3, with platooning, extended sensors, automated driving, and remote driving as main key points. More 5G system enhancements are set to follow in Release 17[26]. It is scheduled for delivery in 2021. Enhanced V2X services are announced in this release.

3 Threat Modeling

The increasing connectivity demands of various handheld devices, Internet of Things (IoT) and infrastructure assets together with the built-in automotive components, result in new threats from cyber space that are striking directly without any warning time. Therefore, theoretical modeling about the security status of a complex

[24] https://www.3gpp.org/release-15

[25] https://www.3gpp.org/release-16

[26] https://www.3gpp.org/release-17

system is becoming increasingly important. A theoretical modeling approach is threat modeling, which has the goal to identify potential threats and vulnerabilities based on the architecture of the given IT system. Conceptually different methodologies are used ranging from secure and agile application development to operative and business-driven concepts. Threat modeling is especially useful when applied during the design phase, as it delivers a semiformal security assessment which identifies security issues and the most likely attack vectors.

This section describes different threat modeling methodologies and their possible applications and limitations to model the domain of connected cars.

> Threat modeling is a process for identifying security issues for various IT systems. By using different methodologies, threat modeling can be used for plenty of scenarios and is not limited in any way of creating new methodologies or even adapting existing ones for new purposes. Hence, with regard to connected cars, the research community has already adapted well-established methods and achieved great results.

3.1 Threat Modeling Overview

Different threat modeling techniques have been developed addressing not only different aspects, like data and data flow, application and assets, and risk based or impact oriented but also different application areas like the software engineering or system architectures overall. However, more general threat modeling is split into two approaches [17]:

- Application Threat Modeling
- Operational Threat Modeling

The former is focusing on identifying threats of applications or IT architectures, which are represented using process-flow diagrams (PFD). These threats can then be addressed by software developers, software testers, as well as system architects and cyber security experts to work on mitigation. The latter are created from an attacker's point of view using data-flow diagrams (DFD). Operational threat models provide a visualization of the infrastructure's big picture in order to give a better view on the full attack surface. The result is usual used within (Sec)DevOps life cycles. Regardless of the approach and the application field, threat modeling is usually performed in four steps [18]:

1. Model system
2. Find threats
3. Address threats
4. Validate

While the first point is usually done using different software tools, the second point usually differs from the used threat modeling approach, which will be discussed in more detail in the following subsections. The third point then focuses on addressing the found threats by coming up not only with mitigation strategies but also, depending on the used approach, with a risk assessment. In the last step, a validation of the work done in point one to three should be performed.

The following subsections discuss several techniques of threat modeling, outlining the different aspects they address, in order to understand the different approaches.

3.1.1 ATT&CK

Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK) was created by MITRE and is used as a knowledge base and model for cyber adversary behavior. ATT&CK reflects the life-cycle phases of an adversary attack and the corresponding platforms. It can be used during various scenarios like red teaming or to improve defenses against network intrusion attacks. It started for Windows systems only, but now includes also Linux, macOS, cloud platforms, and mobile devices [19]. As stated in [20], the behavior model consists of three core models:

- Tactics, denoting short-term, tactical adversary goals during an attack (the columns);
- Techniques, describing the means by which adversaries achieve tactical goals (the individual cells);
- Documented adversary usage of techniques and other metadata (linked to techniques).

To illustrate an example, Table 3 [21] shows the ATT&CK Cloud Matrix. Since this technique has already been adapted for various platforms and systems, it could also be possible to adapt it for connected cars.

3.1.2 Attack Trees

Attack trees are a rather old but a still valid and valuable approach to discover threats in various environments. The concept was invented by *Bruce Schneier* [22] for modeling threats against computer systems. Attack Trees are not limited to computer systems, but in the information technology, they are a formal and methodical way to describe security threats based on possible attacks. *Shostak* describes in [18] three ways of using them: (1) use an attack tree created by someone else for your purposes; (2) create a tree specifically for your project; and (3) create a tree for general use, with the indent others will use it. Now, if you want to use a tree created by someone else, a modeled system is necessary first. Once this is done, the attack tree can be applied for each node of the model to see if the threat might

Table 3 ATT&CK cloud matrix

Initial access	Persistence	Privilege escalation	Defense evasion	Credential access	Discovery	Lateral movement	Collection	Exfiltration	Impact
Drive-by compromise	Account manipulation	Valid accounts	Application access token	Account manipulation	Account discovery	Application access token	Data from cloud storage object	Transfer data to cloud account	Resource hijacking
Exploit public-facing application	Create account		Redundant access	Brute force	Cloud service dashboard	Internal spearphishing	Data from information repositories		
Spearphishing link	Implant container image		Revert cloud instance	Cloud instance metadata API	Cloud service discovery	Web session cookie	Data from local system		
Trusted relationship	Office application startup		Unused/ Unsupported cloud regions	Credentials in files	Network service scanning		Data staged		
Valid accounts	Redundant access		Valid accounts	Steal application access token	Network share discovery		Email collection		
	Valid accounts		Web session cookie	Steal web session cookie	Permission groups discovery				
					Remote system discovery				
					System information discovery				
					System network connections discovery				

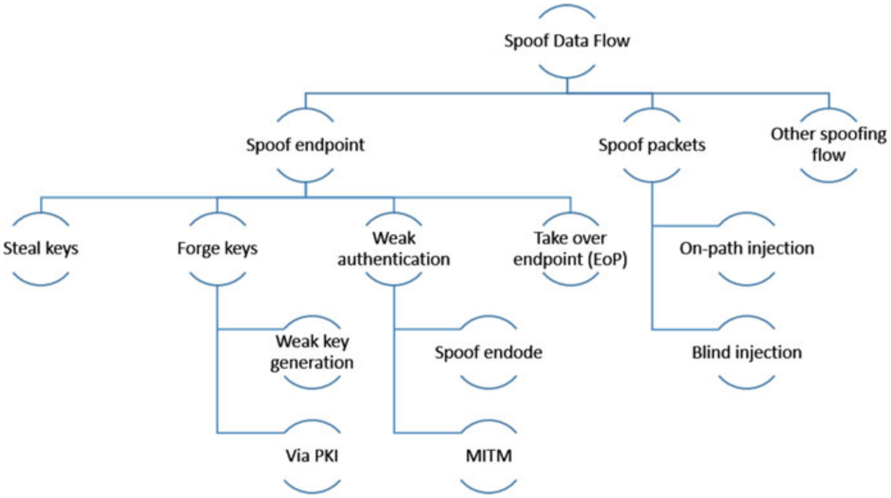


Fig. 3 Attack tree: spoofing of data flow

be applicable or not. To illustrate the approach, an example attack tree for spoofing data flow is given in Fig. 3.

The illustration shows that the root node is most properly the goal of the attack but might also represent a component of the system. If it represents a component, the subnotes should indicate what could get wrong. If the root node is the goal of the attack, the next steps show how to achieve it. When adding multiple subnotes, it is important to decide if the relationship between the nodes represents AND or OR. However, most of the time, the attack trees are using OR relationships. Once the goal of the tree as well as every single step how to achieve it is drawn, you should consider to prune the tree. This way, each node will once again be evaluated if it is duplicative or maybe even already mitigated by your system. Lastly, an attack tree should not exceed a single page in order to keep it clearly represented. If that is not the case, a tree might need to be split up into several trees.

To sum up, in order to create an attack tree, six steps need to be followed:

1. Decide on representation (AND or OR)
2. Create the root node (goal or components)
3. Create subnotes
4. Consider completeness
5. Prune the tree
6. Check the representation

As there are plenty of general attack trees which already can be applied to various projects, this approach can also be applied on automotive systems. Also, as this section discussed, there is always the possibility to create new, specifically for automotive vehicles, attack trees.

3.1.3 STRIDE

STRIDE was originally introduced by Microsofts' cyber security professionals Loren Kohnfelder and Praerit Garg as part of Microsoft's Security Development Lifecycle (SDL) concept. STRIDE uses data flow diagrams to describe the communication between processes and data stores in order to generate threats that are divided into the following six categories [23]:

- **S**poofing identity: A user or service illegally accesses and uses other authentication information to gain illegitimate access to a system or data.
- **T**ampering with data: Data tampering occurs when data is maliciously modified. This includes data at rest, data in use, as well as data in transit.
- **R**epudiation: This means that an entity may plausibly deny an action that it has taken. Countering these threats usually requires a combination of authentication, authorization, and logging, ideally in a cryptographically secured way.
- **I**nformation disclosure: Refers to any information exposed to unauthorized users.
- **D**enial of service (DoS): DoS attacks deny services availability to valid users.
- **E**levation of privilege: These threats occur when unprivileged users gain privileged access and, thus, are able to compromise an entire system.

The Microsoft Threat Modeling Tool[27] offers different templates for various scenarios and also gives the possibility to create new templates. STRIDE therefore has already been adapted also for the automotive domain [24–27] and hence will be discussed in Sect. 3.2.1.

3.1.4 TARA

Threat Agent Risk Assessment (TARA), developed by Intel, is a methodology that not only identifies threats but also shows which of them are most likely to occur [28]. This is achieved by focusing on threat agents, their motivations and methods. Threat agents represent attackers with certain capabilities of skills and resources. These properties of a threat agent are then mapped to methods that can occur, which might lead to possible threats. TARA is also taking a step further, by considering acceptable levels of corporate risks. This means that although an attack is likely to occur, the impact might be too little, and TARA might not identify this threat has a high-priority item.

TARA consists of three components:

- Threat agent library (TAL)[2]: The library identifies 22 threat agent archetypes including from internal employees to different kinds of external criminals.
- Common exposure library (CEL): The CEL includes common security vulnerabilities and exposures and maps them to known mitigations.

[27] https://www.microsoft.com/en-us/download/details.aspx?id=49168

- Methods and objective library (MOL): The MOL lists a set of methods on how threat agents usually plan to achieve their objectives.

When mapping these three components together, it becomes clear how the methodology works. As an example, Table 4 shows a sample from the MOL library. In Sect. 3.2.2, it is discussed how the approach and its components are adapted for the automotive industry.

3.2 Examples of Threat Modeling in the Automotive Industry

Since the previous section gave an overview about the different threat modeling methodologies, this section focuses on how to adapt threat modeling approaches for the automotive industry. Therefore, two examples on how popular approaches were already adapted in related works are given.

3.2.1 STRIDE for the Automotive Industry

The adaption of STRIDE is done using the template feature of the Microsoft Threat Modeling Tool. Here, a new template with regard to the automotive industry has been created by the NCC Group [26, 27]. The authors claim to provide following features:

- Processes and Data Stores related to connected cars;
- External Interactors tailored to an automotive system;
- Data Flows including over the air (OTA) and CAN bus;
- Trust Boundaries including vehicle-to-vehicle (V2V) networks;
- Known threats to components of connected cars, following the STRIDE categories.

Based on this template, a sample architecture of a connected car has been created and can be seen in Fig. 4. Here, a driver using a Human Machine Interface (HMI)/In-Vehicle Infotainment (IVI) and various sensors and cameras of the connected car is illustrated. These sensors and cameras are gathering information from the environment entity and are then passed to the Sensor Fusion Electronic Control Unit (ECU). The ECU sends the data to the gateway, which stores the data to the respective database. Also, a firmware update server and the respective data storage are drawn.

Figure 4 shows a simple example of a connected car threat model using an automotive industry template. A sample of the generated threats can be seen in Fig. 5, which shows newly added threats like tricking the sensor fusion ECU in into triggering an emergency stop, which, for example, would affect the safety of the vehicle, the passengers, and most probably also outside traffic participants.

Table 4 Sample from MOL library

Agent name	Attacker					Objective		Method									Impact				
	Access	Trust				Motivation	Goal	Acts					Limits								
		None	Partial trust	Employee	Administrator			Copy, Expose	Deny, withhold, ransom	Destroy, delete, render unavailable	Damage, alter	Take, remove	Code of conduct	Legal	Crimes against property	Crimes against people	Loss of financial assets	Business operation impact	Loss of competitive advantage, market share	Legal or regulatory exposure	Degradation of reputation, image, or brand
Employee error	Internal		X	X	X	Accidental/Mistake	No malicious intent, accidental	X		X	X		X				X	X	X	X	X
Reckless employee	Internal		X	X	X	Accidental/Mistake	No malicious intent, accidental	X		X	X			X			X	X	X	X	X
Information partner	Internal		X			Accidental/Mistake	No malicious intent, accidental	X		X	X						X	X	X	X	X
Competitor	External	X				Personal gain (Financial)	Obtain business or technical advantage	X							X				X		
Radical activist	External	X				Social/Moral gain	Change public opinion or corporate policy	X	X	X	X	X				X		X			X
Data miner	External	X				Personal gain (Financial)	Obtain business or technical advantage	X							X				X		
Vandal	External	X				Personal gain (Emotional)	Personal recognition or satisfaction			X	X				X			X			X
Disgruntled employee	Internal		X	X	X	Personal gain (Emotional)	Damage or destroy organization		X	X	X				X			X	X		X

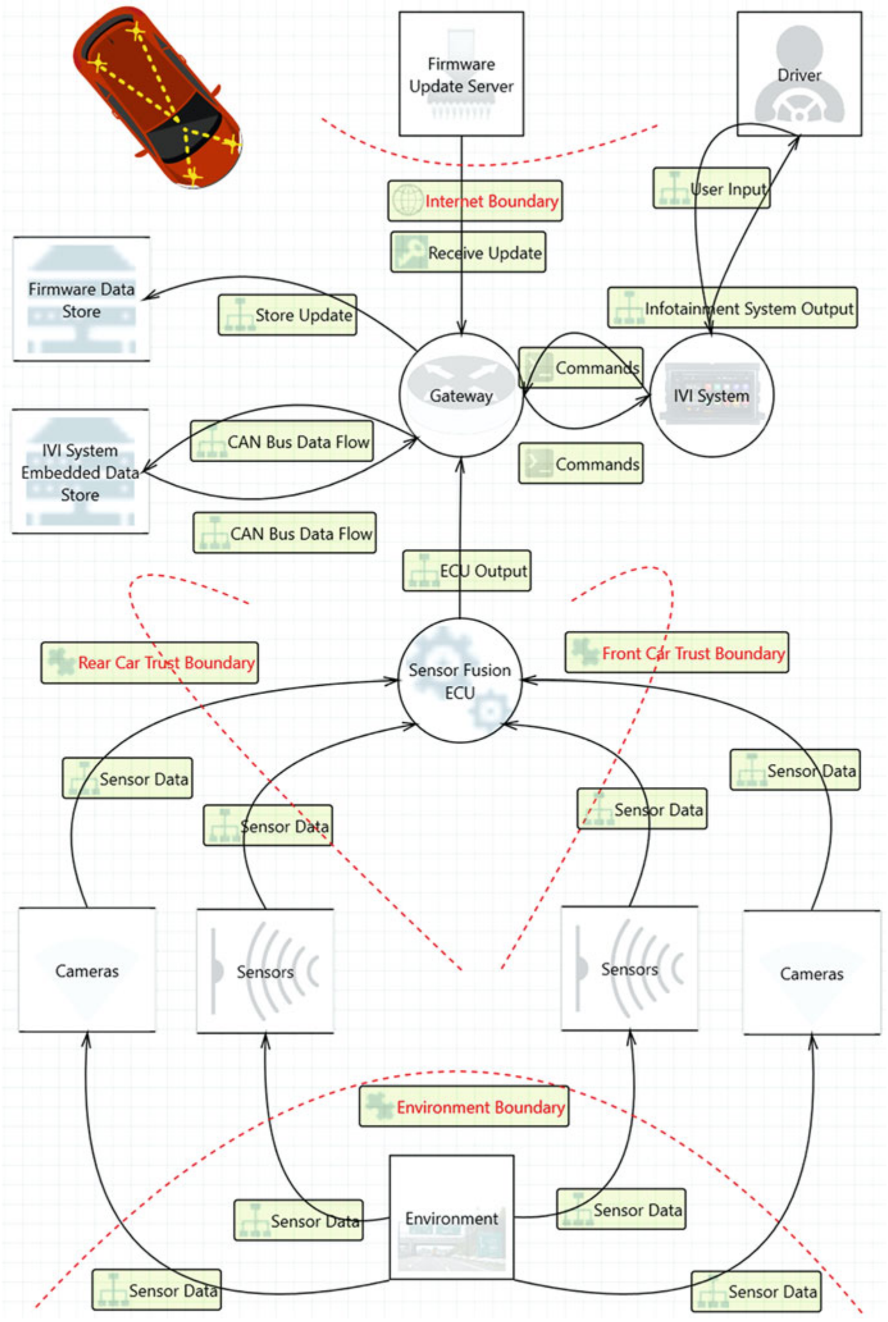
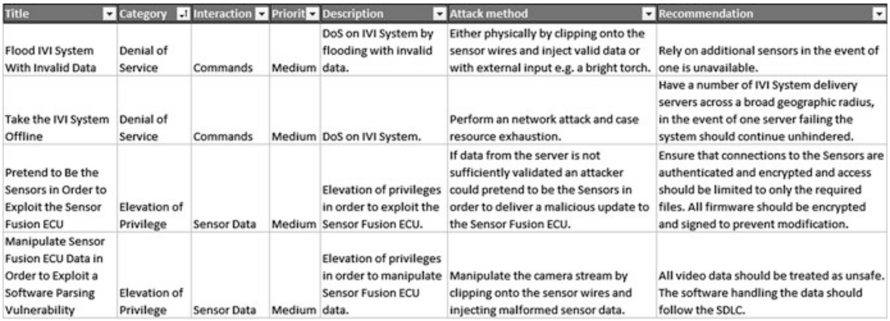


Fig. 4 Connected car: sample threat model



Title	Category	Interaction	Priorit	Description	Attack method	Recommendation
Flood IVI System With Invalid Data	Denial of Service	Commands	Medium	DoS on IVI System by flooding with invalid data.	Either physically by clipping onto the sensor wires and inject valid data or with external input e.g. a bright torch.	Rely on additional sensors in the event of one is unavailable.
Take the IVI System Offline	Denial of Service	Commands	Medium	DoS on IVI System.	Perform an network attack and case resource exhaustion.	Have a number of IVI System delivery servers across a broad geographic radius, in the event of one server failing the system should continue unhindered.
Pretend to Be the Sensors in Order to Exploit the Sensor Fusion ECU	Elevation of Privilege	Sensor Data	Medium	Elevation of privileges in order to exploit the Sensor Fusion ECU.	If data from the server is not sufficiently validated an attacker could pretend to be the Sensors in order to deliver a malicious update to the Sensor Fusion ECU.	Ensure that connections to the Sensors are authenticated and encrypted and access should be limited to only the required files. All firmware should be encrypted and signed to prevent modification.
Manipulate Sensor Fusion ECU Data in Order to Exploit a Software Parsing Vulnerability	Elevation of Privilege	Sensor Data	Medium	Elevation of privileges in order to manipulate Sensor Fusion ECU data.	Manipulate the camera stream by clipping onto the sensor wires and injecting malformed sensor data.	All video data should be treated as unsafe. The software handling the data should follow the SDLC.

Fig. 5 Connected car: sample of generated threats

Although most of the threats are created specifically for the automotive template, all of them are still categorized in the STRIDE categories.

3.2.2 TARA for the Automotive Industry

In order to adapt TARA for the automotive industry, *Karahasanovic et al.* in [29] extended the TAL and MOL with industry-specific threat agents, methods, and objectives. The adapted version of the TAL includes a total of 19 threat agents for the automotive industry, which all have 9 different attributes. When used by security experts during the first two steps of TARA, the library helps to determine which threat agents present the greatest risk to the system. Table 5 illustrates the adapted TAL, showing the 19 threat agents and their attributes.

Next, the Common Exposure Library is extended with all interfaces of a modern vehicle. Beside the interfaces, it also shows the impact level, the type of access, as well as the impact potential. Figure 6 illustrates the adapted library, which however is not complete as the properties might differ from various car manufactures.

Lastly, for the methods and objectives library (MOL), the “Acts” and “Limits” sections were removed, and the “Method” section was newly introduced, containing the values “Theft of PII and business data,” “Denial of Service,” “Intentional Manipulation,” “Unauthorized Physical Access,” and “Unpredictable Action.” These methods conclude most of the cyber attacks which threaten the connected vehicles. Furthermore, the attribute “Impact” has new impact levels: “reputation damage,” “privacy violation,” “loss of financial assets/car,” and “traffic accidents and injured passengers.” The updated MOL can be seen in Table 6.

Table 5 Adapted TAL

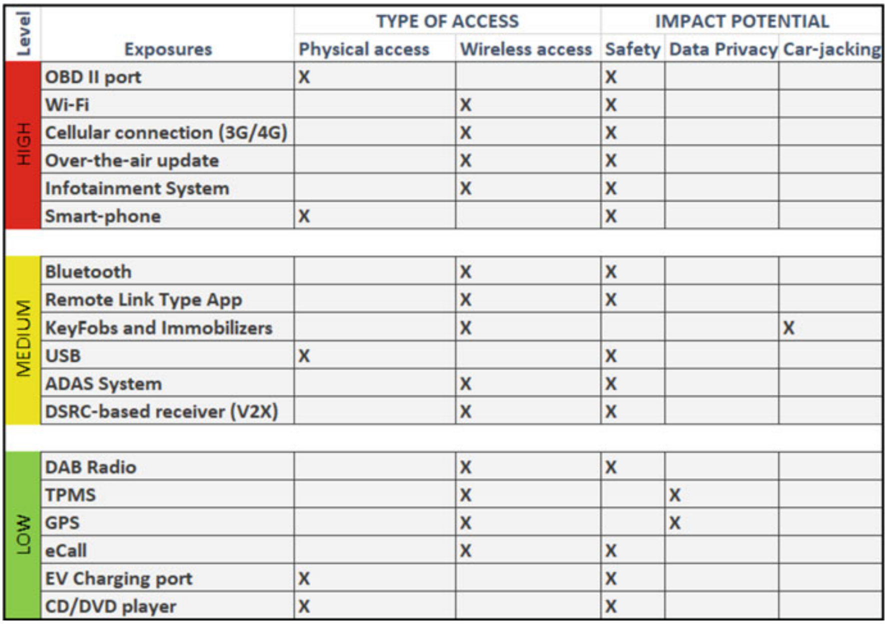
	Non-hostile intent				Hostile intent															
Threat agent attributes		Reckless employee	Untrained employee	Outward sympa-thizer	Information partner	Hackti-vist	Compet-itor	Cyber vandal	Data miner	Online social hacker	Script kiddies	Government cyber warrior	Organized crime	Radical activist	Sensatio-nalist	Cyber terror-ist	Cyber crimi-nal	Govern-ment spy	Internal spy	Disgr-untled empl-oyee
Access	Internal	X	X	X	X													X	X	X
	External					X	X	X	X	X	X	X	X	X	X	X	X			
Outcome	Acquisition/t							X		X	X		X				X		X	
	Business advantage						X		X									X		
	Material damage	X	X					X				X	X	X		X	X			X
	Harm to the passengers											X	X			X		X		
	Reputation damage	X	X	X	X	X		X			X	X		X	X					X
	Technical advantage						X		X									X	X	
	15 min of fame										X				X					
Resources	Individual	X	X	X													X			X
	Club										X									
	Contest							X												
	Team								X	X										
	Organization				X	X	X						X	X		X			X	
	Government											X						X		

Table 5 (continued)

		Non-hostile intent			Hostile intent															
Threat agent attributes		Reckless employee	Untrained employee	Outward sympathizer	Information partner	Hacktivist	Competitor	Cyber vandal	Data miner	Online social hacker	Script kiddies	Government cyber warrior	Organized crime	Radical activist	Sensationalist	Cyber terrorist	Cyber criminal	Government spy	Internal spy	Disgruntled employee
	None																			
Skills	Minimal		X								X				X		X			
	Operational				X			X		X										X
	Adept	X		X		X	X		X			X	X	X		X		X	X	
Visibility	Overt		X											X	X					
	Covert	X		X		X		X					X			X	X			
	Clandestine				X		X		X	X								X	X	
	“Don’t care”										X	X								X
Limits	Code of conduct		X		X															
	Legal	X																		
	Extra-legal – minor			X		X	X	X	X	X	X			X	X		X		X	
	Extra-legal – major											X	X			X		X		X

Table 5 (continued)

		Non-hostile intent			Hostile intent															
Threat agent attributes		Reckless employee	Untrained employee	Outward sympa-thizer	Information partner	Hackti-vist	Compet-itor	Cyber vandal	Data miner	Online social hacker	Script kiddies	Government cyber warrior	Organized crime	Radical activist	Sensatio-nalist	Cyber terror-ist	Cyber crimi-nal	Govern-ment spy	Internal spy	Disgr-untled empl-oyee
Objective	Copy					X	X		X	X								X	X	
	Deny											X								
	Injure											X	X	X		X		X		
	Destroy											X				X				X
	Damage			X								X				X				X
	Take												X			X				
	All above/Don't care	X	X		X			X			X			X	X		X			
Motivation	Accidental	X	X		X															
	Coercion																			
	Disgruntleme																			X
	Dominance							X				X								
	Ideology					X								X		X		X		
	Notoriety														X					
	Organizationa gain						X		X				X							
	Personal financial gain									X							X		X	
	Personal satisfaction			X							X									
	Unpredictabl																			



Level	Exposures	TYPE OF ACCESS		IMPACT POTENTIAL		
		Physical access	Wireless access	Safety	Data Privacy	Car-jacking
HIGH	OBD II port	X		X		
	Wi-Fi		X	X		
	Cellular connection (3G/4G)		X	X		
	Over-the-air update		X	X		
	Infotainment System		X	X		
	Smart-phone	X		X		
MEDIUM	Bluetooth		X	X		
	Remote Link Type App		X	X		
	KeyFobs and Immobilizers		X			X
	USB	X		X		
	ADAS System		X	X		
	DSRC-based receiver (V2X)		X	X		
LOW	DAB Radio		X	X		
	TPMS		X		X	
	GPS		X		X	
	eCall		X	X		
	EV Charging port	X		X		
	CD/DVD player	X		X		

Fig. 6 Adapted CEL

4 Formal Modeling and Verification

Another possibility to identify possible attacks and to minimize the attack vectors at an early stage is the use of formal verification methods. By using a diverse set of mathematical and logical methods, security guarantees with respect to a given model developed from, e.g., a protocol specification, an implementation or (parts of) a system can be obtained.

In general, there are two types of formal verification tools, *model checkers* and *theorem provers*. Model checkers are usually more restricted to a certain problem domain and the verification of properties in that field. Based on a given model and its specification, the dependent state space is automatically and exhaustively checked. Theorem provers are useable for a wider field of potential problem settings. However, they often need human expertise as the formulation of algebraic constrains or theorems to guide a proof of correctness [30, 31].

For *intra*- and *inter*-vehicle protocols, a wide variety of tools for formal verification are applied. Approaches include:

- techniques to prove functional correctness
- detection of possible attacks

(continued)

Table 6 Adapted MOL

Agent name	Attacker					Objective		Method					Impact				
	Access	Trust				Motivation	Goal										
		None	Partial trust	Employee	Administrator			Theft of PII and business data	Denial of service	Intentional manipulation	Unauthorized physical access	Unpredictable action	Reputation damage	Privacy violated	Loss of financial assets/car	Traffic accidents	Injured passengers
Competitor	External	X				Organizational gain	Technical advantage	X					X				
Car thief	External	X				Personal financial gain	Acquisition/Theft				X		X		X		
Cyber terrorist	External	X				Ideology	Physical harm; Damage			X						X	X
Cyber vandal	External	X				Dominance	Personal satisfaction	X	X	X			X	X	X	X	
Data miner	External	X				Organizational gain	Technical advantage	X					X	X			
Disgruntled employee	Internal		X	X	X	Disgruntlement	Reputation damage	X		X			X		X		
Government cyber warrior	External	X				Dominance	Physical harm; Damage	X	X	X						X	X
Government spy	Internal		X	X	X	Ideology	Technical advantage	X	X	X	X			X		X	X
Hacktivist	External	X				Ideology	Reputation damage	X					X	X			
Information partner	Internal		X			Organizational gain	Business advantage					X	X	X			
Internal spy	Internal		X	X	X	Personal financial gain	Acquisition/Theft	X					X	X	X		
Online social hacker	External	X				Personal financial gain	Acquisition/Theft	X							X		
Organized crime	External	X				Organizational gain	Acquisition/Theft	X	X	X	X			X	X	X	X
Outward sympathizer	Internal		X	X	X	Personal satisfaction	No malicious intent		X	X			X	X		X	
Radical activist	External	X				Ideology	Material damage	X	X	X			X	X		X	
Reckless employee	Internal		X	X	X	Accidental/Mistake	No malicious intent					X	X	X			
Script kiddies	External	X				Personal satisfaction	"15 Minutes of Fame"	X	X	X			X	X	X		
Sensationalist	External	X				Notoriety	"15 Minutes of Fame"	X					X	X			
Untrained employee	Internal		X	X	X	Accidental/Mistake	No malicious intent					X	X	X			

- considerations of the performance or worst-case scenarios

The focus of most of the publications is different depending on the protocols:

- *intra-vehicle protocols:* publications mainly focus on proving functional correctness and investigation of performance
- *inter-vehicle protocols:* especially for 5G, the focus is on security properties as secrecy and authentication.

Enhanced protocols are mainly considered for CAN, which do not provide authentication by default, and for 5G, where most of the work focus on 5G-AKA.

4.1 Formal Verification Tools Overview

Commonly used tools in literature are the security protocol model checkers `AVISPA, ProVerif, Scyther`, and `Tamarin`. In the class of probabilistic/statistical model checkers, the tools `UPPAAL` and `PRISM` are widely used (see also [32]). For those tools, a short description shall be given.

The push-button tool *AVISPA*[28] stands for Automated Validation of Internet Security Protocols and Applications. The tool suite contains different verification techniques as On-the-Fly model checker (OFMC), Constraint-Logic-based Attacker Searcher (CL-AtSE), SAT-based model checking (SAT = satisfiability problems in propositional logic), and tree automate-based automatic approximation [33] for the security's protocols analysis, applicable on the same protocol specification. Some of the techniques can deal with unbounded verification. Furthermore attack finding and visualization is supported.

The command-line tool *ProVerif*[29] was developed for the automatic analysis of the security of cryptographic protocols. It can handle an unbounded number of runs of a protocol. The analysis of (weak) secrecy properties can be performed via reachability properties, authentication properties by using correspondence assertions. Additionally, ProVerif can prove observational equivalence, which can, e.g., be used for proving strong secrecy or real or random secrecy. In case a proof fails, the tool assists in the reconstruction of an attack.

The tool *Scyther*[30] has a similar goal as ProVerif. Proofs can be obtained based on a symbolic representation of sets of protocol runs with the backward search

[28] http://www.avispa-project.org/

[29] https://prosecco.gforge.inria.fr/personal/bblanche/proverif/

[30] https://people.cispa.io/cas.cremers/scyther/

algorithm. Moreover, it can be used for attack finding and visualization, can handle an unbounded number of sessions, and additionally has a GUI.

Another similar tool is *Tamarin*,[31] which is both, a model checker and a theorem prover. Tamarin uses a generalization of Scyther's backward search, which makes it capable of handling protocols with non-monotonic mutable global states and complex control flows such as loops. Tamarin enables attack finding and visualization. It can deal with models such as the eCK model for key exchange protocols and equational theories as Diffie-Hellman. Moreover, it can handle bilinear pairings as well as user-specified subterm-convergent theories.

The toolbox *UPPAAL*[32] focuses on system's modeled as a collection of non-deterministic processes with finite control structures and real-value clocks, where the communication is performed via shared variables or through channels. Therefore, suitable application areas of the tool are, e.g., real-time controllers and communication protocols including critical timing aspects. The toolbox has three main parts, a description language, a simulator used for validation, and a model-checker based on timed automata theory.

The probabilistic model checker *PRISM*[33] developed at the University of Birmingham is intended for formal modeling and the analysis of systems that exhibit random or stochastic behavior. The tool can handle several probabilistic models as probabilistic automata and probabilistic timed automata, discrete-time and continuous-time Markov chains, as well as Markov decision processes. The underlying probabilistic verification techniques include quantitative abstraction refinement and symmetric reduction. Furthermore, the generation of optimal adversaries/strategies is supported.

4.2 Examples of Formal Modeling and Applications in Connected Cars

In this section, an overview of different intra- and inter-vehicle protocols where formal methods are applied is given. It extends our previous work in [2], provides more details, and addresses a wider range of protocols.

4.2.1 Intra-vehicle Protocols Formal Verification

For intra-vehicle protocols, formal methods are applied to the CAN, Automotive Ethernet, and FlexRay.

[31] https://tamarin-prover.github.io/manual/tex/tamarin-manual.pdf

[32] http://www.uppaal.org/

[33] https://www.prismmodelchecker.org

Table 7 *Intra*-vehicle protocols

Protocol	Tool(s)	Model	Properties of interest	Reference
CAN	SHVT	Four car components: fieldbus, telemetric ECU, backend-server, terminals representing clients	Replay messages, unlocking someone else's car, downgrading	[34]
CAN	UPPAAL	Focus on arbitration and transmission process and the fault confinement mechanism	11 properties out of the categories: safety, liveness, invariant	[35]
CAN/ Ethernet	CPA	Multiplexing strategies at gateways	Buffering, triggering, and mapping with focus on worst-case and end-to-end latency and load.	[36]
FlexRay	Isabelle/ HOL	FlexRay bus guardian component	Correct relay and integrity	[37]
FlexRay	CPN	AUTOSAR FlexRay transport protocol	Deadlock-free for selected configurations	[38]

Most of the work there are applications of formal method to specifications/standards in order to check selected properties. For CAN there are also enriched schemes/protocols checked. Additional security for the low-level protocol CAN is considered in [39], where an authentication protocol for CAN is presented. Furthermore, a clock synchronization service for CAN is proposed. An overview about different approaches is given in Table 7. Details on the existing approaches are given below.

Guergens et al. propose in [34] an abstract vehicle communication system model providing telemetric functions and onboard communication. It considers four car components, namely, the fieldbus, the telemetric ECU (electronic control unit), a backend server, and also terminals representing clients. As main attack points, the interface GSM/GPRS for a remote attacker and the fieldbus interface for a local attacker is considered. For formal modeling under the Dolev-Yao attacker model [40], the authors use Asynchronous Product Automata (APA), an operational description concept for cooperating systems. As tool, the Simple Homomorphism Verification tool (SHVT), providing components for the complete cycle from formal specification to exhaustive analysis and verification and supports APA, is used. The authors consider a real-world example, which is – with support from SHVT – analyzed for three different scenarios which differ by the foreknowledge of the attacker. There, especially *replay messages, unlocking someone else's car*, and *downgrading of security mechanisms* are taken into account. Additionally, a formal model of a fieldbus is given.

Pan et al. consider in [35] a formal verification with UPPAAL of the *CAN* bus protocol with a focus on the arbitration process, the transmission process, and the fault confinement mechanism. The authors formalize 11 properties, which can be divided into three categories, namely, *safety, liveness*, and *invariant*. The formal verification with UPPAAL shows that the main security issues of the CAN bus system are *deadlock, starvation, data inconsistency*, and *the fault confinement mechanism*. The authors state that the detected problems can at least be partly solved in the application layer.

Bruni et al. give in [39] a formal analysis of *MaCAN*. MaCAN is an authentication protocol developed in order to enable authentication in the CAN bus. By using ProVerif, the authors detected two flaws. The first one leads to unavailability during key establishment. The second one allows a re-using of authenticated signals for different purposes. The authors state that some aspects of MaCAN had to be adjusted (e.g., the usage of timestamps for ensuring message's freshness). However, it is stated by the authors that they could not express the freshness of timestamps in ProVerif, since ProVerif abstracts away the state information. Furthermore, the presence of an attack in their own implementation of the protocol is experimentally verified.

Rodriguez-Navas et al. apply in [41] model checking on a proposed clock synchronization service for the Controller Area Network (*CAN*) for highly synchronized clocks even in the occurrence of faults in the system. For modeling and verification, the tool UPPAAL is used. The model is based on timed automata, and a novel technique for drifting clocks is proposed. The author's solution achieves the desired precision event in case of the presence of various node and channel faults. Furthermore, their results indicate that inconsistent channel faults pose a big threat to clock precision. However, it is possible to reduce their negative impact by using a suitable resynchronization period.

Thiele et al. focus in [36] on an analysis of timing impact, which is introduced by various *CAN/Ethernet* multiplexing strategies at gateways. The authors state that the timing determinism of critical control and streaming data is crucial in the automotive network design. In particular, three different aspects of multiplexing are considered: buffering, triggering, and mapping. By using Compositional Performance Analysis CPA [42], the authors model and analyze three different multiplexing scenarios. In the evaluation, the authors focus on the effect of multiplexing on the design metrics worst-case and end-to-end latency and load. Furthermore, their analysis allows to capture and quantify differences between different multiplexing strategies.

Zhang considers in [37] the *FlexRay* bus guardian component. The bus guardian component helps to protect the communication channel against faulty behavior of communication controllers in FlexRay. The author uses Isabelle/HOL, a theorem prover for higher-order logic for specifying and verifying. The focus in the paper is on two properties of the bus guardian, namely, the *correct relay* and the *integrity*. In order to verify the properties, the correctness of the FlexRay clock synchronization is assumed.

Gordon and Choosang give in [38] a formal analysis of the AUTOSAR *FlexRay* Transport Protocol by using Colored Petri Nets, a mathematical modeling language. The authors prove that the FlexRay Transport Protocol is *deadlock-free* for certain configurations in case of delivering a single-protocol data unit from the sender to the receiver. Furthermore, it captures the desired service language. Moreover, it is stated by the authors that their results indicate the absence of

Table 8 *Inter*-vehicle protocols

Protocol	Tool(s)	Model	Properties of interest	Reference
IEEE 802.11p	PRISM	MAC protocol abstracted into four modules	Collision avoidance mechanism	[48]
5G	Tamarin	5G AKA	Confidentiality, authentication, privacy	[49]
5G	Tamarin	5G AKA, modelled all four parties involved in the protocol	Fine-grained analysis	[50]
5G	Scyther	5G-EAP-TLS, mutual authentication between subscribers and home network	Secrecy of SUPI and session key, non-injective synchronization of events, non-injective agreement on data	[51]
5G	ProVerif	5G-EAP-TLS, severing network and home network are considered as single entity	Authentication and secrecy statements	[52]

functional errors in the protocol specification and that the protocol is likely error-free.

4.2.2 Inter-vehicle Protocols Formal Verification

Formal verification is considered for the MAC of IEEE 802.11p and a wide variety of approaches for 5G (see Table 8). For 5G especially, 5G-AKA and 5G-EAP-TLS are considered. Furthermore, there are several approaches to formally verify enhanced versions of 5G in general – not focusing on the automotive domain explicitly – as [43–47]. Details of the approaches are given below. A review of formal verification method approaches considering 5G is also given in [32].

Zou et al. in [48] consider the Media Access Control (MAC) of *IEEE 802.11p*. The MAC abilities are essential in order to reach requirements as high-speed data transmission and self-organization of networks. In the MAC protocol of 802.11p, the collision avoidance mechanism is used. That means in a first step, a node needs to listen to a channel. Then, two cases can be distinguished: The channel is free (for a specific time period), and then data packets can be sent directly. Otherwise, the node has enter the backoff procedure and wait. For modeling probabilistic timed automaton (PTA) is used, since it fully takes the characteristics of the MAC into account due to the non-deterministic existence and the support of continuous time and probabilistic choice. As tool `PRISM` is used. For modeling, the MAC protocol is abstracted into four modules, a destination node, two sending nodes, and a transmission channel, which are sufficient to cover any transfer case in 802.11p. As performance measures, two different types, the probabilistic and the expected reachability, are considered. With the probabilistic reachability, the successful completion of the data transmission process in 802.11p can be verified. Moreover, the probability of reaching the max backoff counter of any station is much less for 802.11p than for the 802.11 standard. Therefore, the data can be transmitted forward under a high speed. Furthermore, by using expected reachability, it is shown that a collision event in 802.11p is less likely than in 802.11. The authors also point out an approximately four times higher average transmission speed for 802.11p compared to the one of 802.11 standard.

Basin et al. provide in [49] an extensive formal analysis of the Authenticated Key Exchange protocol used in *5G* (5G AKA). This protocol and especially its security guarantees are important for ensuring the security of the users' calls, text messages, and mobile data. The contribution of the authors is very broad. First, the authors formalize the standard, targeting a wide range of properties – *confidentiality, authentication*, and *privacy* – and fine-grained versions of them. Second, the authors create a formal model, which is then evaluated by using the `Tamarin` tool. The formal, systematic security evaluation shows that some critical requirements are underspecified (especially for authentication) or even missing. It is pointed out that without further assumptions, some properties are violated, as the agreement properties on the session keys. Furthermore, the authors criticize the standard's choice of implicit authentication as well as the absence of key confirmation. The authors explicitly state that this introduces weaknesses if the protocol is not used in the way it is intended for. Moreover, the authors detect a likely realistic privacy attack, due to the fact that 5G AKA does not provide unlinkability against an active attacker. Additionally, the authors suggest a fix for the security issue.

Cremers and Dehnel-Wild also study in [50] *5G* AKA, performing a fine-grained formal analysis with the `Tamarin` tool. The authors state several challenges which complicated their work: first, the complexity of the specification documentation; second, the complexity of the protocol involving all four parties which are defined in the protocol specification and third, the informal nature of the security requirements, forcing the modeler to make complex assumptions on the basis of possible use cases. All four parties are modeled by the authors. Furthermore, possible assumptions on the channels connecting these four parties have been modeled precisely. The proposed formal model from 5G AKA standard enables a detailed view of the interactions between several security-critical components. The results show that 5G AKAs security is based on unstated assumptions on the inner workings of underlying channels. This results in an attack which exploits a potential race condition. However, even for the honest case, solving the race condition does not necessarily prevent the attack. It is stated that in practice, the standard can be implemented "correctly" in an insecure manner. Moreover, the authors propose a possible fix based on their findings.

Zhang et al. focus in [51] on the *5G*-EAP-TLS protocol, which is defined in 5G networks for subscriber authentication in limited use cases as private networks or IoT environments. One main security goal is to ensure mutual authentication between subscribers and their home network. The authors state to provide the first 5G-EAP-TLS formal protocol model and perform a security analysis with the use of the Scyther model checker. In their model two roles are considered, user equipment (UE) and network (NW). The last is a composition of the home network and the server network. The authors check four security related properties: the *secrecy* of the Subscription Permanent Identifier (SUPI) and the one of the session key; for UE, the secrecy hold for SUPI and the session key; and for NW, only the secrecy of the SUPI can be verified. The other two security-related properties, the *non-injective synchronization* of events and the *non-injective agreement* on data, are falsified with Scyther, for both UE and NW.

Zhang et al. focus in [52] on the *5G*-EAP-TLS protocol. There, the protocol is modeled in applied pi calculus, while ProVerif is used for the security analysis. The authors extend their previous work in [51] by using a more expressive formal language which is capable of modeling the protocol's behavior more precisely. Moreover, a more fine-grained formal model is provided, i.e., the severing network and the home network are considered to be a single entity. The authors check three secrecy and two authentication statements. For the authentication, it is falsified with ProVerif that the subscriber and the home network agree after successful termination on the identification of each other. Furthermore, ProVerif falsifies that after successful termination, both parties agree on the pre-master key. The secrecy statements can be successfully verified with ProVerif. Those statements include: The adversary must not be able to obtain the SUPI of an honest subscriber, nor the pre-master key, nor the session key. The authors propose a provable fix, showing that their revised version fulfills the stated security properties.

Koutsos considers in [43] the privacy of *5G*-AKA. Although asymmetric randomized encryption is used in order to reach a better privacy than for 3G or 4G, only the IMSI-catcher attacker can prevented. Other known privacy attacks as the Failure Message Attack and Encrypted IMSI Replay Attack still hold. In a second step, the 5G-AKA protocol is modified for the prevention of those attacks. The security proof is performed by Bana-Comon indistinguishability logic and shows the absence of those privacy attacks.

Braeken et al. propose in [44] based on the detected security issues in *5G*-AKA in [49] a new version. There a non-monotonic logic – also known as RUBIN – is used to successfully verify the proposed scheme. The reason for choosing RUBIN was that this method is quite close to the actual protocol's implementation.

Sharma et al. proposes in [45] an enhanced handover AKA protocol for being used in *5G* communication networks in order to overcome security vulnerabilities as false base-station attack, key compromise, DoS attacks, and high authentication complexity. The authors use AVISPA to show that their proposed protocol is not vulnerable to the stated attacks.

Han et al. in [46] suggest the employment of Mobile Edge Computing (MEC) servers into the traditional authentication architecture for re-authentication. Furthermore, instead of using one-way hash functions and permanent names for authentication, the use of existing Extensible Authentication Protocol-Authentication and Key Agreement (EAP-AKA) protocol pseudonyms is proposed. In their security analysis, the authors use AVISPA and especially consider *mutual authentication, confidentiality*, and *anonymity*. All those security attributes can be verified.

Cao et al. propose in [47] a group-based handover authentication and re-authentication protocol for massive machine type communication (mMTC) in *5G* wireless networks. By using BAN logic and the model checkers `AVISPA` and `SPAN`, the authors verify that their proposed protocol is secure against various malicious attacks.

5 Conclusions and Key Points

Connected cars services, which increase road safety, and contribute to traffic flows' efficiency and passengers' comfort, require complex communication infrastructure behind. V2X, the most important technology in connected cars communication, includes two main subnetworks – *intra*-vehicle network, including a collection of in-vehicle controlling and processing units and sensors, and *inter*-vehicle network, including the communication between the vehicle and surrounding.

Intra-vehicle communication usually involves bus protocols and media-oriented protocols. The most common bus protocols are LIN, CAN, and FlexRay. Widely used media protocols are MOST and LVSD. Nowadays, Automotive Ethernet, due to increased bandwidth and cheap components, is taking over both purposes.

Inter-vehicle communication includes two types of protocols, categorized by the used technology. The first type is a WiFi-based protocol, often referred to as IEEE802.11p from the name of the first standard designed to this scope. The IEEE802.11p protocol is now superseded and became part of WiFi protocol IEEE802.11. Two other initiatives based on IEEE802.11p are ETSI ITS-G5 in Europe and IEEE 1609 in the USA. In parallel to WiFi-based protocol, cellular technologies also offer solutions for V2X communication. Cellular solutions are known as C-V2X (Cellular-V2X) and include LTE-V2X and recently 5G, with additional V2X functionalities announced for future releases, including platooning, extended sensors, automated driving, and remote driving.

Because of significant growth and advancements in V2X technology, security issues related to them are on the rise. The ***security-by-design*** frameworks, including threat modeling and formal methods, have the potential and means to answer these challenges.

The threat modeling section discussed state-of-the-art methodologies and the ability to adapt those for the automotive industry. It was shown that various methodologies already exist for plenty of scenarios by either using more general approaches or even adapting those general approaches for more specific settings. For the latter, two explicit, for the automotive domain adapted, methodologies were discussed. Upcoming challenges will therefore not only include enhancing those methodology in the research but more likely to consider this research into the development process of the automotive domain.

Another security-by-design framework – formal verification and its applications in automotive industry were also discussed in this chapter. It was shown that formal verification approaches are clearly different depending on the type of the protocol. While for *intra*-vehicle protocols the focus of the approaches are mainly *functional correctness* and the *investigation of performance*, for *inter*-vehicle protocols – especially for 5G – the focus is clearly *security properties* as secrecy and authentication. However, none of the approaches focus on implementations of the corresponding protocols. So far applying formal verification tools to verify those implementations, with different purposes as checking for implementation errors, but also to check if the implementation follows the standard/specification and does not pose additional security issues, is still an open issue. As stated in [53] and the references therein, for several implementations for widely used implementations of different application layer protocols, several security issues have been detected, opened by the implementation since they do not follow the corresponding standard.

Further research might consider – as stated in [2] – forced protocol downgrading, which might arise due to the unavailability of the technology. Further research also might deal with [32] a combination of tools for better overall results in case of restrictions of the model checker, model checking for different versions of a protocol, and an in-depth analysis in order to provide a very broad verification for the connected vehicle by a suitable combination of different in-depth verifications of pieces in the protocol and some an overall analysis.

Acknowledgments This work was funded by the Austrian Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK).

References

1. A. Bazzi, G. Cecchini, M. Menarini, B.M. Masini, A. Zanella, Survey and perspectives of vehicular wi-fi versus sidelink cellular-V2X in the 5G era. Future Internet **11**(6), 122 (2019)
2. B. Stojanović, K. Hofer-Schmitz, Formal methods for connected vehicle protocols, in *2019 27th Telecommunications Forum (TELFOR)* (IEEE, 2019), pp. 1–4
3. E. Hamida, H. Noura, W. Znaidi, Security of cooperative intelligent transport systems: standards, threats analysis and cryptographic countermeasures. Electronics **4**(3), 380–423 (2015)
4. A. Alnasser, H. Sun, J. Jiang, Cyber security challenges and solutions for V2X communications: a survey. Comput. Netw. **151**, 380–423 (2019)
5. P. Sewalkar, J. Seitz, Vehicle-to-pedestrian communication for vulnerable road users: survey, design considerations, and challenges. Sensors **19**(2), 358 (2019)
6. S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, L. Kilmartin, Intra-vehicle networks: a review. IEEE Trans. Intell. Transp. Syst. **16**(2), 534–545 (2014)
7. C. Specification, Version 2.0, Robert Bosch GmbH
8. R.I. Davis, A. Burns, R.J. Bril, J.J. Lukkien, Controller Area Network (CAN) schedulability analysis: refuted, revisited and revised. Real-Time Syst. **35**(3), 239–272 (2007)
9. A.X.A. Sim, B. Sitohang, OBD-II standard car engine diagnostic software development, in *2014 International Conference on Data and Software Engineering (ICODSE)* (IEEE, 2014), pp. 1–5
10. I.A. Grzemba, *MOST: The Automotive Multimedia Network* (Franzis Verlag, Munchen, 2012)
11. S.B. Huq, J. Goldie, An overview of LVDS technology. Natl. Semicond. Appl. Note **971**, 1–6 (1998)

12. M. Ruff, Evolution of local interconnect network (LIN) solutions, in *2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No. 03CH37484)*, Vol. 5 (IEEE, 2003), pp. 3382–3389
13. P.-S. Murvay, B. Groza, Practical security exploits of the FlexRay in-vehicle communication protocol, in *International Conference on Risks and Security of Internet and Systems* (Springer, 2018), pp. 172–187
14. C. Corbett, E. Schoch, F. Kargl, F. Preussner, Automotive ethernet: security opportunity or challenge? Sicherheit 2016-Sicherheit, Schutz und Zuverlässigkeit
15. A. Masmoudi, K. Mnif, F. Zarai, A survey on radio resource allocation for V2X communication. Wirel. Commun. Mob. Comput. **2019**, 1–12 (2019)
16. F. Arena, G. Pau, An overview of vehicular communications. Future Internet **11**(2), 27 (2019)
17. ThreatModeler, Application Threat Modeling vs Operational Threat Modeling (2016). https://threatmodeler.com/operational-application-threat-modeling/
18. A. Shostack, *Threat Modeling* (Wiley, Indianapolis, 2014)
19. B. Strom, ATT&CK 101 (2018). https://www.mitre.org/capabilities/cybersecurity/overview/cybersecurity-blog/attck-1011
20. B.E. Strom, A. Applebaum, D.P. Miller, K.C. Nickels, A.G. Pennington, C.B. Thomas, *MITRE ATT&CK: Design and Philosophy* (MITRE, 2018)
21. MITRE, Cloud Matrix (2019). https://attack.mitre.org/matrices/enterprise/cloud/
22. B. Schneier, Attack trees. Dr. Dobb's J. **24**(12), 21–29 (1999)
23. M. Howard, D. LeBlanc, *Writing Secure Code* (Microsoft Press, Redmond, 2014)
24. S. Marksteiner, H. Vallant, K. Nahrgang, Cyber security requirements engineering for low-voltage distribution smart grid architectures using threat modeling. J. Inf. Secur. Appl. **49**, 102389 (2019)
25. R. Ankele, S. Marksteiner, K. Nahrgang, H. Vallant, Requirements and recommendations for IoT/IIoT models to automate security assurance through threat modelling, security analysis and penetration testing, in *Proceedings of the 14th International Conference on Availability, Reliability and Security* (2019), pp. 1–8
26. C. Corradini, The Automotive Threat Modeling Template (2016). https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2016/july/the-automotive-threat-modeling-template/
27. NCCGroup, The Automotive Threat Modeling Template (2017). https://github.com/nccgroup/The_Automotive_Threat_Modeling_Template
28. Intel, Prioritizing Information Risk Security with Threat Agent Risk Assessment, Technical report, Intel (2009)
29. A. Karahasanovic, K. Pierre, M. Almgren, Adapting threat modeling methods for the automotive industry, in *15th ESCAR Conference*
30. D. Basin, C. Cremers, C. Meadows, Model checking security protocols, in *Handbook of Model Checking* (Springer, Cham, 2018), pp. 727–762
31. K. Keerthi, I. Roy, A. Hazra, C. Rebeiro, Formal verification for security in IoT devices, in *Security and Fault Tolerance in Internet of Things* (Springer, 2019), pp. 179–200
32. K. Hofer-Schmitz, B. Stojanović, Towards formal verification of IoT protocols: a review. Comput. Netw. **174**, 107233 (2020). `doi:https://doi.org/10.1016/j.comnet.2020.107233`. http://www.sciencedirect.com/science/article/pii/S1389128619317116
33. L. Viganò, Automated security protocol analysis with the AVISPA tool. Electron. Notes Theor. Comput. Sci. **155**, 61–86 (2006), in *Proceedings of the 21st Annual Conference on Mathematical Foundations of Programming Semantics (MFPS XXI)*. `doi:https://doi.org/10.1016/j.entcs.2005.11.052`. http://www.sciencedirect.com/science/article/pii/S1571066106001897
34. S. Gürgens, N. Lahr, D. Zelle, On formal security analysis of automotive systems, in *15th Embedded Security in Cars (escar)* (2017). http://sit.sit.fraunhofer.de/smv/publications/download/guergensESCAReu2017.pdf

35. C. Pan, J. Guo, L. Zhu, J. Shi, H. Zhu, X. Zhou, Modeling and verification of CAN bus with application layer using UPPAAL. Electr. Notes Theor. Comput. Sci. **309**, 31–49 (2014)
36. D. Thiele, J. Schlatow, P. Axer, R. Ernst, Formal timing analysis of CAN-to-Ethernet gateway strategies in automotive networks. Real-Time Syst **52**(1), 88–112 (2016)
37. B. Zhang, On the formal verification of the FlexRay communication protocol, in S. Merz, T. Nipkow (eds.), *Automatic Verification of Critical Systems, Automatic Verification of Critical Systems (AVoCS 2006)* (Nancy, 2006), pp. 184–189. https://hal.inria.fr/inria-00091667
38. S. Gordon, S. Choosang, Verification of the FlexRay transport protocol for AUTOSAR in-vehicle communications. Int. J. Veh. Technol. **2010**, 1–23 (2010)
39. A. Bruni, M. Sojka, F. Nielson, H.R. Nielson, Formal security analysis of the MaCAN protocol, in *International Conference on Integrated Formal Methods* (Springer, 2014), pp. 241–255
40. D. Dolev, A. Yao, On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983). doi:10.1109/TIT.1983.1056650
41. G. Rodriguez-Navas, J. Proenza, H. Hansson, An UPPAAL model for formal verification of master/slave clock synchronization over the controller area network, in *Proceedings of the 6th IEEE International Workshop on Factory Communication Systems*, Torino (IEEE Computer Society Press, Los Alamitos, 2006)
42. C. Tofts, Compositional performance analysis, in E. Brinksma (ed.), *Tools and Algorithms for the Construction and Analysis of Systems* (Springer, Berlin/Heidelberg, 1997), pp. 290–305
43. A. Koutsos, The 5G-AKA authentication protocol privacy, in *2019 IEEE European Symposium on Security and Privacy (EuroS P)* (2019), pp. 464–479. doi:10.1109/EuroSP.2019.00041
44. A. Braeken, M. Liyanage, P. Kumar, J. Murphy, Novel 5G authentication protocol to improve the resistance against active attacks and malicious serving networks. IEEE Access **7**, 64040–64052 (2019). doi:10.1109/ACCESS.2019.2914941
45. A. Sharma, I. Sharma, A. Jain, A construction of security enhanced and efficient handover AKA protocol in 5G communication network, in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (2019), pp. 1–6. doi:10.1109/ICCCNT45670.2019.8944569
46. K. Han, M. Ma, X. Li, Z. Feng, J. Hao, An efficient handover authentication mechanism for 5G wireless network, in *2019 IEEE Wireless Communications and Networking Conference (WCNC)* (2019), pp. 1–8. doi:10.1109/WCNC.2019.8885915.
47. J. Cao, M. Ma, H. Li, Y. Fu, X. Liu, EGHR: efficient group-based handover authentication protocols for mMTC in 5G wireless networks. J. Netw. Comput. Appl. **102**, 1–16 (2018). doi:https://doi.org/10.1016/j.jnca.2017.11.009. http://www.sciencedirect.com/science/article/pii/S1084804517303776
48. C. Zhou, Y. Wang, M. Cao, J. Shi, Y. Liu, Formal analysis of MAC in IEEE 802.11 p with probabilistic model checking, in *2015 International Symposium on Theoretical Aspects of Software Engineering* (IEEE, 2015), pp. 55–62
49. D. Basin, J. Dreier, L. Hirschi, S. Radomirovic, R. Sasse, V. Stettler, A formal analysis of 5G authentication, in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (ACM, 2018), pp. 1383–1396
50. C. Cremers, M. Dehnel-Wild, Component-based formal analysis of 5G-AKA: channel assumptions and session confusion, in *Network and Distributed Systems Security (NDSS) Symposium 2019* (Internet Society, San Diego, 2019)
51. J. Zhang, Q. Wang, L. Yang, T. Feng, Formal verification of 5G-EAP-TLS authentication protocol, in *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)* (2019), pp. 503–509. doi:10.1109/DSC.2019.00082
52. J. Zhang, L. Yang, W. Cao, Q. Wang, Formal analysis of 5G EAP-TLS authentication protocol using proverif. IEEE Access **8**, 23674–23688 (2020). doi:10.1109/ACCESS.2020.2969474
53. K. Hofer-Schmitz, B. Stojanović, Towards formal methods of IoT application layer protocols, in *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)* (IEEE, 2019), pp. 1–6