Security and Privacy for Innovative Automotive Applications: A Survey

Van Huynh Le[a], Jerry den Hartog[a], Nicola Zannone[a]

[a]*Eindhoven University of Technology, Postbus 513, 5600 MB Eindhoven, the Netherlands*

Abstract

Software applications play an important role in vehicle innovation, aiming at improved safety, efficiency, and comfort, and creating the new areas of cooperative intelligent transport systems and autonomous vehicles. To accommodate modern applications, vehicles have become increasingly computerized and connected. Despite the benefits that the adoption of these applications bring to the automotive sector and consumers, automotive applications along with the enabling technological innovation open great challenges to security and privacy. Addressing these challenges is a prerequisite for the acceptance and deployment of innovative applications at a large scale. This survey investigates the main security and privacy challenges for the design of automotive applications and platforms. Based on the main security and privacy requirements and threats identified, we review existing protection mechanisms proposed within the automotive domain. We then identify the main gaps in existing research and draw a roadmap for future research.

Keywords: automotive, application, platform, security, privacy, literature study

1. Introduction

Modern vehicles have become complex systems containing a large number of interconnected embedded computers, sensors and actuators as well as interfaces to communicate with the outside world. These components are often coupled with an increasing use of software applications that aim to improve safety, efficiency and comfort. This trend is still ongoing; future vehicles will become even more complex and connected to accommodate innovative applications and enable new ways of using vehicles. Despite their benefits, these innovations open new security and privacy challenges. This survey investigates the current state of the art in security and privacy for automotive applications and platforms.

Existing automotive applications cover three main categories: *control systems* that manage physical functions of the vehicle including engine, chassis, body, and passive safety functions; *telematics systems* that provide information and support entertainment as well as financial transactions; and complex *advanced driver assistance systems (ADAS)* that aim to turn vehicles into intelligent systems, improve safety, and enhance driving experience. The complexity that accompanies all these different applications, as well as the need to quickly adopt new applications, is challenging the traditional approach of rigidly building applications into the vehicles during assembly. This approach of adding embedded computers for each new application has reached its cost and complexity limits [1]. Therefore, a current trend is the adoption of application platforms to increase flexibility while reducing cost and complexity.

Application platforms offer easier development and deployment of new services. Flexible software distribution allows applications to be selected according to customers' requirements [2]. By providing common application programming interfaces (APIs) that abstract the underlying hardware, application platforms can also increase software reuse [3] compared to low-level and hardware-specific approaches that are currently largely adopted by the automotive industry. A platform can also reduce hardware complexity and cost by hosting multiple applications on the same hardware [1].

Until recently, most applications and platforms were implemented inside vehicles for control systems with little communication with the outside world. Yet, a vehicle is no longer isolated within intelligent transport systems; information sharing is essential for many advanced applications. Vehicles need to communicate with other entities, such as personal devices, other vehicles, road-side infrastructure, and the Internet. Modern vehicles thus involve three main areas as illustrated in Fig. 1. *In-vehicle systems* were initially designed for applications in control systems but have been extended to support ADAS and telematics. *Vehicular Ad Hoc Networks (VANETs)* are ad-hoc communication networks among vehicles and between vehicles and road side infrastructures enabling collaborative ADAS as well as telematics applications. Finally, *Internet-based applications* have telematics as main purpose.

All these trends of more complex systems, flexible application platforms and increased connectivity also come with significant security and privacy challenges that must be addressed before smart vehicles and innovative applications can be widely adopted and large scale intelligent transport systems can be successfully deployed. In particular, the increasing number of vehicle assets (e.g., more vehicular communication, personal information stored in vehicles, and audio-visual media) along with the ease to interact with modern vehicles has attracted considerable attention to the study of their security, and several attacks have already been demonstrated [4–13]. The potential impact of attacks can range from slight inconvenience to serious safety, financial and/or privacy consequences. For instance, Miller and Valasek [6] demonstrated an attack in which they could remotely

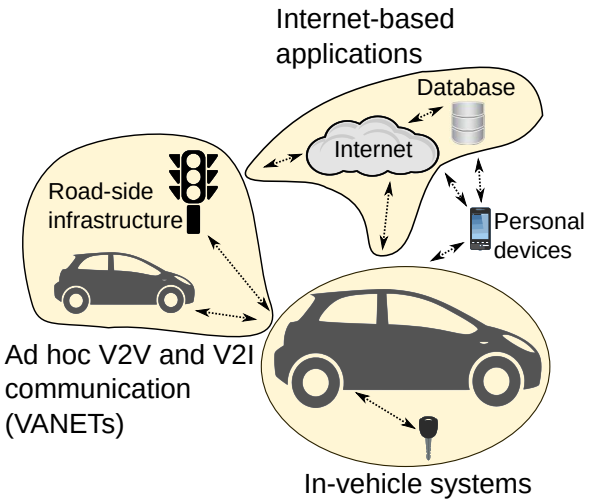

Fig. 1: The anatomy of a connected vehicle

track a vehicle and perform various actions, such as changing radio volume, killing the vehicle's engine, and disabling the brakes.

Automotive security and privacy is a wide topic encompassing multiple subjects, such as automotive system architectures, the plethora of applications and platforms with different requirements, attackers with various motivations and levels of skills, and the diversity of threats and their countermeasures. To identify the most important issues and identify research directions that will enable the acceptance and adoption of innovation in the automotive sector, a broad overview of the field is needed. This survey aims to provide such an overview by presenting a literature review on automotive security and privacy. In particular:

- We review automotive system architectures, applications, and application platforms to provide the context for security and privacy analysis. Applications and platforms from all major areas (in-vehicle systems, VANETs, and Internet-based applications) are discussed, as security and privacy issues can arise not only from individual areas but also from their interconnection.
- We study security and privacy requirements, attackers, threats, and countermeasures. These will form a baseline for the identification of research gaps. Our analysis focuses on VANETs and in-vehicle systems, and does not cover the Internet side of applications. The reason is that the Internet is too broad and heterogeneous with many technologies (databases, web servers, cloud computing, communication protocols, personal devices, etc.). Many of these technologies have open problems in themselves and may require different research methodologies.
- We perform a gap analysis based on our review of applications, platforms, and threats, and provide directions for future research, such as vehicle data collection, firmware update, automotive Ethernet, and open application platforms.

Related work. The automotive sector is widely studied. Several surveys have summarized the main technological advances in automotive applications, platforms, and security and privacy

Table 1: Surveys of automotive security and privacy

	Simonot-Lion & Trinquet [14]	Coppola & Morisio [15]	Kleberger et al. [16]	Parno & Perrig [17]	Raya & Hubaux [18]	Gillani et al. [19]	Razzaque et al. [20]	Isaac et al. [21]	Othmane et al. [22]	This survey
Applications										
Control systems	x									x
ADAS		x			x		x		x	x
Telematics	x	x			x		x		x	x
Platforms										
Control systems	x									x
ADAS										x
Telematics	x	x								x
Security and privacy—In-vehicle systems										
Requirements									x	x
Attackers			x						x	x
Vulnerabilities		x	x							x
Attacks		x	x	x					x	x
Countermeasures			x						x	x
Security and privacy—VANETs										
Requirements				x	x	x	x	x	x	x
Attackers				x	x		x		x	x
Vulnerabilities				x	x	x	x	x		x
Attacks				x	x	x	x	x	x	x
Countermeasures				x	x	x	x	x	x	x
Security and privacy—Internet-based applications										
Requirements		x							x	
Attackers									x	
Vulnerabilities										
Attacks									x	
Countermeasures									x	

issues. For example, Simonot-Lion and Trinquet [14] provide an overview of applications and platforms for automotive control systems and telematics. Coppola and Morisio [15] review services provided by connected vehicles, communication technologies that enable these services, and popular application platforms for infotainment. They also provide an overview of security and privacy issues concerning connected vehicles. Kleberger et al. [16] review security and privacy issues of in-vehicle systems in connected vehicles. Security and privacy issues in VANETs have also been largely studied. Existing surveys on this area [17–21] cover several aspects ranging from requirements, vulnerabilities, attackers and attacks, to security and privacy measures. Othmane et al. [22] perform an analysis for all three areas, but only focus on a limited range of automotive applications (i.e., ADAS and telematics) and do not analyze automotive platforms.

Together, previous works have covered a wide range of the

security and privacy aspects, sometimes in great detail. However, most of the previous surveys focus on a few aspects at a time. Aiming at a broader view of the field, we provide an overview of all major automotive security and privacy aspects. This view helps in identifying important issues and research gaps. Table 1 compares the scope of existing surveys and this survey.

Organization. The remaining of the paper is structured as follows. The next section provides an overview of vehicular IT architectures and external interfaces in modern vehicles. Sections 3 and 4 identify important technical characteristics of automotive applications and application platforms, providing the context for an analysis of security and privacy threats (Section 5) and existing security and privacy mechanisms (Section 6). Based on this analysis, Section 7 discusses open questions and draws research directions. Finally, Section 8 concludes the paper.

2. System Overview

In this section, we present an overview of the in-vehicle architecture and vehicles' external interfaces.

2.1. In-vehicle Architecture

Several in-vehicle architectures have been proposed [23–28]. Fig. 2, 3, and 4 show three possible modern in-vehicle network architectures. Regardless of the specific network topology, an in-vehicle network consists of interconnected *Electronic Control Units (ECUs)* coupled with sensors and actuators. ECUs are embedded computers that monitor automotive systems via sensors, control the vehicle via actuators, or report vehicle data.

ECUs, sensors, and actuators are usually grouped into functional domains [14, 23, 27]. Typical functional domains are *drivetrain*, *chassis*, *interior*, and *telematics* [23]. While covering the essentials of in-vehicle architectures, this classification is neither universal nor complete. First, vendors can define additional domains or use different groupings. For example, some authors define *powertrain* instead of *drivetrain* [14, 27][1], *body* instead of *interior* [14, 27][2], *human-machine interface (HMI)*, *multimedia*, and *telematics* instead of *telematics* [14]. Some additional domains that have been proposed are *active safety* [14], *passive safety* [14, 27], and *diagnostics* [14][3]. Secondly, the classification of a given component into domains is not always clear-cut [14]. For example, an airbag can be considered either as a passive safety component or as a body component. Furthermore, some modern cars also have *ADAS* enabled through powerful ECUs and associated sensors, such as cameras and radars. These components do not belong to any of the aforementioned functional domains.

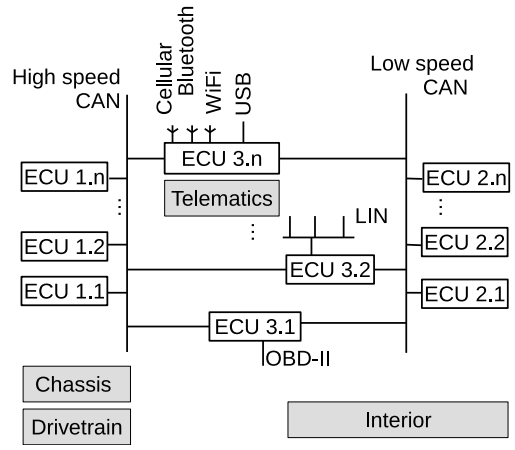

Fig. 2: An in-vehicle network architecture without gateways. This figure is based on the description of Jeep Cherokee 2014 [24].

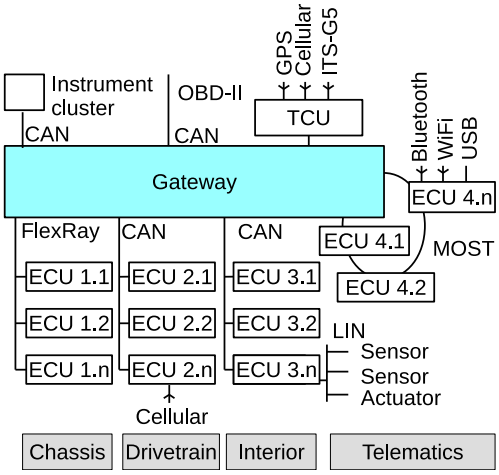

Fig. 3: A network architecture with a central gateway

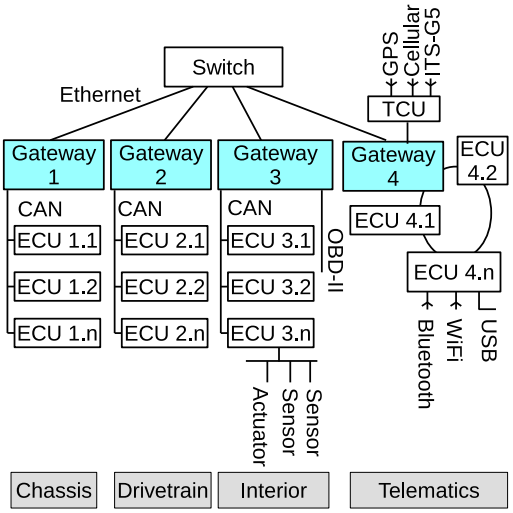

Fig. 4: A network architecture with distributed gateways

ECUs are connected by system buses to perform distributed, inter-ECU functions. Bus technologies are selected based on

[1] Some authors distinguish *drivetrain* and *powertrain* as follows: the drivetrain includes the components that deliver power to the wheels; the powertrain includes the drivetrain plus the engine. In this report, we do not make this distinction but rather consider the engine as a part of the drivetrain.

[2] The *body* domain includes components that are not strictly *interior*, for example wipers. We do not make this distinction between body and interior.

[3] In this report, *active safety* means avoiding or minimizing the effects of an accident, before a crash. *Passive safety* is understood as reducing the effects of an accident that has already happened. The terms are used with the same meanings as presented in [14].

Table 2: Bus systems in vehicles

Class	Transfer rate	Applications	Representative
A	≤ 10 kbps	Actuator and sensor networking	LIN
B	≤ 125 kbps	Controlling components in the body and comfort domain	Low speed CAN
C	≤ 1 Mbps	Real-time applications in the powertrain and chassis domains	High speed CAN
C+	≤ 10 Mbps	Real-time applications in the powertrain and chassis domains	FlexRay
D	> 10 Mbps	Telematics and media data transfer	MOST

Table 3: Types of vehicle communication. VANET refers to both ad hoc Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) networks. The collective term for all vehicle communication is Vehicle-to-Everything (V2X).

Terminology	Definition	Range
Vehicle-to-Device (V2D)	Communication with a device inside the vehicle	Physical or short-range wireless
Vehicle-to-Vehicle (V2V)	Communication among vehicles	Short-range wireless
Vehicle-to-Infrastructure (V2I)	Communication between vehicles and roadside infrastructure	Short-range wireless
Vehicle-to-Network (V2N)	Communication between vehicles and IT or cellular infrastructure	Long-range wireless

cost and technical constraints. The main technical constraints are data transfer rate, interference immunity, real-time capability, and number of network nodes [23, 29]. Table 2 provides a classification of bus systems based on their transfer rate and applications [23, 27–29][4].

The most representative network technologies are Local Interconnect Network (LIN), Controller Area Network (CAN), FlexRay, and Media Oriented Systems Transport (MOST). LIN is a low-cost, low-bandwidth network. Its main usage is to connect ECUs with sensors and actuators. CAN is a reliable, medium- to high-bandwidth network with sufficient real-time capability for vehicles. Its main usage lies in the body domain (low speed CAN) and in the drivetrain and chassis domains (high speed CAN). FlexRay is a reliable, high-bandwidth, real-time network. It is mainly used for the drivetrain and chassis domains. MOST is a high speed network intended for multimedia applications. Further technical details about these network technologies can be found in [23, 27–29].

Apart from these technologies, Ethernet[5] is emerging for in-vehicle communications [28, 31–33], and it has been deployed for diagnostics and backbone networks [34]. Fig. 4 shows a possible network architecture in which Ethernet serves as the backbone (further discussed below). Ethernet has also been proposed to connect components in individual sub-networks [28, 31, 32]. However, the use of standard Ethernet technology usually do not guarantee maximum message delay, thus they are unsuitable for safety-critical automotive applications [28]. This drawback has spurred the definition of Ethernet solutions that minimize message delay. The most commonly discussed Ethernet candidates for the automotive domain are IEEE 802.1Q, Audio/Video Bridging (AVB) Ethernet, and TTEthernet [28].

Some ECUs need to communicate with other ECUs from multiple domains. For example, a car's Central Locking Systems not only interacts with non-safety-critical systems (physical door locks, wireless key fobs, remote commands to open the doors) but also with passive safety systems: if the car crashes and airbags are deployed, the system must automatically unlock the doors [4]. One solution is to directly attach these ECUs to multiple sub-networks. In this case, the ECUs are responsible for choosing the relevant sub-networks to read messages from or write messages to. Fig. 2 illustrates an in-vehicle architecture in which some ECUs are connected to two CAN buses, a high-speed bus for drivetrain and chassis and a low-speed one for the interior domain.

Using *gateways* is another method for cross-domain communication [23, 27, 35]. Gateways exchange data between bus lines by reading and converting data from one format to another [23, 35]. A car may employ one central gateway or multiple distributed gateways. A central gateway connects all bus lines, while a distributed one connects two or more buses.

Fig. 3 and Fig. 4 show two examples of vehicle networks employing gateways. Fig. 3 features a network with a central gateway, which is similar to the architectures of BMW 7 Series and VW Passat [25]. Fig. 4 presents a network architecture with distributed gateways, similar to the distributed architectures presented in [23, 26]. This figure also features a point-to-point, switched Ethernet backbone. The switch bridges different domains without expensive message conversions [31, 32]. Note that in both figures, the bus technologies and the external interfaces (WiFi, Bluetooth, etc.) are for demonstration purposes only. For example, an actual vehicle may have FlexRay and MOST replaced by CAN, or have a different arrangement of external interfaces.

2.2. External Interfaces

A car can communicate with various devices including keys, sensors, mobile phones, mp3 players, other vehicles, roadside infrastructures, and IT infrastructures. Table 3 presents the common terminology used to denote vehicle communication.

The communication between a car and the outside world is performed via external interfaces. Usually, external interfaces are attached to a head unit or a telematics control unit (TCU). A head unit typically consists of a screen, navigation applications, telephone applications, and interfaces to connect to consumer electronic devices [36]. A TCU is a dedicated ECU providing wireless connectivity [5, 37]. It is also called communication and control unit [36] or simply communication unit [38]. In the case of V2V and V2I communication, the device is also called on-board unit (OBU) in contrast to roadside units (RSUs) that are located along the roads.

[4]Note that different classification variants have been proposed. For example, in [29], class D includes all networks with bandwidth over 1 Mbps, and there is no class C+. Also, the transfer rate is not an exact limit: the maximum bandwidth of LIN is 20 kbps, but it is still considered a class A network.

[5]Note that Ethernet is not a single technology but a family of physical layer and link layer specifications, which are defined in IEEE 802.3 standards [30].

External interfaces can be classified into three main groups based on access range [39]: physical access, short range wireless access, and long-range wireless access. In the remainder of the section, we provide an overview of these interfaces.

2.2.1. Physical access

The most common interface that requires physical access is for diagnosis. *On-board Diagnostics (OBD)* regulations (for example, OBD-II in the United States, EOBD in the European Union, or JOBD in Japan) require monitoring and detection of emission system deterioration, malfunction, and electrical faults. The required functions are typically provided through an OBD-II port[6]. Specifically, the diagnostic capabilities provided through this port include [40]:

- Requesting stored data about faults (e.g., electrical short-circuit).
- Requesting real-time data from sensors and ECUs. The data includes compulsory emission-related information and additional data defined by the manufacturer.
- Reading and setting fault codes.
- Controlling actuators.

The OBD-II port also allows access to vehicle data in non-diagnostic modes, including normal messages and Direct Memory Read (DMR) [41]. Normal messages are broadcast during normal vehicle operation without the need for requests. With DMR, developers can request a controller's memory contents. Normal messages and DMR are manufacturer-specific.

In addition, vehicles are often equipped with other external interfaces for a variety of purposes. A car can have a CD/DVD player for entertainment. A car can also have USB ports for storing and reading files from nomadic devices [42], or for integration of smartphones.

OBD-II and USB interfaces can be extended with after-market adapters. Such adapters can have wireless interfaces to upload vehicle data for purposes beyond diagnostics. For instance, there are OBD-II adapters providing services, such as usage-based insurance, car usage and driving behavior feedback, and emergency call [43–45].

Apart from these interfaces, some trucks and buses made by major European manufacturers have optional firewalled Fleet Management System (FMS) interfaces that provide access to vehicle data. The interfaces are standardized by FMS-Standard and Bus-FMS-Standard [46]. Currently, there is no equivalent interface for other types of vehicles (the usage of OBD-II adapters is an unofficial use of the interface).

2.2.2. Short-range wireless access

Vehicles are often equipped with interfaces for short-range communication for different purposes. One application of these interfaces is to connect sensors and actuators to the in-vehicle network, for example, keyless entry and tire pressure monitoring systems (TPMS). Viable technologies for sensor and actuator networking are *ZigBee*, *radio-frequency identification (RFID)*, and *ultra-wideband* [47]. ZigBee provides secure low-data-rate, low-power networking [48]. RFID refers to technologies that use radio waves for identification purposes. Among RFID technologies, DASH7 is an open-source protocol tailored for low-power sensor and actuator communication [49]. Ultra-wideband is a radio technology operating in a wide frequency band[7]. It supports high-data-rate, short-range communication at low power consumption levels [50].

In the telematics and multimedia domain, *Bluetooth* and *WiFi* are ubiquitous. Bluetooth is a short-range radio frequency technology, operating in the 2.4 GHz radio band [51]. Common applications of Bluetooth in vehicles are hands-free phone call, address book, and wireless headphone [23]. WiFi is a wireless technology for local networking based on the IEEE 802.11 standards. It can be used for Internet tethering between cars and mobile devices. Both Bluetooth and WiFi can enable car-smartphone integration, as in MirrorLink and Ford's AppLink.

For ad-hoc V2V and V2I communication, the current standards are *Dedicated Short Range Communication (DSRC)* in the United States [52] and *Intelligent Transport Systems operating in the 5 GHz frequency range (ITS-G5)* in Europe [53].[8] At the physical layer, both DSRC and ITS-G5 are based on the IEEE 802.11p standard. DSRC mainly targets safety applications [52], but it is also used for other applications like toll collection, navigation assist, and garage door openers. On the other hand, ITS-G5 is a collection of four channels, each with its own purpose: ITS-G5A (ITS-G5 class A) targets safety applications; ITS-G5B targets non-safety applications; ITS-G5C is for wireless local area networks; and ITS-G5D is reserved for future applications [53].

An alternative to DSRC/ITS-G5 is the next generation of wireless mobile telecommunications technology (5G). Apart from long-range communication (see below), 5G is expected to have advanced broadcast mechanisms and a low-latency Device-to-Device (D2D) communication mode [54, 55]. These features would make 5G suitable for V2V and V2I communication.

2.2.3. Long-range wireless access

Vehicles also have external interfaces for long-range communication, which refers to communication at a distance greater than 1 km. Long-range wireless channels can be classified into *broadcast* and *addressable channels* [39]. Using a *broadcast channel*, a station can broadcast its signals to multiple vehicles without knowing the receivers' address. For example, modern vehicles have global navigation satellite system (e.g., GPS or Gallileo) receivers for positioning information, Traffic Message Channel receivers for traffic and travel information, and satellite radio or digital radio receivers.

In *addressable channels*, messages are sent to devices with specific addresses. These channels are typically used for long-range data and voice transmission. Long-range addressable

[6] We do not consider older diagnostic interfaces, such as OBD-I.

[7] The specific frequency band depends on the country or region. For instance, in the United States, the frequency band for ultra-wideband communication is 3.1–10.6 GHz [50].

[8] In Europe, DSRC refers to the ETSI EN 300-674 standard, which is different from US DSRC. The main application of European DSRC is electronic fee collection [53].

Table 4: Mobile network technologies [58]

Generation	Technology
1G	AMPS, NMT
2G	GSM, D-AMPS
2.5G	GPRS, EDGE, CDMA
3G	CDMA 2000, WCDMA, TD-SCDMA
3.5G	HSPA, EVDO
4G	LTE, WiMAX

channels are provided by 2G/3G/4G cellular networks, or 5G networks in the future.[9] The letter 'G' in 1G/2G/3G/4G/5G stands for generations of mobile telecommunications technology. The first generation (1G) covers analog communication. The second generation refers to digital communication before the IMT-2000 standard specified by the International Telecommunication Union (ITU). The third and fourth generations refer to ITU's IMT-2000 and IMT-Advanced standards. The standards do not define specific technologies but rather lists of service, spectrum, and technical requirements [56]. As an example of service requirements, an IMT-Advanced (4G) technology must support [57]: messaging, voice telephony, high-quality video telephony, video conference, Internet browsing, interactive gaming, file transfer/download, machine-to-machine communication, remote sensor, emergency calling, etc.

Each generation of mobile networks is realized by multiple technologies. Sometimes, the capabilities of an implementation exceed that required by a generation, but do not satisfy the next. This results in informal non-integer generation numbers. Table 4 lists common technologies for each generation [58].

At the time of writing, the technical requirements of 5G are still under development. Nevertheless, many organizations agree that 5G should support a wide range of applications [55, 59, 60]. For example, the Next Generation Mobile Networks Alliance aims at eight 5G application areas [55]: (1) broadband access in dense area, (2) broadband access everywhere, (3) high user mobility, (4) massive Internet-of-Things, (5) extreme real time communication, (6) lifeline communication, (7) ultra-reliable communication, and (8) broadcast-like services.

3. Automotive Applications

A large amount of innovation in the automotive domain is driven by software. In 2012, electronics and software contributed to about 90 percent of automotive innovations [61]. New automotive technologies will enable even more software innovations. First, sensors generate a large amount of data, which can be used, for instance, to improve future vehicles or to provide after-market services. Secondly, more processing power and communication capabilities will make it possible to store, transmit, and analyze vehicle data, and to run a wider range of infotainment applications. Emerging V2I and V2V technologies will also give the means to create applications for safer, more efficient, and more comfortable traveling.

Alongside functionalities, applications also bring complexity. The first source of complexity is the large amount of code: a typical high-end car in 2016 runs more than 100 million lines of software code [62]. The deployment of additional applications will further increase this complexity. The second source of complexity is application interactions. Applications from multiple domains with different technical, security, and privacy requirements may communicate and interfere with each other.

To better understand the heterogeneous applications, we propose a classification of applications based on their functional domains and key technical characteristics. We then discuss current and near-future automotive applications, showing their benefits and complexity.

3.1. Application Classification

There is a large variety in automotive applications. Kroh et al. [63] and Kargl et al. [66] propose a classification of applications for VANETs. We revise and extend their classification to also deal with in-vehicle and Internet-based applications. In particular, we have included the functional domain as this is an important characteristic to understand the needs of automotive applications, while omitting low-level communication characteristics, such as singe-hop vs. multi-hop communication and relevancy-based vs. periodic communication. Our classification of automotive applications is based on five dimensions:

1. *Functional domain.* This dimension indicates the functional domains of the electronics involved in an application. As discussed in Section 2.1, functional domain definition varies across vendors. Here we settle on five common domains and one emerging domain, namely *powertrain*, *chassis*, *body*, *passive safety*, *telematics*, and *ADAS*.
2. *Influence on safety.* This dimension describes the level of potential safety loss if an application is unavailable or misbehaves. If it has strong impact on safety, then the application is *safety-critical*. If it improves safety to some extent but its absence does not cause serious consequences, then the application is *safety-related*. Otherwise, if the application is not intended to improve safety, we say that it is *not safety-related*. The more safety-critical an application is, the more assurance about its proper functioning is required.
3. *Driver involvement.* This dimension characterizes how much attention the driver must pay to the notifications generated by an application. An application can be *vehicle-only*, meaning that it does not display notifications to the driver. Otherwise, applications may display messages that require driver's *awareness, attention*, or even *reaction*. If an application requires a high level of driver involvement, then we need a high level of assurance on the integrity of presented messages.
4. *Connectivity.* This dimension characterizes the type of connectivity required by an application. Some applications only involve ECUs inside the vehicle. Other appli-

[9] In a cellular network, each mobile device is connected to a fixed-location cell tower; each cell tower provides connection coverage for one area, or "cell". Future 5G networks is expected to support device-to-device communication, so they will not be purely cellular.

Table 5: Characteristics of selected applications

	Application	Domain	Influence on safety	Driver involvement	Connectivity	Time constraint
AP1	Engine management system	P	■	□	L	■
AP2	Transmission control	P	■	□	L	■
AP3	Anti-lock brake systems	C	■	□	L	■
AP4	Automatic stability control	C	■	□	L	■
AP5	Control of wipers, doors, windows, seats, mirrors	B	□	■	L	■
AP6	Control of doors and windows via phones [38]	B	□	■	D	■
AP7	Airbag control	S	■	□	L	■
AP8	eCall [38, 63]	S T	■	□	I N	■
AP9	eTolling [38]	T	□	□	I	■
AP10	Point-of-Interest [38, 63]	T	□	■	I	■
AP11	Web-browsing, video streaming, instant messaging	T	□	■	I N	■
AP12	Remote diagnosis [38]	C P T	■	□	I N	□
AP13	Vehicle data upload for usage-based insurance	C P T	□	□	N	□
AP14	Local danger warning [38]	A C T	■	■	V	■
AP15	Intersection collision warning [63]	A T	■	■	V	■
AP16	Traffic signal violation warning [63]	A T	■	■	I	■
AP17	Emergency vehicle signal pre-emption [63]	A T	■	□	I	■
AP18	Pedestrian crossing information [63]	A T	■	■	V	■
AP19	Pedestrian detection [64]	A	■	■	L	■
AP20	Active brake [38]	A C T	■	□	V	■
AP21	Parking assistant [65]	A	■	■	L	■
AP22	Adaptive cruise control [64]	A	■	■	L	■

Legend:

Domain	A: ADAS	B: Body	C: Chassis
	P: Powertrain	S: Passive safety	T: Telematics
Influence on safety	□: not safety-related	■: safety-related	■: safety-critical
Driver involvement	□: vehicle-only	■: awareness	■: attention/reaction
Connectivity	D: Vehicle-to-Device	I: Vehicle-to-Infrastructure	N: Vehicle-to-Network
	V: Vehicle-to-Vehicle	L: In-vehicle only	
Time constraint	□: Non real-time	■: Soft real-time	■: Hard real-time

cations may require *V2D*, *V2V*, *V2I*, or *V2N* connections (Table 3). This dimension implies which communication services a platform should provide to accommodate its target applications.

5. *Time constraint.* This dimension characterizes the level that an application has to adhere to a specified response time. If occasional missing of deadlines does not have serious effects, then the application is *soft real-time* [23]. If an application must always adhere to a specific response time and missing a deadline can lead to serious problems, then it is *hard real-time* [23]. Otherwise, if an application does not have constraints on the response time, then it is *non real-time*. This dimension determines whether the platform hosting the application should provide real-time capabilities. Note that this dimension is different from the "time constraint" dimension in [63, 66], which is based on an application's estimated maximum response time (in seconds).

Below we discuss applications grouped by their domain, while Table 5 shows some representative applications according to the identified dimensions. Sometimes the characteristics depend on the implementation, so we refer to the specific descriptions from the cited sources.

3.2. Review of Existing and Near Future Applications

Vehicles have evolved beyond just a means of moving people and goods. To passengers, cars should be safe, comfortable, and fun. To the community, cars should be environmentally friendly and enable customized services, such as insurance and tolling. Cars should also help manufacturers and mechanics to enhance future cars and simplify mechanics' job. To service providers, cars present opportunities to provide new business models and value-added services. A car can host many applications to cater for different stakeholders.

In the powertrain and chassis domains, vehicles have applications controlling vehicle dynamics [14, 27]. The main applications in the powertrain domain are engine control, and transmission and gear control [27]. The chassis domain is responsible for driving dynamics, driving assistance, and active safety. Examples of applications in this domain include anti-lock braking system, automatic stability control, adaptive cruise control, anti-slip regulation, etc. [27].

Body/comfort domain applications control non-dynamic vehicle components such as climate control systems, dashboard, wipers, lights, doors, windows, seats, mirrors, etc. [14, 27]. These components can be controlled by direct interaction with the car, but also via smartphones [63] or voice control.

When an accident happens, passive safety applications aim to alleviate its consequences. For instance, airbag functioning is regulated by a complex algorithm with input from various sensors [14]. Another application is eCall, which automatically calls an emergency number in case of accidents [38, 63].

While not directly related to driving tasks or safety, multimedia and telematics applications also add great values to vehicles. Common applications are radio, CD/DVD player, and road toll collection. Internet connectivity and more powerful in-vehicle computers/smartphones have started to bring sophisticated applications for media downloading (e.g., web-browsing, video streaming, on-demand navigation, and Point-of-Interest notification) and two-way correspondence (e.g., voice over IP, instant messaging, parking reservation). Internet connectivity also enables transmission of vehicle data for use in a broad range of applications, for example, usage-based insurance [67], traffic management [63, 68], and remote diagnosis [38]. Wireless connectivity also enables local services, e.g. automatic parking management and local electronic commerce [68], or integration of smart-homes [69] and mobile devices [63, 70].

Apart from the more conventional functions, complex advanced driver assistance systems (ADAS) are an emerging area in the automotive domain. ADAS applications process data from various on-board sensors and external systems to assist the driver in controlling the vehicle, preventing accidents and reducing the burden on the driver. Moreover, those applications can inform the driver of the current situation or actively intervene in vehicle dynamics [64, 65, 71]. Some examples of applications involving control of the vehicle are adaptive cruise control, lane-following assistant, and automatic emergency braking [65]. Examples of informing applications include collision warning, awareness about emergency vehicles, road hazard warnings [38, 63, 68, 72, 73], and parking assistant [65]. There are also applications focusing on human factors, including software to detect driver stress, fatigue, and anger [64, 74, 75] or to improve the driver's perception, such as night vision assistant [65, 76]. Today's ADAS are the first step towards autonomous vehicles, which will handle most or even all driving tasks.

4. Application Platforms

In the previous section, we discussed various automotive applications. The current approach of adding embedded computers for new applications, however, is rigid and has reached its cost and complexity limits [1]. Application platforms offer a flexible solution. First, platforms can provide an environment for customers to select applications according to their requirements [2]. Also, platforms provide common APIs that abstract the underlying hardware. This encourages software reuse and avoids recurring development expenses. A platform can reduce cost and complexity by sharing the same hardware among multiple applications.

In this section we propose a classification of application platforms and use it to analyze to what degree existing platforms can fulfill the demands of modern automotive applications.

4.1. Platform Classification

We propose six dimensions to classify application platforms. The dimensions address both technical concerns, e.g., the type of applications a platform can host, and business concerns, e.g., how open a platform is for different stakeholders. They address the development and deployment of the platform (dimensions 1, 4, and 5), the development and deployment of applications on top of the platform (dimensions 2, 3), and the platform's access to in-vehicle and external networks (dimension 6).

1. *Platform location.* This dimension describes the location where a platform is deployed with respect to the vehicle. Some platforms are *in-vehicle*, some run *on smart-phones*. Other platforms are *Internet-based*, i.e. they provide services or collect vehicle data via the Internet.
2. *Platform purpose.* This dimension categorizes the functional domains of applications supported by a platform, including *telematics*, *ADAS*, and *automotive control systems*, which covers functional domains powertrain, chassis, body, and passive safety.
3. *Openness for application distribution.* This dimension describes the stakeholders that can distribute applications for a platform. Some platforms are *for vehicle manufactures and suppliers only*. Other platforms *allow third-party developers to distribute applications*.
4. *Openness for platform development.* This dimension characterizes how open a platform itself is for verification and modification by parties other than the platform developer/distributor. A platform can be *closed*, meaning that only the platform distributor has access to the source code. Other platforms are *open* to others for verification and additionally for modification.
5. *Platform maturity.* This dimension characterizes to what degree a platform has been developed and used in practice. There are platforms that are *commercially used* in real-life vehicles. Some platforms only have *prototypes*. There are also platforms in the *design/concept* phase.
6. *Network access.* This dimension describes a platform's access rights to in-vehicle and external network interfaces. We distinguish two levels of access rights (*active/write* and *passive/read*) for three network types, namely *safety-critical in-vehicle networks* (for chassis, powertrain, and passive safety), *non-safety-critical in-vehicle networks* (for body train and telematics), and *external networks*, such as V2V and V2I. This dimension has implications

Table 6: Automotive application platforms

	Platforms	Platform location	Platform purpose	Application distribution	Platform development	Platform maturity	Network access
PL1	ERIKA Enterprise—OSEK/VDX implementation [77]	In-vehicle	Control systems	Closed	Open	Production	WC
PL2	osCAN—OSEK/VDX implementation [78]	In-vehicle	Control systems	Closed	Closed	Production	WC
PL3	Artic Core—AUTOSAR Classic implementation [79]	In-vehicle	Control systems	Closed	Open	Production	WC
PL4	MICROSAR—AUTOSAR Classic implementation [80]	In-vehicle	Control systems	Closed	Closed	Production	WC
PL5	QNX CAR [81]	In-vehicle	Telematics	Closed	Closed	Production	WE
PL6	Windows Embedded Automotive [82]	In-vehicle	Telematics	Closed	Closed	Production	WE
PL7	GENIVI [83]	In-vehicle	Telematics	Closed	Open	Production	WB,WE
PL8	Automotive Grade Linux [84]	In-vehicle	Telematics	?	Open	Prototype	WB,WE
PL9	Apertis [85]	In-vehicle	Telematics	Open	Open	Prototype	WE
PL10	Automotive Grade Android [86]	In-vehicle	Telematics	?	Open	Prototype	WE
PL11	QNX Platform for ADAS [87]	In-vehicle	Control systems, ADAS	Closed	Closed	Production	WC,WE
PL12	Green Hills Platform for ADAS [88]	In-vehicle	Control systems, Telematics, ADAS	Closed	Closed	Production	WC,WE
PL13	Wind River Helix Drive [89]	In-vehicle	ADAS	Closed	Closed	Production	WC,WE
PL14	OVERSEE [90]	In-vehicle	Control systems, Telematics, ADAS	?	Open	Prototype	WC,WE
PL15	CarPlay [91]	On smartphones	Telematics	Closed	Closed	Production	WE
PL16	Android Auto [92]	On smartphones	Telematics	Open	Closed	Production	WE
PL17	MirrorLink [93]	On smartphones	Telematics	Open	Closed	Production	WE
PL18	AppLink [94]	On smartphones	Telematics	Open	Closed	Production	RC,WE
PL19	Manufacturers' data servers	Internet-based	Telematics	Closed	Closed	Production	RC,WE
PL20	Automatic—based on OBD adapters [44]	Internet-based	Telematics	Open	Closed	Production	RC,WE
PL21	AT&T UBI—based on OBD adapters [45]	Internet-based	Telematics	Closed	Closed	Production	RC,WE
PL22	APPSTACLE [95]	In-vehicle, Internet-based	Telematics	Open	Open	Design	RC,WE

Legend:

Platform location	In-vehicle	On smartphones	Internet-based
Purpose	Control systems	Telematics	ADAS
Application distribution	Closed	Open	?: Info not available
Platform development	Closed	Open	
Platform maturity	Production	Prototype	Design
Network access			
• Access right	R Read	W Write	
• Network:	C Internal, critical	B Internal, non-critical	E External

in the amount of vehicle data that the platform can access and the components that the platform can control.

4.2. Review of Existing Application Platforms

A variety of application platforms have been designed to address the diversity of applications and business models. In this section, we study existing platforms and classify them according to the dimensions identified in the previous section. Table 6 summarizes the classification results with platforms grouped by their deployment location and purposes. Note that in "network access" column, write access infers read access; access to safety-critical internal networks infers access to non-safety-critical networks.

4.2.1. In-vehicle platforms

OSEK/VDX[10] and AUTOSAR[11] are two standards for application platforms for automotive control systems. OSEK/VDX is a standards body formed by eight German and French automotive companies[12]. It specifies a software architecture for automotive control systems, with the goal of software portability and reusability [96]. OSEK/VDX includes specification of

[10] OSEK stands for "Offene Systeme und deren Schnittstellen für die Elektronik in Kraftfahrzeugen" in German, which can be translated as "Open Systems and their Interfaces for the Electronics in Motor Vehicles". VDX stands for "Vehicle Distributed eXecutive".

[11] AUTOSAR is short for "AUTomotive Open System ARchitecture". The AUTOSAR development partnership is a standardization initiative by a large number of automotive, electronics, semiconductor, hard- and software companies.

[12] BMW, Robert Bosch GmbH, Daimler Chrysler, Opel, Siemens, Volkswagen Group, Renault, and PSA Peugeot Citroën.

real-time operating systems (OS-OSEK OS and OSEKtime OS), communication within and between ECUs (OSEK COM), and strategies for network configuration and monitoring [96].

AUTOSAR Classic Platform [97] is a successor of OSEK/VDX. Similar to OSEK/VDX, AUTOSAR Classic Platform is targeted at embedded automotive applications with hard real-time and safety constraints [98]. AUTOSAR Classic Platform specifications cover three areas: (1) a software architecture, (2) a methodology and templates for system development, and (3) compatible software interfaces at the application level [98]. OSEK/VDX specifications are reused in the AUTOSAR software architecture: the AUTOSAR operating system is backward compatible with OSEK OS (while providing more functions) [99], and the AUTOSAR communication module is also derived from OSEK COM [100].

OSEK/VDX and AUTOSAR are mature open standards. They have been realized by various open-source and proprietary implementations, some of them have been certified for use with safety-critical applications. Some examples of OSEK/VDX implementations are presented in [77, 78, 101]. For some AUTOSAR-related products, including platform implementations, tools, and training, we refer to [79, 80, 102]. These platforms are deeply integrated into the in-vehicle networks. As a result, only car manufacturers and suppliers can install software on these platforms. In fact, applications are statically configured during the production of ECUs [99, 103].

With respect to ADAS, some production platforms are QNX Platform for ADAS [87], Green Hills Platform for ADAS [88], and Wind River Helix Drive [89]. These platforms are proprietary, and their target audience is car manufacturers and suppliers. The platforms provide a rich feature set for ADAS, such as automotive connectivity (CAN, LIN, MOST, Ethernet), graphics libraries, connections to in-vehicle networks and outside worlds, and storage services (the exact features depend on the specific platform). While their focus is ADAS, some platforms additionally provide environments to run AUTOSAR software (QNX Platform and Green Hills Platform) and guest general-purpose operating systems (Green Hills Platform). These platforms are mature, with various components and tools certified to various safety standards [87–89]. However, they do not follow a unified standard, which hinders interoperability and code reuse. The AUTOSAR Adaptive Platform standard [104] is an effort to address this problem. This standard describes platforms for high-performance ECUs for highly automated and autonomous driving [104]. It will remain in draft mode until October 2018.

For multimedia and telematics applications, some popular in-vehicle platforms are QNX CAR Platform for Infotainment [81], Windows Embedded Automotive [82], GENIVI [83] (production), Automotive Grade Linux [84], Apertis [85], and Automotive Grade Android [86] (experimental). These platforms provide rich features for infotainment applications, including user interface libraries, audio/video playback, voice services, Internet, WiFi, Bluetooth, cellular connectivity, and integration of mobile devices. In addition, they provide software modules (either built-in or developed by third-parties) for in-vehicle networks access. However, in most deployed systems, access to in-vehicle networks and vehicle data is fairly limited. In Table 6, only access rights found in typical deployed systems are listed.

To support multiple applications, modern cars are often equipped with 70 to 100 ECUs, or even more in the case of luxury cars. The high number of ECUs has an impact on costs, complexity, and space problems [1, 105]. Application platforms can reduce this problem by consolidating several functions in one ECU. Multipurpose platforms, which can host applications from multiple domains and support cross-domain communication [105], can further increase the level of ECU consolidation.

There have been several proposals for multipurpose platforms. COQOS [105, 106] and the aforementioned Green Hills Platform for ADAS and QNX Platform for ADAS are three production multi-purpose platforms (the focus of QNX and Green Hills' platforms remains ADAS). These are proprietary platforms built for vehicle manufacturers and suppliers. A notable open-source experimental platform is OVERSEE [90]. It is targeted at both manufacturers and third-party developers (although the software distribution method is not detailed in the project). Regarding functionality, all mentioned multipurpose platforms support applications of mixed criticality by enforcing isolation of applications.

4.2.2. Mobile platforms

A popular alternative to embedded infotainment platforms is to run applications on smartphones. This approach has several advantages [107]. First, users are already familiar with their smartphones. Secondly, smartphones are easier and more affordable to update compared to in-vehicle systems. Thirdly, smartphones provide user information that can be used to personalize applications. Moreover, this approach allows continuous use of applications in different situations.[13] For application developers, mobile platforms provide familiar software development environments and distribution channels (i.e., application stores).

However, mobile platforms are not without disadvantages [1]. First, running applications on smartphones poses difficulties in HMI integration and guarantee of service quality for real-time applications. Secondly, access to in-vehicle networks from smartphones is limited. In addition, as automotive application platforms, smartphones can be subject to environments that they are not designed for.

Despite these limitations, several mobile platforms have been designed. Notable mobile platforms are CarPlay [91], Android Auto [92], MirrorLink [93], and AppLink [94]. These platforms use the vehicle as peripherals to a mobile smartphone, with applications running on the smartphone's CPU. The vehicles must have appropriate hardware and software to support the integration. Typical interfaces between the vehicles and smartphones are USB, WiFi, and Bluetooth. Users can interact with applications via vehicle displays, sound systems, input controls, or via the user interfaces of the smartphones.

Mobile platforms typically offer only read access or no access to the in-vehicle network. For example, AppLink has read-

[13] For example, when a user's destination is far from the parking lot, the navigation task can be divided into two steps. First, the integrated smartphone-car system navigates the user to the parking lot. After that, the user continues to use his smartphone to reach his destination.

only access to vehicle information, such as location, speed, fuel, braking, external temperature, current gear, tire pressure, and airbag status [108]. To the best of our knowledge, current versions of Android Auto and MirrorLinks do not provide vehicle sensor data to applications. The applications on these platforms are typically limited to infotainment. The most common applications are phone call, messaging, audio player, and navigation [91, 93, 94, 109]. AppLink additionally provides eCall functionality [94].

For platforms other than CarPlay, anyone can develop applications and distribute them via application stores. To develop applications for CarPlay, developers need to register with Apple's MFi program [110], which is available for companies, organizations, government entities and educational institutions [111].

4.2.3. Internet-based platforms

Instead of residing on vehicles or mobile phones, applications can communicate with vehicles over the Internet. An Internet-based application platform supports the development, deployment, and management of such applications. These platforms typically offer two main functions. First, they provide contents to in-vehicle or mobile platforms. For example, Google Play Store and Apple's App Store provide services, such as navigation, music streaming, and application distribution to Android Auto and CarPlay (see "mobile platforms").

In addition, Internet-based platforms provide a central location for the collection and storage of vehicle data (the use of such data has been discussed in Section 3.2). For this purpose, Internet-based platforms rely on in-vehicle components to extract and upload data. An instance of such components is represented by dedicated interfaces deployed by vehicle manufacturers (e.g., ECU 2.n in Fig. 3). These dedicated interfaces can be used to collect and send data to the manufacturers or their contracted providers' data centers via wireless connections [112]. In this setting, vehicle data is typically available only to the manufacturers and their partners.[14] In a proposal by the European Commission (*data server platforms* [113]), data is also collected by dedicated interfaces; however, the data is made available to more stakeholders, including manufacturers, government agencies, and third-party developers.

Another data collection method is to extend a vehicle interface with an adapter. Trucks and buses made by major European manufacturers are optionally equipped with FMS interfaces (see Section 2.2). These interfaces provide secure (firewalled) access to vehicle data for third-party applications [46]. For other vehicles, the OBD-II port can be extended with aftermarket adapters. However, the OBD-II port was originally designed to work with readers for diagnosis. As a result, attaching an adapter to the OBD-II port can cause unexpected security and privacy consequences. To this end, the European Commission has proposed an upgraded and standardized OBD port with more protection for data collection [113].

[14] For some buses and trucks, third-party application developers can access vehicle data under agreements with vehicle owners and manufacturers. Vehicle manufacturers provide the access via the remote Fleet Management System (rFMS) web service [46].

The APPSTACLE project [95] is another initiative proposing an Internet-based platform. APPSTACLE is a recently launched project involving European car manufacturers, suppliers, cellular service providers, and universities. The project aims at a secure cloud platform that interconnects vehicles and the cloud via a 5G network infrastructure. The platform also comprises an open in-vehicle platform for gathering data from in-vehicle networks.

5. Threat Analysis

With the increasing number of vehicle assets (e.g., more vehicular communication, personal information stored in vehicles, and audio-visual media), the possibilities and incentives for automotive attacks increase. This section provides an overview of important security and privacy issues related to in-vehicle systems and VANETs. First, we present relevant security concepts and their relationship. Then, we discuss security goals and requirements. Next, we study the capabilities of attackers and the impact of attacks, relating them to the security goals. Finally, we examine typical vulnerabilities and attacks. This analysis provides a baseline for an analysis of existing security and privacy measures (Section 6). We do not cover the Internet side of automotive applications. The Internet is a broad and heterogeneous area that is typically considered untrusted; in this respect, Internet technologies have many open security and privacy challenges in themselves and their analysis falls outside the scope of this survey.

5.1. Terminology

The following terminology is used in the remainder of the paper: an *asset* is a resource that has value for stakeholders. Assets may have weaknesses that can be exploited by attackers; such weaknesses are called *vulnerabilities* [114]. An *entry point* is an asset or a system component that enables access to the system. An *attack* is an intentional attempt to compromise some desired security properties of a system such as integrity, availability, confidentiality, or others [114]. In an attack, the attacker usually exploits some vulnerabilities via entry points. The goal of *threat analysis* is to identity potential attacks and determine their consequences. In particular, threat analysis includes the identification of attackers, their goals, and how they might achieve them [115].

5.2. Security and Privacy Goals

In this section, we identify security and privacy goals based on applications' needs while avoiding assumptions about potential mechanisms to achieve them.

5.2.1. Security goals

Traditionally, vehicles were isolated systems with only a few applications. The design goals were mostly limited to performance, safety, and cost. However, together with the increasing connectivity and flexibility of applications and platforms, security concerns also grow. Specifically, we consider *confidentiality*, *integrity*, *availability*, *non-repudiation*, and *data freshness*. Previous studies (e.g., [18, 63, 116]) have refined these goals to

obtain more specific security requirements (as summarized in Table 7).

Confidentiality means that only authorized individuals can access information [117]. Confidentiality is important when the information is of high value. For instance, ECU firmware and firmware updates contain intellectual properties of manufacturers and thus need to be kept confidential [116]. Moreover, applications running in ECUs may store personal information. Even if a vehicle does not have network connection with the outside world, stored information can still leak when ECUs are replaced. Personal information should be kept confidential even in these circumstances [116].

For VANET applications, vehicles frequently broadcast messages containing information, such as location, direction, and financial transaction details. Ruddle et al. [116] identify that eToll transactions and Point-of-Interest (PoI) configuration should be kept confidential between the participating vehicle and RSU. Moreover, the identity and location of the sender should be kept secret to a certain level [63, 116].

Integrity requires that information (including programs and applications) is modified only in a specified and authorized manner [117]. Integrity encompasses *authentication*, meaning that the receiver of a message should be able to verify the message origin. All applications require integrity to function correctly. However, this property is even more important in the case of safety applications.

For both VANETs and in-vehicle systems, information causing reactions by vehicles (such as automatic braking) must be authentic in terms of origin and content [116]. In addition, timing information of sensor data must be accurate [116]. To ensure the integrity of ECUs, access to ECU reprogramming function must be controlled. VANET applications also require that messages generated in the same spatial and temporal proximity are consistent [18].

Availability requires that systems work promptly and services are not denied to authorized users [117]. All applications require availability of resources (e.g., CPU, memory, communication buses and devices, and sensors), but safety-critical applications must be prioritized [116].

Non-repudiation guarantees that a party can prove that another party has performed a particular action. This property is related to *auditability* [66] and *accountability* [118]. Ruddle et al. [116] argue that non-repudiation is motivated by legal requirements from law, liability, or billing rather than functional safety.

To the best of our knowledge, non-repudiation requirements for automotive applications have not been largely investigated and the ones that have been currently identified are rather rudimentary. For example, ETSI TR 102 893 [118] specifies that *all* changes to security parameters and applications should be auditable without specifying the parameters and applications. Ruddle et al. [116] give only one example of non-repudiation requirements, namely eToll providers should be able to prove the authenticity of billing information using sensor data. Kroh et al. [63] identify several applications that need non-repudiation; however, for most applications, they do not describe which information to audit.

Data freshness denotes that received messages should be recent and have not been replayed by attackers [116]. Regarding in-vehicle networks, the freshness of messages generated by all ECUs and gateways must be ensured to prevent undesirable activation of commands. In addition, the freshness of ECU flashing commands must also be guaranteed. Regarding VANETs, freshness of messages carrying environment-related data should be ensured, especially messages that can trigger undesirable reactions.

5.2.2. Privacy goals

Privacy is a serious concern in intelligence transport systems. Applications can disseminate messages containing personally identifiable information. For example, messages typically contain vehicle IDs or pseudonyms required for authentication. In addition, for V2V and V2I applications, vehicles regularly broadcast data including location and direction. These data can be captured and analyzed to reveal the identity of vehicle occupants or to track vehicles. As a result, automotive applications and platforms must ensure *ID privacy* and *location privacy*. Kroh et al. [63] and Harding et al. [73] define these goals for VANETs, but they are also relevant to wireless sensor networks and V2D communication.

- *ID privacy:* An attacker should not be able to use messages to identify a vehicle or individual.
- *Location privacy:* An attacker should not be able to link data in safety messages to determine a vehicle's path.

On the other hand, vehicular systems might support jurisdictional access [63] for accountability purposes.

- *Jurisdictional access:* Qualified public authorities should have access to identity or location information.

5.3. Attackers

Understanding the attackers is essential to protect any system. In this section, we study the capabilities and potential impacts of automotive attackers.

5.3.1. Attacker capabilities

Multiple actors with different levels of skills and resources may be interested in attacking vehicles and associated systems (as summarized in Table 8). Based on the level of physical access, Wolf [119] classifies attackers of in-vehicle systems as *internal* or *external*. An internal attacker is typically an authorized user trying to gain additional rights (privilege escalation). Internal attackers are assumed to have full physical access to the vehicle. On the other hand, an external attacker has limited or no physical access to the vehicle. For VANET scenarios, internal attackers can be defined as the ones who possess suitable key materials to act as authenticated members of the network [18, 120].

Attackers may have a *low*, *medium*, or *high* level of knowledge and resources.[15] Wolf [119] identifies three classes of

[15] Wolf [119] distinguishes technical and financial resources. We argue that they are highly correlated, so we do not make such a distinction. In addition, we do not distinguish four levels of physical access proposed in [119] (no, limited, extensive, and virtually unlimited access). The reason is that, these levels are already captured by the "knowledge" and "resource" dimensions.

Table 7: Security requirements

Confidentiality	Integrity	Availability	Non-repudiation	Data freshness
In-vehicle systems				
• Firmware and updates must be kept secret [116]. • Exchanging ECUs does not violate the confidentiality of personal information [116].	• Information causing reactions by vehicles must be authentic in terms of origin and content [116]. • Timing information of sensor data must be accurate [116]. • Access to ECU flashing function must be controlled [116].	• The availability of communication buses, CPU, memory must be ensured for safety-critical applications [116].	• (Requirements depend on legal frameworks.)	• Freshness of messages generated by ECU and gateways must be ensured [116]. • Freshness of flashing commands must be ensured [116].
VANETs				
• eToll transactions and Point-of-Interest configurations should be kept confidential [116]. • The identity and location of the sender should be kept secret to a certain level [63, 116].	• Information causing reactions by vehicles must be authentic in terms of origin and content [116]. • Timing information of sensor data must be accurate [116]. • Messages generated in the same spatial and temporal proximity should be consistent [18].	• Safety applications require high availability of the communication systems [63]. • Strict time constraints (Table 5) should be respected [63].	• (Requirements depend on legal frameworks.)	• Freshness of messages carrying environment-related data should be ensured [116].
Impacts				
• financial • privacy	• safety • operational • financial	• safety • operational • financial	• financial	• financial • safety • operational

Table 8: Automotive attackers

#	Internal/ External	Knowledge	Financial resources	Examples
AT1	Internal	low	low	drivers, owners
AT2	Internal	medium to high	medium	mechanics
AT3	Internal	high	high	manufacturers, organized crime
AT4	External	low to high	low	thieves, vandals
AT5	External	high	high	government agencies

internal attackers with varying levels of resources and one class of external attackers with a low level of financial resource. A recent WikiLeaks disclosure [121] suggests that government agencies could also be interested in automotive attacks, making them external attackers with substantial funding. In summary, vehicles and related IT systems can be targeted by a wide range of attackers with a varying level of physical access, knowledge, and resources.

5.3.2. Impacts caused by attackers

In general, attackers can cause negative *safety, financial, operational,* or *privacy* impacts on the system and stakeholders [122]:

- *Safety*: Harming vehicle occupants and other road users.
- *Financial*: Performing unauthorized commercial transactions and disclosure of intellectual property.
- *Operational*: Interfering with the intended operational performance of non-safety functions.
- *Privacy*: Unauthorized access to data about the activities and identity of vehicle owners or drivers.

The type of impact typically depends on which security goal is compromised. The last row of Table 7 shows the relation between the security goals presented in the previous section and the impacts of attackers. Breaches of confidentiality usually have privacy and financial consequences. For instance, manufacturers can install devices to collect user data, affecting passengers' privacy. Similarly, government agencies might perform mass surveillance. On the other hand, a counterfeiter may steal intellectual properties, causing financial loss for manufacturers. The disclosure of sensitive information like payment transactions to a third-party can cause both financial and privacy loss to users.

Compromising the integrity of system components and data can have safety, operational and financial impact. For example, vehicle owners or technicians may modify a component in order to circumvent software and hardware restrictions, activate locked features, or manipulate driving records, causing financial loss for manufacturers and application providers. In addition, these modifications could have an impact on both safety and operational performance of the vehicle. For example, an attacker can trigger or disable brakes by injecting bogus messages into the CAN bus (or altering CAN messages), which can have serious implications on vehicle safety and operation.

Loss of availability can have financial, safety and operational consequences. For instance, an attacker might compromise the normal working of safety-critical applications like AP14–AP17 in Table 5, thus affecting safety, or other non-safety applications like AP5 and AP6 in Table 5, affecting the operational performance of vehicles. Clearly, the unavailability of applications and platforms, especially the ones that are Internet-based such as applications AP11–AP13 in Table 5 and platforms PL5–PL22 in Table 6, disrupts business, causing financial loss for service providers.

Violating non-repudiation requirements has mainly a financial impact as stakeholders cannot prove that given transactions occurred and/or an attacker could deny (legal) responsibilities.

The freshness of data is important for safety applications,

such as AP1–AP4 and AP14–AP17, and non-safety applications like AP10 and AP11. For instance, the intersection collision warning application (AP15) can incorrectly react to late location information of a remote vehicle, causing accidents. In instant messaging applications (AP11), delayed or out-of-order messages interrupt conversations. In addition, data freshness can have financial implications as, without guarantee of data freshness, an unwanted transaction might be performed.

It is worth noting that the violation of a security requirement can have other (indirect) impacts in addition to the ones listed in Table 7. This is because an attacker might have to compromise a security requirement in order to compromise another security requirement. For example, if non-repudiation is not guaranteed, attackers have more opportunities to disseminate bogus information (a violation of integrity) to harm user safety. In this case, the breach of integrity directly causes user harm, while the compromise of non-repudiation affects safety indirectly. As another example, attackers may first compromise the confidentiality of ECU firmware and then find exploits to compromise firmware integrity, thus affecting the normal operation of vehicles. Therefore, the violation of integrity is the direct cause of operational abuse while the compromise of confidentiality impacts operation indirectly.

5.4. Vulnerabilities

Vehicles and related IT infrastructure consist of numerous software and hardware components (Section 2). Many of them can exhibit weaknesses that can be exploited. Table 9 summarizes common vulnerabilities. An underlying reason for these vulnerabilities is the addition of communication, applications, and platforms to automotive technologies that were designed with little consideration for security and privacy. The lack of measures against IT security threats was not a serious issue when vehicles were isolated and applications were mainly for control systems. However, when vehicles are connected and accessible remotely, the lack of security measures becomes more exploitable. In addition, vehicles have a long life-time. During that time, new attacks are proposed while older technologies and vehicles are not always updated to protect against them. Next, we discuss the main vulnerabilities that affect automotive systems.

5.4.1. In-vehicle systems

ECUs. In some vehicles, the control of re-flashing and diagnostic functions is performed by a challenge-response protocol with fixed challenges [4][16]. Moreover, there are ECU implementations that deviate from standards, resulting in a weaker control and more capabilities for attackers (e.g., re-flashing ECUs or disabling communication between ECUs while driving) [4].

Among ECUs, telematics systems present a large attack surface. They contain communication interfaces that can be exploited as entry points by attackers. In addition, telematics systems include a large amount of software for communication protocols and applications. Such non-trivial software typically contains implementation and configuration defects, for example, buffer overflows or unnecessary services. Via the communication entries and software defects, attackers can target the telematics systems in a fashion similar to attacks against traditional IT systems. If the telematics systems are not adequately segregated from critical in-vehicle networks, they can be used as stepping stones to perform more damaging attacks, including control of safety-critical actions.

In-vehicle networks. The design of automotive buses such as CAN and FlexRay was mostly based on safety and cost requirements. Security concerns were hardly considered. Until recently this was not a serious problem. However, nowadays applications and vehicles are increasingly connected to the external world, with the implication that not all messages send over the buses can be trusted.

The CAN bus lacks security measures to ensure confidentiality, integrity, authenticity, availability, and non-repudiation [4, 8]. First, CAN messages are not encrypted and are broadcast to every node on the same bus; an ECU can read all messages sent by other ECUs on the same bus or forwarded by gateways. Secondly, a compromised ECU can flood the bus or send high-priority messages to deny messages from other ECUs. Third, CAN messages do not contain sender/receiver identification fields or other authentication mechanisms. Moreover, the checksum field in CAN messages can detect transmission errors but not purposely forged messages. In addition, there are no mechanisms for an ECU to prove that it has received a message. Finally, depending on the network configuration, some non-safety-critical components can send messages to a safety-critical network.

The FlexRay protocol also lacks built-in security to guarantee confidentiality, authentication, and data freshness [124]. Some protection is available for availability and integrity (e.g., time division multiplexing to ensure availability, and checksum to detect transmission errors), but it is mainly intended for safety, and thus protects against faults but not necessarily against attacks.

Wireless sensor networks. Wireless sensor networks such as TPMS and remote key entries can also contain vulnerabilities. Because of the wireless communication channel, messages can be captured and replayed. For example, Francillon et al. [9] show that it is possible to relay messages between a vehicle and its key over a distance up to 50 m.

Other problems are related to the design and implementation of communication protocols. Typical problems include the lack of cryptographic protection or the use of less validated ones, reliance on obscurity for security, bad programming practices, and insufficient hardware protection against side channel attacks. For instance, Rouf et al. [7] reverse-engineer a TPMS and show that it exhibits several vulnerabilities. First, messages in this TPMS are unencrypted and can be spoofed easily. Second, in-vehicle systems accept incoming messages without input validation. In addition, messages sent by the TPMS sensor contain message IDs, which enable tracking. Another example is KeeLoq [126], a proprietary encryption algorithm used in

[16] The motivation for fixed challenges is to avoid storing the challenge-response algorithm on ECUs [4].

Table 9: Vulnerabilities

			Vulnerability	Examples
Component	Hardware	V1	Lack of hardware protection	• Weak protection against key extraction in KeeLoq [123]
	Software	V2	Software bugs	• Stack overflow, heap overflow, redundant services [6, 39] • Lack of input validation [7]
		V3	Weak device authentication	• Fixed challenges to authenticate diagnostic devices [4]
		V4	Deviation from standards	• Possibility to re-program ECUs during rides [4]
Communication	In-vehicle networks (wired)	V5	Protocols designed without security in mind.	• CAN bus susceptible to DoS [4, 8] • ECUs filter messages by unprotected message IDs [4].
		V6	Lack of segregation among domains	• Telematics systems can send messages to safety-critical bus [6, 39]
	Wireless networks	V7	Unreliable communication channel	• Wireless channels are typically vulnerable to jamming and congestion [19].
		V8	Messages are broadcast	• Messages can be captured, analyzed, and replayed [7, 9]. • Anyone can send messages [7, 9].
		V9	Weak cryptography/ Security by obscurity	• Weak encryption in KeeLoq remote key entries and Megamos immobilizers [10, 11] • TPMS messages sent in the plain [7]
Environment		V10	High availability requirements vs. limited resources	• Possible trade-offs between real-time requirements and DoS resistance [17].
		V11	High vehicle mobility	• Possible trade-offs between lower interaction time and protocol security [17].
		V12	Adding new systems to existing architecture	• Systems designed without security and privacy in mind becomes vulnerable [4, 7, 8, 124]. • Cryptographic primitives got broken [10, 11].
		V13	Long vehicle life-time	
		V14	Manufacturers' control	• Manufacturers install interfaces for data collection [112]. • Manufacturers program ECUs to cheat in emission tests [125].

remote keyless entry systems. The details of the encryption algorithm and the authentication protocols built on top of it were leaked in 2006 [126]. Various practical cryptanalysis and side-channel attacks on the encryption algorithm and protocols have been proposed [11, 123]. As another example, the Megamos Crypto immobilizer uses a proprietary cryptographic algorithm and a proprietary protocol. The algorithm, the protocol, and their implementation have all been broken [10].

V2D communication. Devices connected to a vehicle, such as OBD-II devices and smartphones, can be abused to attack in-vehicle systems. OBD-II testers and aftermarket adapters typically have their own communication interfaces like USB, Bluetooth, and cellular interfaces. Via these interfaces, an attacker can compromises OBD-II devices, which can be used as stepping stones to attack in-vehicle systems.

Smartphones can be infected with malware. In-vehicle applications and platforms must be protected against such malware. Protection mechanisms currently in place are sometimes inadequate. For example, MirrorLink trusts all contents from registered smartphones (which may have been infected) and contains exploitable bugs [12].

In-vehicle equipment for V2V and V2I communication. In-vehicle equipment for V2V and V2I communication could also serve as an entry point for attackers. As any other connected computers, ECUs for VANETs can suffer from implementation and configuration bugs. In addition, these devices contain cryptographic keys that can become targets of side-channel attacks.

Other vulnerabilities. Vehicle manufacturers have full control over the manufacturing process. They can exploit gaps in manufacturing regulations and conformance testing procedures. For example, loose privacy regulations will allow manufacturers to collect more driving data than necessary [112]. As another example, the current method of emission testing (measuring emission when the vehicle's behavior simulates a predefined speed profile) allows manufacturers to program their ECUs to detect such tests and behave differently from under normal conditions [125].

5.4.2. VANETs

V2V and V2I communication are wireless. Without additional protection, wireless communication is susceptible to congestion, jamming, transmission errors, eavesdropping, bogus messages, and tracking. Therefore, wireless communication usually needs built-in protection. Compared to other wireless networks, VANETs have additional constraints that make their protection more difficult. While these constraints are not intrinsically vulnerabilities, they require novel protection mechanisms or trade-offs between security and performance. This could lead to vulnerabilities if the proposed mechanisms are insufficiently investigated. Some noteworthy constraints are:

- System resources (processing power, memory) and connectivity (bandwidth) are limited while many applications have stringent timing constraints (e.g., AP14–AP20 in Table 5). In addition, the communication channel typically has limited bandwidth, while the number of vehicles can be large [73].
- Vehicles have a high level of mobility and the network topology can change quickly. As a result, we cannot always rely on reputation-based schemes (rating other vehicles based on their messages) and protocols requiring significant interaction between vehicles [17].

5.5. Attacks

The large number of vulnerabilities in automotive systems opens the opportunity for a variety of attacks. Table 10 summarizes notable attack methods, the most likely attackers that use such methods, the vulnerabilities that could be exploited to perform these attacks, and the security and privacy goals affected by the attacks.

5.5.1. In-vehicle systems

An attacker who has direct access to in-vehicle networks can perform a wide range of malicious activities, some of which can directly affect safety and vehicle operation. For instance, by injecting messages into CAN buses, an attacker can affect integrity, data freshness, and availability requirements. In particular, he can reprogram ECUs, control display and sound systems, control the instrument cluster (e.g., display falsified sensor data, display random messages, and adjust brightness level), control body components, interfere with the engine operation, lock and disable brakes, prevent the car from turning on or off, etc. [4, 6], or install counterfeit components [8]. To perform DoS on CAN, an attacker may send fragments instead of complete CAN frames [13].

Communication interfaces could increase the access range, making attacks more scalable. Multimedia and telematics systems are of particular interest because they contain multiple communication interfaces and a large amount of software. Checkoway et al. [39] and Miller and Vaselek [6] demonstrate that it is possible to send CAN messages from telematics ECUs after compromising them via their interfaces. Therefore, this type of attacks has the same effects as directly injecting messages into CAN buses, affecting in-vehicle systems' integrity, data freshness, and availability. In addition, it has been shown that an attacker can record voice data and track the target's location, violating confidentiality, identity privacy, and location privacy. The interfaces typically involved in those attacks are media players, Bluetooth, TPMS, FM radio [39], and cellular [6, 39] (each attack exploits a separate interface). The enabling vulnerabilities are implementation and configuration bugs in the telematics ECU, inadequate segregation of sub-networks, insecurities of the CAN bus, and weak control of ECU flashing functions.

Short-range wireless sensors have also been exploited. By spoofing TPMS sensor messages, Rouf et al. [7] were able to turn on low tire pressure warnings and crash the ECU processing the messages, compromising the system's integrity and availability. This attack is possible because the ECU does not authenticate the sensor and lacks input validation. In addition, the TPMS allows vehicle tracking from 10–40 m depending on the tracking equipment. In another attack, Verdult et al. [10] describe three ways to lockpick the Megamos Crypto immobilizer system. In addition, the cryptographic key stored in the physical ignition key can be overwritten and hence can no longer be authenticated. Specifically, this attack compromises the confidentiality and integrity of cryptographic keys and affects the availability of the ignition key (in the case of key overwriting) or of the whole vehicle (in the case of car theft). The attack exploits the vulnerabilities in the immobilizer's proprietary cipher, protocol, and their implementation. In a different attack, Francillon et al. [9] demonstrate how to relay messages between cars and keys to enter and start a vehicle, violating data freshness and, ultimately a vehicle's availability. It is worth noting this attack works regardless of the underlying encryption algorithms and protocols.

Instead of directly targeting a vehicle, attackers can first compromise external devices. Several OBD-II devices have been exploited to inject messages into in-vehicle networks in order to compromise a vehicle's integrity, data freshness, and availability requirements. For example, Checkoway et al. [39] compromise vulnerable OBD-II testers, using them to inject messages into the CAN bus. In [5, 127, 128], aftermarket OBD-II adapters are abused in both local and remote attacks; some of which enable the execution of safety-critical actions.

Smartphones represent another type of exploitable external devices. Checkoway et al. [39] create an Android Trojan horse to run arbitrary code on a paired telematics unit. It exploits a vulnerability in Bluetooth handling software of the telematics unit. Because of inadequate segregation of sub-networks, the compromised telematics unit was able to send CAN messages. As a result, this attack can violate integrity, data freshness, and availability requirements of in-vehicle networks. Mazloom et al. [12] propose an attack on a MirrorLink implementation. This implementation enables an in-vehicle program (client) to receive and present contents sent from a smartphone. The client program trusts the contents sent by the smartphone and the client has memory corruption vulnerabilities, which are not mitigated by the underlying application platform. By exploiting these vulnerabilities, an attacker, who has compromised the smartphone, can modify the control flow of the client program to display custom debug messages, thus disrupting message integrity. Mazloom et al. conjecture that similar attacks could be mounted to send CAN messages.

Vehicle manufacturers have full physical access to vehicles during production and have the expertise to perform severe attacks. In a recently discovered attack, several companies employ software to cheat in emission tests, hiding the fact that their vehicles are not compliant with emission standards [125]. Under test conditions, this software alters the vehicle' behavior by sacrificing performance for compliance, which however is not the behavior in normal driving conditions. We can see this attacks as a violation of the integrity of emission test results and, in general, of the normal functioning of the vehicle. In another attack, a majority of vehicle manufacturers collect driving history from vehicles operating in the United States without sufficient communication with car owners, clear statements of collection purposes, and the means to secure data [112], leading to breaches of confidentiality, identity privacy, and location privacy.[17]

5.5.2. VANETs

V2V and V2I technologies have not been widely deployed and, thus, there is little information on specific incidents and

[17] It is unclear from [112] how data is collected, but we can infer that manufacturers install some form of dedicated hardware for data collection, including a cellular communication interface as shown in Fig. 3.

Table 10: Attacks on in-vehicle systems and VANETs

Method	Attacker	Violated goals	Enabling vulnerabilities
In-vehicle systems			
Inject CAN messages via ECUs connected to the bus [4, 6, 8, 13]	AT2	I A F	V5, V6
Remotely control telematics systems/ OBD-II devices/ smartphone [5, 6, 39, 127, 128]	AT4	C I A F PL PI	V2
Inject CAN messages via telematics systems/ OBD-II devices/ smartphones [5, 6, 39], [127, 128]	AT2 AT4	I A F	V2 – V6
TPMS spoofing [7]	AT4	I A	V2, V8, V9
TPMS tracking [7]	AT4	C PL	V8, V9
Software attacks on remote keys and immobilizers [10]	AT4	C I A	V2, V9
Attack cryptographic algorithms and protocols [10, 11]	AT4	C I A	V1, V9
Side-channel attacks on components containing cryptographic keys [123]	AT2	C I	V1, V9
Relay attacks on remote keys and immobilizers [9]	AT4	A F	V8
Circumvent exhaustion standard tests [125]	AT3	I	V14
Install devices for data collection [112]	AT3	C PL PI	V14
VANETs			
Jamming [17–20, 118]	AT4	A	V7, V10
Message suppression [17–21, 118]	AT2	A	V1, V2, V9
Message fabrication [17–21, 118]	AT1 AT2	I A N	V1, V2, V8, V9
Replay [17, 19–21, 118]	AT4	F	V8
Wormhole [18, 20, 118]	AT2	I	V8
Sybil [17, 19–21, 118]	AT2	I A	V1, V2, V8, V9
Eavesdropping [18–21, 118]	AT4	C PL PI	V8
Man-in-the-middle [17, 20, 118]	AT4	C I F N PI	V8, V9

Legend:
C: Confidentiality I: Integrity A: Availability F: Data freshness N: Non-repudiation PL: Location privacy PI: Identity privacy

attack vectors. Nevertheless, potential (generic) attack classes have been identified based on characteristics of wireless channels and additional constraints of VANETs. We only briefly describe these attacks and refer to existing literature [17–21, 36, 118, 129, 130] for more details. This section focuses on the relationship between these attacks and vulnerabilities, security goals, and privacy goals (as summarized in Table 10).

Jamming. The attacker sends interfering signals to prevent communication. This attack can affect the availability of any application depending on wireless communication. This is a low-effort attack which does not require compromising cryptographic mechanisms [19]. The attacker does not need high transmission power either, because the network coverage is well-defined within VANETs.

Message suppression. The attacker selectively drops messages from the network. In addition, the attacker can trap messages by claiming that he is in the best position to forward them to a destination (the so-called *black hole* attack). As a result, messages needed by applications become unavailable. To perform this attack, the attacker needs to compromise the in-vehicle application or platform responsible for forwarding messages.

Message fabrication. The attacker creates messages containing bogus information. This attack class also includes *spamming*—where attackers send a large volume of messages to exhaust the resources of receiving parties, such as OBUs and service centers. In simple attacks, the attacker could abuse legitimate functions, e.g., SOS services or instant messaging. More sophisticated attacks require compromising hardware or software of in-vehicle platforms, or using equipment such as GPS simulator (to broadcast wrong positioning information).

Replay. The attacker captures messages and replays them at a difference place or time, violating data freshness. This attack causes receivers of replayed messages to misperceive the environment, e.g., to persuade other vehicles that there is a traffic jam and divert them from a route. The attacker must bypass protections such as timestamps, if these protections are employed.

Wormhole. Two colluding vehicles capture or generate valid messages and tunnel them through a high-speed communication channel. The tunneled messages are broadcast at a wrong location and introduce faults in applications that depend on location information.

Sybil. An attacker creates numerous false identities to influence the system's behavior [131], compromising its integrity and availability. Specifically, one vehicle can send messages associated with multiple identities at the same time to create the illusion that the messages come from multiple vehicles, for example,

to deceive other vehicles that there is a traffic jam. The attacker must obtain multiple identities or pseudonyms, for instance, by extracting them from VANET applications and platforms.

Eavesdropping. The attacker listens to message broadcast in wireless channels. This could lead to location tracking, identity revealing, or revealing information about financial transactions. This attack is possible because (1) vehicle IDs or pseudonyms need to be broadcast for authentication purposes, (2) vehicle location information is required by several applications (e.g., AP14, AP15, AP20 in Table 5), and (3) additional information may be sent without sufficient protection.

Man-in-the-middle. The attacker listens to and possibly modifies messages before forwarding them to another receiver. This attack impacts confidentiality and privacy, and integrity if messages are modified. The attacker must defeat protection mechanisms such as digital signatures, if such mechanisms are employed.

6. Security and Privacy Mechanisms

As discussed in Section 5, there is a wide range of attackers with different capabilities and skills. The impact of attacks can range from slight inconvenience to serious financial, privacy, and safety consequences. Security and privacy mechanisms are essential to avoid these negative impacts and maintain customer confidence.

Many mechanisms have been proposed to address security and privacy threats. Due to the huge variety of mechanisms, this section aims to identify the current trends in automotive security and privacy rather than to give an exhaustive list and description of all mechanisms. To understand the various mechanisms, we present them from different perspectives:

1. *Stage view.* This view provides a classification of mechanisms based on the stages of attacks in which they act. We consider four classes: *prevention* mechanisms aim to block unauthorized external accesses and interaction within the vehicle, *deflection* mechanisms aim to divert attacks from system assets, *detection* mechanisms aim to discover attacks and abuses, and *response* mechanisms react to identified attacks.[18] Table 11 summarizes this view.
2. *Purpose view.* This view describes the typical attacks or undesirable situations that the mechanisms target, and their security and privacy goals. An overview of this view is presented in Table 12.
3. *Application platform view.* This view describes the changes to the application platform needed to accommodate the mechanisms, including changes to hardware components, software libraries, and management mechanisms. This view provides an indication of the cost for implementing and deploying the mechanisms. Table 13 summarizes this view. Mechanisms that can be implemented at the application level without changes to platforms are not reported in the table.

The remainder of this section reviews the mechanisms proposed to secure vehicle systems by discussing each mechanism with respect to these views.

6.1. In-vehicle Systems

ECUs. A main concern for vehicle security is to protect the hardware and applications running inside ECUs. To this end, the EVITA project has proposed hardware security modules (HSMs) for secure boot, processing, and storage [133]. To run multiple applications on the same ECU securely and to regulate the resources available to applications, a number of isolation techniques have been proposed [90, 136, 137]. The main techniques for application isolation are virtualization, containers, and microkernel. The main idea of virtualization is that a *hypervisor* hosts multiple virtual machines, each of them runs a separate operating system. Using containers, an operating system runs processes in different name spaces; each name space presents a different view of system resources to these processes. A microkernel provides only fundamental services needed to build applications on top of it; other services, such as device drivers and user interfaces, are delegated to applications. Many of the platforms mentioned in Table 6 implement isolation techniques. For example, QNX CAR for Infotainment [81] uses a microkernel. Apertis [85] supports containers. The OVERSEE project [90] uses a microkernel that also serves as a hypervisor. Although isolation techniques can theoretically provide an effective line of defense, platforms might still remain insecure because of bugs in the implementation of isolation techniques or poor isolation policies. Despite advances in formal verification, the verification of policies and their implementations still requires enormous effort, especially for complex systems such as automotive platforms hosting heterogeneous applications.

Firmware updates (reprogramming/reflashing) are often used to patch vulnerabilities in ECUs. The most common method today is over-the-wire update, in which a technician connects a serial communication tool to the in-vehicle networks to access the target ECU. With the increasing trend of connected vehicles, over-the-air (OTA) firmware update has gain more attention. OTA updates allow more frequent and location-independent updates, reduction of warranty cost, and centralized servers (which means updates are not distributed to multiple dealers) [189]. However, compared to over-the-wire updates, which are usually performed in a controlled environment such as a garage, OTA updates are exposed to a larger attack surface. Thus, OTA updates must satisfy several constraints to ensure vehicle operation and safety [189], including: (1) updates should be performed at the right moment, which depends on various factors, such as remaining battery, whether the target ECU is operating, whether the vehicle is moving; (2) ECUs should work after updates without human intervention; (3) the update mechanism should withstand a period of ten years or more, and it should be upgradeable.

[18] Our classification is inspired by Nilsson and Larson's defense-in-depth paradigm [132]. Compared to this paradigm, we introduce a class for *response* mechanisms and consider forensic mechanisms as a sub-class of detection mechanisms.

Table 11: Security and privacy mechanisms—Stage view

Target components	Prevention	Deflection	Detection	Response
In-vehicle systems				
ECUs	• Hard-ware security module [133] • (OTA) firmware update [132, 134, 135] • Application isolation [90, 136, 137]		• Forensic [8, 125]	
In-vehicle networks	• Hiding system states [138] • ECU and message authentication [35, 97, 132, 138–143] • Gateway firewall [35, 144, 145] • Replacing CAN bus [13]	• Honeypot [146]	• Intrusion detection [147–151] • Forensic [8, 132] • Honeypot [146]	• Adaptive dynamic reaction [147] • Limit CAN communication of non-safety-critical subnets under attacks [152]
Wireless sensor networks	• Standard algorithms and protocols • Distance bounding			
V2D	• Disabling OBD-II ports while driving [153] • Upgrading OBD ports [113] • Device, application, and content type attestation [154]			
VANETs				
V2V/V2I	• Digital signatures and PKIs [36, 73, 133, 155–157]. - Congestion control [158–162] - Reducing computational overhead [163] - Pseudonym change [164, 165] • Alternatives: - Symmetric cryptography [166–169] - Group signatures [170–174] - Identity-based encryption [174–176]		• Misbehavior detection [177–184]	• Misbehavior reporting [178, 185] • Certificate revocation [178, 185] • Revocation information dissemination [186–188]

Various schemes for OTA updates have been proposed in the last years. For example, Nilsson and Larson [132] propose a protocol ensuring firmware integrity, confidentiality, and freshness. However, this proposal does not discuss the requirements that a platform should meet to support the protocol. Idrees et al. [134] provides a complete hardware-software architecture for OTA updates. In this scheme, integrity requirements are guaranteed through the use of HSMs. The scheme notifies the user about available updates and only downloads the new firmware after the user has approved it. Updates are only possible when the vehicle is in an idle state and has access to the necessary infrastructure. In case of errors during flashing, the process stops and puts the ECU in a locked state, thus not satisfying requirement (2) above.

Detection mechanisms have been proposed to discover ECUs cheating during emission tests. For instance, Contag et al. [125] reverse engineer and statically analyze ECU firmware images to detect ECUs that deliberately identify the execution of emission tests. The identification of emission tests indicates the intent to evade such tests. In particular, a firmware image is recognized as cheating if it actively compares the distance traveled over time to the distance/time profiles of known emission tests. However, manufactures can use other profiles, e.g. based on speed over time, to identify emission tests and, thus, avoid detection.

In-vehicle networks. As discussed in Section 5.4, CAN lacks security measures to ensure confidentiality, integrity, authenticity, availability, and non-repudiation. For instance, Palanca et al. [13] show its weaknesses against DoS attacks. We observed two main trends for addressing the inherent limitations of CAN technology. On the one side, a number of mechanisms have been proposed to compensate for the weakness of the CAN protocol. For instance, to thwart selective DoS attacks, Glas et al. [138] propose to hide system states by encrypting CAN messages. In addition, various ECU and message authentication schemes have been proposed to ensure the integrity and freshness of CAN messages [35, 97, 132, 138–142]. Among others, the AUTOSAR standard [97] defines a Secure Onboard Communication module that performs message authentication and freshness verification. The standard proposes the use of message authentication codes (MAC) for efficiency. To generate and verify MAC, two ECUs need to share a secret key. This creates an attack vector, especially if the same keys are used during the whole vehicle life cycle or shared across vehicles. A viable solution to address this issue is the use of asymmetric cryptography and certificates to authenticate ECUs and share symmetric keys [143]. These operations, however, should be executed while the vehicle is in an idle state due to the low performance of asymmetric cryptography.

Another trend aims at the replacement of CAN technology. In particular, several authors have proposed Ethernet as a suitable replacement for CAN [28, 31–33]. However, their main concerns are bandwidth, latency, and error rates while the impact of Ethernet on the security of in-vehicle network has not been investigated. In general, the adoption of new network technologies requires major changes in network architectures, ECU hardware, and application platforms. These changes can have a significant impact on the system functioning as well as on the applicability of other security mechanisms.

With the increasing connectivity of vehicles, gateways should implement firewalls to allow only authorized ECUs to send messages to safety critical-networks and to regulate the exchange of information across functional domains. Wolf et al. [35] argue that a reliable gateway firewall should filter traffic using authorization information (e.g., digital signatures), but this in-

Table 12: Security and privacy mechanisms—Purpose view

Mechanism	Prevented attacks/undesirable scenarios	Security and privacy goals
In-vehicle systems		
Hardware security modules [133]	Prevent side-channel attacks	C
	Prevent unsigned software from starting	I
	Increase performance of cryptographic functions	A F
(OTA) firmware update [132, 134, 135]	Correct known firmware defects †	
Application isolation [90, 136, 137]	Prevent privilege escalation	C I A F
Forensic [8, 125]	Detect unusual events that happened within the vehicle	N
Hiding system states [138]	Prevent DoS of CAN bus and ECUs	A F
ECU and message authentication [35, 97, 132, 138–143]	Prevent injection of CAN messages	I F
Gateway firewall [35, 144, 145]	Prevent CAN injection that crosses application domains	I A F
Replacing CAN bus [13]	Remove inherent weaknesses of CAN bus	C I A F
Honeypot [146]	Redirect attacks from system assets †	
	Collect information about attacks †	
Intrusion detection [147–151]	Detect intrusion attempts †	
Limit CAN communication of non-safety-critical subnets under attacks [152]	Mitigate the effects of CAN message injection	I A F
Standard algorithms/protocols	Prevent exploitation of weak algorithms and protocols †	
Distance bounding	Prevent relay attacks	I F
Disable/Upgrade OBD-II ports [113, 153]	Prevent injection of CAN messages via OBD-II devices	I A F
Device/application/content type attestation [154]	Prevent remote control of telematics systems	C I A F
	Prevent CAN message injection via smartphones	I A F
VANETs		
Digital signatures and PKIs [36, 73, 133, 155–157].	Prevent attacks from external attackers, including spamming, message fabrication, replay, man-in-the-middle attacks, and revealing ID or other information	C I A F PI
Congestion control [158–162]	Mitigate congestion of communication channels	A F
Reducing computational overhead [163]	Avoid exhaustion of system resources	A F
Pseudonym change[164, 165]	Prevent location tracking	C PL
Misbehavior detection [177–184]	Detect internal attackers †	
Misbehavior reporting and revocation [178, 185–188]	Remove known attackers †	

†: This is a generic method able to prevent and/or mitigate multiple types of attacks.
Security goals:
C: Confidentiality I: Integrity A: Availability F: Data freshness N: Non-repudiation PL: Location privacy PI: Identity privacy

formation is often not provided by ECUs. Alternatively, firewalls can filter traffic by domains, preventing ECUs in non safety-critical domains from sending messages to safety-critical domains [35]. Building a reliable automotive firewall is challenging. First, it should satisfy common firewall design goals, which include [190]: (1) all traffic from inside to outside and vice-versa should pass through the firewall, (2) only traffic conforming a security policy is allowed to pass, and (3) the firewall should be immune to penetration. In particular, requirement (1) can be hard to achieve due to the choice of network architecture and the several external interfaces. For example, one ECU in the architecture in Fig. 2 can be connected to two buses, making this architecture incompatible with the gateway design. Another example is the cellular interface on ECU 2.n in Fig. 3, which connects a safety-critical network without passing through the gateway. In addition, a gateway firewall should satisfy automotive requirements [144]. It should have real-time capabilities to accommodate real-time applications and operate reliably without reboot. In addition, it should boot quickly and should be updatable due to the long lifetime of vehicles. A possible approach to achieve real-time capabilities is to combine software and hardware firewalls [145]. A hardware firewall is used to handle simple and generic rules. Messages passing through this firewall are further filtered by a software firewall that implements more complex rules. However, the design proposed in [145] is based on a simplified Ethernet-based network architecture and traffic model. In addition, reliability and updatability of the firewalls have not been investigated. Therefore, more development and testing for real-world architectures and other network technologies like CAN are needed.

Honeypots have been proposed to detect attacks, deflect them from real assets and collect information about attacks. In the context of vehicle security, Verendel et al. [146] propose three honeypot models based on three ways to simulate in-vehicle network traffic. The first model uses prerecorded network traffic. The second model retrieves real traffic from the in-vehicle network

Table 13: Security and privacy mechanisms—Application platform view

Security mechanisms	Added/changed platform components
Hardware security module [133]	• The HSM itself • Software to manage access to the HSM
(OTA) firmware update [132, 134, 135]	• Wireless-capable gateway • Management software • Additional storage capacity
Application isolation [90, 136, 137]	• Platform mechanisms (virtualization, container, or microkernel) • Management software
Forensic [8, 125]	• In-vehicle hardware and software to collect and store evidence • External hardware and software to collect and process evidence
Hiding system states [138]	• Cryptographic protocols on top of CAN protocols • (Optional) hardware security modules
ECU and message authentication [35, 97, 132, 138–143]	• Cryptographic protocols on top of CAN protocols • (Optional) hardware security modules
Gateway firewall [35, 144, 145]	• Software/hardware for firewalls
Replacing CAN bus [13]	• Major changes to in-vehicle network architectures • ECU hardware and firmware changes to accommodate new network technologies
Honeypot [146]	• Hardware and software for the honeypot • Management software
Intrusion detection [147–151]	• Hardware and software components to collect and process evidence • Management software
Limit CAN communication in presence of attacks [152]	• CAN network architecture
Standard cryptographic algorithms/protocols	• Cryptographic library • (Optional) hardware security modules
Upgrading OBD-II port [113]	• Development of an enhanced OBD-II port
Digital signatures and PKIs [36, 73, 133, 155–157].	• Hardware for creating, sending, and processing messages. • Cryptographic library
Misbehavior detection, reporting, and revocation [177–188]	• Hardware to collect and store evidence • Management software

via a unidirectional channel. Both models are irresponsive to the actual attack and, therefore, vulnerable to detection and evasion. The third model generates network traffic from environment data (e.g., sensor readings), the attacker's command or data, and the driver's reaction to the environment and attack (e.g., braking). Both environment data and the driver's reaction are simulated. This model is more interactive and can be more resilient against detection depending on how realistic the simulation is.

Another class of detection mechanisms is represented by network intrusion detection systems. These systems aim to identify attacks by analyzing the network traffic. Existing network intrusion detection systems for in-vehicle networks can be grouped into two main classes: signature-based and behavioral-based. Signature-based systems detect malicious behavior by matching messages against signatures of known attacks. They are usually characterized by a low number of false positives but cannot detect unknown attacks. This issue is addressed by behavioral-based intrusion detection systems [147–151] that learn a model of normal behavior and rise an alert whenever the observed behavior does not match such a model. However, building a behavioral-based intrusion detection system for in-vehicle networks is challenging due to the large number and heterogeneity of ECUs deployed in vehicles. In addition, CAN messages provide very limited information, making it difficult to build accurate models of normal behavior. This is exacerbated by the fact that the meaning of CAN messages are specific to manufacturers and vehicle models. Therefore, being not able to rely on semantic models of normal behavior, existing behavioral-based intrusion detection systems exploit other properties of CAN messages like message frequency [147–149], clock skew [150], or message entropy [151]. As a consequence, these systems are only able to detect attacks involving periodic messages but not attacks involving sporadic and irregular messages.

The use of forensic science has been suggested for off-line detection of attacks [8, 132]. Hoppe et al.[8] propose a six-phase forensic process for automotive. The first phase is *strategic preparation*, which happens when vehicles are designed and manufactured. This phase provides facilities to log ECU and network activity. The second phase is *operational preparation*, which starts after an incident. During this phase, potential data sources, tools, network architecture, and potential influences of some actions (e.g., the loss of volatile data when powering off) etc., are documented. This phase is followed by a *data gathering* phase, where all potential data related to the incident are collected. After that, the data is transformed into a human-readable format in the *data investigation* phase. Then, during the *data analysis* phase, relevant data from different ECUs are put in a timeline and a causal context. The final phase is *documentation*, where investigation findings are reported. Among these phases, strategic preparation can require a redesign of vehicles' architecture, thus increasing their cost, but it can greatly streamline the subsequent steps.

To respond to detected attacks, Hoppe et al. [147] introduce the concept of *adaptive dynamic reaction*, in which the driver is notified about attacks using visual, acoustic, or haptic devices depending on the severity of the attack and the current situation (e.g., whether the vehicle is in a noisy environment). This approach, however, requires that detection mechanisms provide detailed information about attacks (e.g., their severity). In another work, Caberto and Graham [152] propose a fail-safe mode. In this mode, (1) cross-domain communication is disabled, and (2) in the compromised sub-networks, communication is limited to safety and control functions. This method is based on the assumption that ECUs can be precisely grouped into functional domains without dependencies affecting safety and control functions. In addition, during the fail-safe mode, because the instrument cluster is disconnected from the ECUs for control functions, the cluster displays outdated information, causing negative consequences for users.

Wireless sensor networks. A lesson learned from attacks on TPMS and immobilizers is that proven cryptographic primitives and authentication protocols are preferable to relying on obscurity. In addition, measures against relay attacks such as distance bounding should be employed [9]. Distance bounding can be used to determine an upper-bound of the distance between a vehicle and its key. If the distance is larger than a given threshold, the key cannot open or start the vehicle. In recent years, distance bounding protocols have significantly advanced and, as demonstrated in [191], they can be implemented in smartcards, making them applicable in real settings.

V2D communication. V2D communication can be divided into two groups: communication with OBD-II devices and communication with nomadic devices. To reduce the risk introduced by OBD-II readers and adapters, it has been proposed to deactivate the interface while driving [153]. However, this solution reduces the use of applications that require run-time data, such as AP12 and AP13 in Table 5. Another proposal is to develop an upgraded OBD interface with a gatekeeper and a central gateway to in-vehicle networks [113]. Currently nomadic devices are used to send information to the car. Some protection mechanisms are device attestation (only smartphones from compliant manufacturers are allowed) and application and content type attestation (each application is assigned with a trust level that determines which information can be exchanged with the vehicle). Regarding nomadic devices, some mechanisms are device attestation (only smartphones from compliant manufacturers are allowed) and application and content type attestation (each application is assigned with a trust level that determines which content types it can use) [154]. However, these mechanisms alone are ineffective if the device is compromised [12]. A more promising solution is to treat nomadic devices as untrusted and validate all data exchanged between nomadic devices and in-vehicle systems. In addition, isolation techniques (discussed above) can be used to separate V2D applications from the rest of in-vehicle systems.

6.2. VANETs

In this section, we first study PKIs for VANETs, which is a standardized preventive solution. Next, we review some alternatives to PKIs. Finally, we examine solutions for detecting and responding to attacks.

6.2.1. PKIs

Several projects, government bodies, and standards [36, 73, 133, 155–157] have proposed the adoption of a PKI and digital signature schemes based on ECDSA[19] to ensure security and privacy in V2V and V2I communication. Accordingly, each vehicle should employ a private cryptographic key to sign messages. One private key can be associated with multiple short-term certificates, so called *pseudonyms*, which are issued by *pseudonym certificate authorities*. Pseudonyms can be used to verify messages signed with the private key; however, unlike certificates in PKIs employed in other domains, pseudonyms do not contain identifying information. As a result, message integrity can be ensured without revealing the identity of the vehicle. If legal investigation is required, authorities that have enough information (e.g., a database mapping issued pseudonyms to vehicle IDs or suitable cryptographic keys) can perform pseudonym-vehicle identity resolution. In addition, authorities should be able to revoke the certificates of misbehaving vehicles.

The use of digital signatures and certificates largely satisfies integrity, authentication, and non-repudiation requirements. This approach also ensures a degree of revocable privacy. When combined with message timestamps, it also ensures message freshness. However, digital signatures and certificates introduce computational overhead in the form of complex cryptographic operations and transmission overhead in the form of certificate transmission. There has been several proposals to alleviate these problems. The EVITA project presents a hardware security module (HSM) to accelerate cryptographic operations (and to securely store keys and generate random numbers) [36, 133]. Krishnan and Weimerskirch [163] propose to verify only relevant incoming messages. A disadvantage of this approach is that, it requires complex cross-layer design: the relevancy of a message is only known at the application level [192]. Various certificate omission schemes have been proposed to reduce transmission overhead [158–162]. In an omission scheme, the receivers cache incoming certificates and the sender omits certificates from selected messages. The messages can be verified if their certificates have been cached.

Last but not least, pseudonyms may be insufficient to prevent location tracking. An attacker could deduce the complete travel path by combining pseudonyms and location information [193]. To this end, various pseudonym changing strategies have been proposed. For example, vehicles can abstain from sending messages at random periods to ensure unlinkability between pseudonyms [164]. In particular, several vehicles can form a group such that only one group member broadcasts messages while the other members stay silent for a period to enhance location privacy. However, silent periods are unsuitable for periodically broadcast messages required by several safety applications. Another work proposes that vehicles trade their

[19] ECDSA is an algorithm for creating and verifying digital signatures based on Elliptic curve cryptography.

pseudonyms [165]. While this method improves privacy, the fact that vehicles can obtain pseudonyms through exchanges makes non-repudiation more difficult to achieve and opens opportunities for Sybil attacks.

6.2.2. Alternatives to PKI

Multiple alternatives and extensions to ECDSA-based digital signature and PKI have been proposed. One approach is to partially or completely replace asymmetric cryptography with symmetric cryptography. Symmetric cryptography typically has higher performance but requires more complex key management or additional assumptions, e.g., trusted hardware or the availability of RSUs. Zang et al. [194] propose an authentication scheme in which trusted and high-performance RSUs assist V2V communication. Studer et al. [167] propose a combination of the TESLA protocol[20] [195] and PKI to provide message authentication. The protocol provides authentication, multi-hop authentication, non-repudiation, and resistance against memory and computation-based DoS. Lin et al. [168] present another scheme based on TESLA and PKI. The scheme provides authentication, revocable privacy, and replay attack resistance. A downside of TESLA-based protocols is a high delay between message arrival and message authentication. All the schemes described above use a combination of symmetric cryptography and PKI and rely on the PKI as a fallback mechanism in case RSUs are unavailable or if the delay introduced by TESLA must be avoided. Paruchuri and Duressi [169] present a scheme based purely on symmetric cryptography. However, the scheme strongly depends on the tamper-proof property of smart cards, which is often hard to achieve.

Two other alternatives to ECDSA-based digital signature are group signature [170–173] and identity-based cryptography [175, 176]. In a group signature scheme, a group member can anonymously sign a message on behalf of the whole group. A group manager is responsible for adding new members to a group and revoking the anonymity of a signer in case of a dispute (these two functions can sometimes be performed by separate entities). Group signatures can ensure anonymity and jurisdictional access. In identity-based cryptography, a node's identity serves as its public key; the private key corresponding to an identity is issued by a trusted authority [196]. With identity-based cryptography, certificates are not needed because private keys are issued only to trusted nodes, thus simplifying key management. In [174], messages from vehicles (which should remain anonymous) are signed with group signatures. Messages from RSUs (whose identities can be revealed) are signed with ID-based signatures to simplify key management. The proposed schemes can potentially provide the desired security and privacy goals (integrity, data freshness, etc.). However, to date they have been evaluated only based on theoretical analysis or through the simulation of specific scenarios, whereas their real-world implementation and performance have not been sufficiently studied.

[20] TESLA is an authentication protocol using hash chains and time synchronization.

6.2.3. Detection of and response to misbehaving vehicles

Preventive mechanisms, e.g., digital signatures, can effectively stop external attackers from injecting bogus messages into communication channels. However, they do not address incorrect information coming from insiders. Such information could be the result of malfunctioning sensors or compromised ECUs. Addressing this issue requires evicting misbehaving nodes from the network. The eviction process can be divided into four main steps [197]: (1) *misbehavior detection*, (2) *misbehavior reporting*, (3) *certification revocation*, and (4) *revocation information dissemination*.[21] Next, we review existing solutions for each of these steps.

Misbehavior detection. Several techniques for detecting misbehaving nodes have been proposed, e.g. [177–182]. Misbehavior detection techniques can be classified based on two orthogonal aspects: *node-centric* vs. *data-centric* mechanisms and *autonomous* vs. *collaborative* mechanisms [183]. Data-centric mechanisms rely on message contents, while node-centric mechanisms are mainly concerned with the behaviors of network nodes, such as message frequency and message correctness. The correctness of messages needs to be verified by a separate method, which is usually data-centric. In an autonomous mechanism, a vehicle (detector) analyzes messages from a single vehicle and may include information from the detector's own sensors. On the other hand, collaborative mechanisms aim to identify misbehavior from multiple vehicles by gathering information from a wider set of sources; thus, they can potentially detect more attacks. However, they are more complex compared to autonomous mechanisms and typically assume that the majority of vehicles in the vicinity are honest.

Existing misbehavior detection systems for VANET [177–182] can also be classified into behavioral-based and signature-based as for in-vehicle network intrusion detection systems and share the same drawbacks (see Section 6.1). Additional challenges in the context of VANET are posed by the heterogeneity of driving behaviors under different traffic and road conditions, weather, driving habits, normal and extreme events like crashes, etc. To address these issues, Le et al. [184] propose a semantics-aware white-box anomaly detection approach for extracting understandable behavioral models in the context of V2V communication. Despite this work making a first step towards the definition of intrusion detection systems for VANET, many challenges remains open. As shown in [184], a main challenge is to identify the relevant features to distinguish normal driving patterns from attacks. In fact, certain behaviors could be anomalous under some conditions and normal under others.

Reporting and revocation decision. Vehicles should warn others or report to an authority about misbehaving nodes. On the other hand, reporting should not be abused to attack innocent vehicles. Raya et al. [178] propose a neighbor warning scheme called LEAVE. In this scheme, each vehicle broadcasts warning messages when it detects a misbehaving vehicle. Each vehicle

[21] Kherani and Rao [197] describes the steps in the context of PKIs. However, the steps are applicable to any scheme that supports revocation.

aggregates all received warnings into a single trust value. If this value exceeds a threshold, the vehicle broadcasts *disregard* messages instructing other vehicles to ignore messages from the suspected vehicle. To discourage false accusations, Moore et al. [185] propose a mechanism called *Stinger*, in which both reporting and reported vehicles are temporarily prohibited from sending messages. Both LEAVE and Stinger assume that the majority of the vehicles are honest, which may not be true in the case of Sybil attacks. In addition, Stinger penalizes vehicles that report others, which may have a negative impact on the detection of misbehaving nodes.

Apart from temporary excluding misbehaving vehicles at a local scale, another question is when a certificate authority should (permanently) revoke certificates of a vehicle. One approach is to combine observations from multiple vehicles. However, this approach always suffers from either false negatives or false positives even if honest vehicles never accuse other honest vehicles [198]. The main problem lies in the fact that (1) it is difficult for a remote authority to verify with certainty the honesty of a given vehicle and (2) a misbehaving vehicle can accuse other vehicles, misbehaving or honest. The main challenge is to determine an acceptable trade-off between false positive/negative rates, and how to achieve them.

Dissemination of revocation information. The main method to disseminate revocation information in VANETs is the use of a certificate revocation list (CRL), which lists the serial numbers of all revoked certificates. Because of the scale and dynamicity of VANETs, CRLs may need frequent updates and can become large. Raya et al. [186] propose two methods to overcome this problem. The first method uses a Bloom filter for lossy compression of CRLs. It allows a configurable false positive rate that can be lowered by increasing either CRL size or computational cost. The second method relies on tamper-proof in-vehicle devices to stop signing messages when requested [186]. This method, however, does not work if the devices have been compromised. Laberteaux et al. [187] propose a method in which vehicles participate in CRL distribution; source vehicles only sends part of the CRL that receiving vehicles do not have. Papadimitratos et al. [188] keep the size of CRLs small by including only regional revocation information. A vehicle operating outside of its home region needs to request foreign certificates from the local certificate authority. In addition, CRLs are divided and transmitted in verifiable pieces. Our analysis shows that each method provides a trade-off, for example between the CRL size and false positives [186], between RSU availability and V2V communication [187], and between CRL size and PKI complexity [188]. The suitability of a method depends on the given scenario and, in particular, on the underlying infrastructure constraints and to what extent false positives can be tolerated in the given scenario.

7. Gap Analysis and Roadmap for Future Work

Having described the security and privacy requirements and existing solutions, this section identifies important open challenges whose resolution will allow a significant step towards the development of secure and privacy-preserving automotive applications and platforms, and their adoption in real-life settings.

Data collection. Most automotive applications require data collected by sensors and ECUs. For example, an intersection collision warning application (AP15 in Table 5) typically needs information about the current location, speed, heading, and acceleration of the host and surrounding vehicles. However, a large body of research on V2V and V2I communication implicitly assumes that in-vehicle data has been collected and are ready to create messages (Section 6.2). Current solutions, such as OBD adapters or manufacturers' dedicated interfaces, introduce security and privacy risks (Section 5.4) in addition to barriers to fair competition. Therefore, the main challenge is to devise novel approaches and techniques to securely collect data from sensors and aggregate them in such a way that security and privacy goals (Section 5.2) are ensured. Privacy-aware data collection is of utmost importance for the adoption of VANET applications in countries with a stringent privacy legal framework. This is the case, for instance, for EU countries where the newly introduced General Data Protection Regulation (GDPR) imposes stringent requirements on the collection and processing of personal data.

OTA updates. Our analysis shows an increasing body of research focusing on enabling OTA updates in the last years (Section 6.1). OTA updates can bring great value in terms of low cost and customer convenience compared to the traditional over-the-wire updates. However, enabling secure and safe OTA updates is challenging. It must satisfies several constraints, including the long life cycle of vehicles, safe update conditions, and recovery from failed updates. In addition, it involves many components such as servers, wireless communication interfaces, gateways, protocols, HSM, etc. Even the more comprehensive schemes like the one presented in [134] do not provide the necessary infrastructure or do not meet all constraints. The challenge is to design OTA schemes that encompasses all necessary components while satisfying all demands of automotive systems.

Resilient in-vehicles network. As discussed in Sections 5.4 and 6.1, CAN has intrinsic weaknesses. Ethernet is a promising replacement for CAN. Although Ethernet has already been deployed in vehicles for diagnosis and as backbone for in-vehicle network [34], questions concerning its security and privacy still remain open: Can Ethernet solve the issues of CAN and FlexRay? Does it introduce new weaknesses? What are the impacts of Ethernet on applications, platforms, and other security mechanisms? Overall, we believe that more development and evaluation of alternatives to CAN technology (such as Ethernet) are needed before it can be replaced.

At the moment, CAN remains the most common technology for in-vehicle networks and will likely remain over the next decade [31]. Therefore, in the meantime, we must rely on security measures built on top of CAN, for example, gateway firewalls and intrusion detection systems. We envision that these measures should remain applicable even when CAN has been replaced, because replacing CAN will not likely solve all security and privacy issues.

Gateway firewall. Some applications (e.g., AP8, AP12–AP14 in Table 5) require cross-domain communication. This introduces the risk of unauthorized applications sending messages to safety-critical networks. This risk has been demonstrated in [6, 39], where CAN messages can be sent from compromised infotainment systems. An automotive firewall could prevent or, at least, limit these attacks. Nevertheless, as discussed in Section 6.1, building automotive gateway firewalls is challenging and more development and testing are needed.

Intrusion detection and response. Preventive mechanisms are not perfect, and we should prepare for the situation in which some attackers can bypass them. Detecting and responding to threats are therefore of utmost importance.

Among existing detection mechanisms, intrusion detection systems appear to be the most promising solution. Our analysis of the literature has revealed two main types of intrusion detection systems for vehicular systems: signature-based and behavioral-based. While providing detection at high accuracy, signature-based systems are not able to detect unknown attacks, making such systems unsuitable for the protection of in-vehicle systems and VANETs, where the exposure to new security threats is constantly increasing. On the other hand, behavioral-based intrusion detection systems can detect both known and unknown attacks but usually suffer a high false positive rate. A key challenge for the design of behavioral-based intrusion detection systems is the understanding of the normal communication patterns, which is complex for both in-vehicle networks and VANETs. Especially since protocols for in-vehicle networks, such as CAN, minimize the information exchanged in messages to reduce communication overhead. Also, in VANETs behaviour shows high heterogeneity and variability, which makes it difficult to distinguish attacks from extreme, but legitimate, driving situations.

A main desideratum of anomaly detection is that alerts are *actionable*. In particular, an intrusion detection engine should provide the information necessary to understand what caused an alert. This allows one to distinguish false positives (e.g., extreme events looking like attacks) from real attacks as well as to decide a proper response to the alert. White-box intrusion detection [199] can provide a viable solution for vehicular networks and recent work [184] has applied them in the context of VANET communication. Although the results are promising, more research efforts are required to build accurate behavioral models and, in particular, to select features suitable to capture normal communication patterns. We believe that this requires a multi-disciplinary approach that accounts for driving style, road type and weather conditions.

How to respond to the detected attacks is another critical challenge because the response can directly affect safety. Identifying the system components that exhibit symptoms of attacks (such as unresponsive brakes) is not sufficient to safely respond to attacks; response mechanisms should also identify and isolate the source of attacks (such as a telematics subsystem or a connected smartphone) and take a suitable response based on surrounding context. As a consequence, existing response mechanisms often require a large amount of information on the attacks (i.e., severity of attacks), the operational status of the vehicle, and the environment. However, as discussed earlier, data collection in vehicles is still an open problem. Research on graceful degradation [200], automatic collision avoidance, and other ADAS applications could provide insights into how to respond in dangerous situations. However, these research areas are mainly concerned with unintentional faults and accidents. An interesting direction is, thus, to investigate how research results in these areas can be adapted to deal with (intentional) security attacks.

PKI solutions for VANETs. The use of digital signatures and PKIs for VANETs has been widely researched and standardized. We believe that this solution is more mature compared to its alternatives. Nevertheless, there is a gap between existing academic research and large scale testing of vehicular PKI. We believe that further analysis and experiments are needed to discover and resolve potential issues which often occur in complex systems such as VANETs, including ambiguous specifications in standards, interoperability among equipment of different vendors, scalability, and the roles of different stakeholders in the PKI. In addition, there are still open questions regarding CRL distribution, pseudonym change strategies, misbehavior detection etc., which can hinder the transition of PKIs to practice. As discussed in the previous section, existing methods make a trade-off between different aspects (e.g., false positive rate, CRL size, PKI complexity). However, we observed a lack of evaluation and comparison among existing methods and their feasibility. Further research is needed to understand and evaluate such trade-offs within VANET scenarios.

Open multi-purpose application platform. Commercial in-vehicle platforms currently either support only one specific type of applications or are closed platforms (Section 4.2). In addition, they do not have built-in support for data collection and security mechanisms, such as gateway firewall, OTA updates, and intrusion detection. An open challenge is to build and deploy an application platform such that:

- It can support applications of heterogeneous characteristics, including different functional domains, safety levels, levels of driver involvement, connectivity, and time constraints (Section 3.1).
- It can support data collection to enable applications and security mechanisms such as intrusion detection and misbehavior detection for both in-vehicle networks and VANETs.
- It is open with respect to application distribution and platform development. This will promote transparency and encourage fair competition among application developers.
- It can enforce separation of concerns while allowing desirable interaction among applications. This requirement is fundamental to build a safe and secure platform.

8. Conclusion

This paper provides an analysis of the security and privacy challenges in the automotive sector with the goal of driving

research and development of security and privacy solutions suitable for intelligent connected vehicles. We find that, despite much work in the area, there are several open challenges.

By studying automotive applications and platforms and comparing their security and privacy requirements with the state of the art of related security and privacy solutions, we establish key gaps in areas such as: data collection, automotive Ethernet, gateway firewall, intrusion detection and response, and PKI for VANET. We believe these gaps hinder societal acceptance, and thus the deployment of connected vehicles at large scale.

Acknowledgment

This work is supported by Rijkswaterstaat under the TU/e Smart Mobility programme, by ITEA3 through the APPSTACLE project (15017), and by ECSEL through the SECREDAS project.

References

[1] J. Holle, A. Groll, C. Ruland, H. Cankaya, M. Wolf, Open platforms on the way to automotive practice, in: Proceedings of the 8th ITS European Congress, European Commission, 2011, p. 130.

[2] D. Gangadharan, O. Sokolsky, I. Lee, B. Kim, C.-W. Lin, S. Shiraishi, Platform-based automotive safety features, in: Proceedings of the SAE World Congress and Exhibition, SAE International, 2016. doi:10.4271/2016-01-0136.

[3] M. Broy, Challenges in automotive software engineering, in: Proceedings of the 28th International Conference on Software Engineering, ACM, 2006, pp. 33–42. doi:10.1145/1134285.1134292.

[4] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, et al., Experimental security analysis of a modern automobile, in: Proceedings of the IEEE Symposium on Security and Privacy, IEEE, 2010, pp. 447–462. doi:10.1109/SP.2010.34.

[5] I. D. Foster, A. Prudhomme, K. Koscher, S. Savage, Fast and vulnerable: A story of telematic failures, in: Proceedings of the 9th USENIX Workshop on Offensive Technologies, USENIX Association, 2015.

[6] C. Miller, C. Valasek, Remote exploitation of an unaltered passenger vehicle, accessed: 2018-07-18 (2015).

[7] I. Rouf, R. Miller, H. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, I. Seskar, Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study, in: Proceedings of the 19th USENIX Security Symposium, USENIX Association, 2010, pp. 323–338.

[8] T. Hoppe, S. Kiltz, J. Dittmann, Security threats to automotive CAN networks — practical examples and selected short-term countermeasures, Reliability Engineering & System Safety 96 (1) (2011) 11–25. doi:10.1016/j.ress.2010.06.026.

[9] A. Francillon, B. Danev, S. Capkun, Relay attacks on passive keyless entry and start systems in modern cars, Cryptology ePrint Archive, Report 2010/332 (2010).

[10] R. Verdult, F. D. Garcia, B. Ege, Dismantling Megamos Crypto: Wirelessly lockpicking a vehicle immobilizer, in: Supplement to the 22nd USENIX Security Symposium, USENIX Association, 2015, pp. 703–718.

[11] S. Indesteege, N. Keller, O. Dunkelman, E. Biham, B. Preneel, A practical attack on KeeLoq, in: Proceedings of the 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer Berlin Heidelberg, 2008, pp. 1–18. doi:10.1007/978-3-540-78967-3_1.

[12] S. Mazloom, M. Rezaeirad, A. Hunter, D. McCoy, A security analysis of an in vehicle infotainment and app platform, in: Proceedings of the 10th USENIX Workshop on Offensive Technologies, USENIX Association, 2016.

[13] A. Palanca, E. Evenchick, F. Maggi, S. Zanero, A stealth, selective, link-layer denial-of-service attack against automotive networks, in: Proceedings of the 14th Conference on Detection of Intrusions and Malware & Vulnerability Assessment, Springer International Publishing, 2017, pp. 185–206. doi:10.1007/978-3-319-60876-1_9.

[14] F. Simonot-Lion, Y. Trinquet, Vehicle functional domains and their requirements, in: Automotive Embedded Systems Handbook, CRC Press, 2008, pp. 3–24.

[15] R. Coppola, M. Morisio, Connected car: Technologies, issues, future trends, ACM Computing Surveys 49 (3) (2016) 46:1–46:36. doi:10.1145/2971482.

[16] P. Kleberger, T. Olovsson, E. Jonsson, Security aspects of the in-vehicle network in the connected car, in: Proceedings of the Intelligent Vehicles Symposium, IEEE, 2011, pp. 528–533. doi:10.1109/IVS.2011.5940525.

[17] B. Parno, A. Perrig, Challenges in securing vehicular networks, in: Proceedings of the Workshop on Hot Topics in Networks, 2005.

[18] M. Raya, J.-P. Hubaux, Securing vehicular ad hoc networks, Journal of Computer Security 15 (1) (2007) 39–68. doi:10.3233/JCS-2007-15103.

[19] S. Gillani, F. Shahzad, A. Qayyum, R. Mehmood, A survey on security in vehicular ad hoc networks, in: Proceedings of the 5th International Workshop on Communication Technologies for Vehicles, Springer Berlin Heidelberg, 2013, pp. 59–74. doi:10.1007/978-3-642-37974-1_5.

[20] M. A. Razzaque, A. S. S., S. M. Cheraghi, Security and privacy in vehicular ad-hoc networks: Survey and the road ahead, in: Wireless Networks and Security: Issues, Challenges and Research Trends, Springer Berlin Heidelberg, 2013, pp. 107–132. doi:10.1007/978-3-642-36169-2_4.

[21] J. T. Isaac, S. Zeadally, J. S. Camara, Security attacks and solutions for vehicular ad hoc networks, IET Communications 4 (7) (2010) 894–903. doi:10.1049/iet-com.2009.0191.

[22] L. B. Othmane, H. Weffers, M. M. Mohamad, M. Wolf, A survey of security and privacy in connected vehicles, in: Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications, Springer New York, 2015, pp. 217–247. doi:10.1007/978-1-4939-2468-4_10.

[23] Robert Bosch GmbH, Automotive networking, in: Bosch Automotive Electrics and Automotive Electronics: Systems and Components, Networking and Hybrid Drive, Springer Fachmedien Wiesbaden, 2014, pp. 82–91. doi:10.1007/978-3-658-01784-2_3.

[24] C. Valasek, C. Miller, A survey of remote automotive attack surfaces, in: Black Hat, 2014.

[25] T. Nolte, Share-driven scheduling of embedded networks, Ph.D. thesis, Institutionen för Datavetenskap och Elektronik (2006).

[26] T. Zhang, H. Antunes, S. Aggarwal, Defending connected vehicles against malware: Challenges and a solution framework, IEEE Internet of Things Journal 1 (1) (2014) 10–21. doi:10.1109/JIOT.2014.2302386.

[27] U. Keskin, In-vehicle communication networks : a literature survey, Computer science reports", Technische Universiteit Eindhoven, 2009.

[28] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, L. Kilmartin, Intra-vehicle networks: A review, IEEE Transactions on Intelligent Transportation Systems 16 (2) (2015) 534–545. doi:10.1109/TITS.2014.2320605.

[29] N. Navet, F. Simonot-Lion, A review of embedded automotive protocols, in: Automotive Embedded Systems Handbook, CRC Press, 2008, pp. 77–108.

[30] IEEE 802.3 Ethernet working group, 802.3 Ethernet, http://www.ieee802.org/3/, accessed: 2017-03-29.

[31] P. Hank, T. Suermann, S. Müller, Automotive Ethernet, a Holistic Approach for a Next Generation In-Vehicle Networking Standard, in: Advanced Microsystems for Automotive Applications, Springer Berlin Heidelberg, 2012, pp. 79–89. doi:10.1007/978-3-642-29673-4_8.

[32] J.-P. Gehrmann, G. Sporer, Higher bandwidth automotive-grade Ethernet and distributed vehicle networks, ATZelektronik worldwide 11 (6) (2016) 60–65. doi:10.1007/s38314-016-0085-8.

[33] L. L. Bello, The case for Ethernet in automotive communications, ACM SIGBED Review 8 (4) (2011) 7–15. doi:10.1145/2095256.2095257.

[34] T. Steinbach, K. Müller, F. Korf, R. Röllig, Demo: Real-time Ethernet in-car backbones: First insights into an automotive prototype, in: Proceedings of IEEE Vehicular Networking Conference, IEEE, 2014, pp.

133–134. doi:10.1109/VNC.2014.7013331.

[35] M. Wolf, A. Weimerskirch, C. Paar, Secure in-vehicle communication, in: Embedded Security in Cars: Securing Current and Future Automotive IT Applications, Springer Berlin Heidelberg, 2006, pp. 95–109. doi:10.1007/3-540-28428-1_6.

[36] J. P. Stotz, N. Bißmeyer, F. Kargl, S. Dietzel, P. Papadimitratos, C. Schleiffer, Security requirements of vehicle security architecture, Deliverable 1.1, PRESERVE Project (2011).

[37] European Union Agency for Network and Information Security (ENISA), Cyber security and resilience of smart cars - good practices and recommendations, https://www.enisa.europa.eu/publications/cyber-security-and-resilience-of-smart-cars/at_download/fullReport (2016).

[38] E. Kelling, M. Friedewald, T. Leimbach, M. Menzel, P. Säger, H. Seudié, B. Weyl, A. Fuchs, O. Henniger, M. S. Idrees, Specification and evaluation of e-security relevant use cases, Deliverable D2.1, EVITA Project (2009).

[39] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno, Comprehensive experimental analyses of automotive attack surfaces, in: Proceedings of the 20th USENIX Conference on Security, USENIX Association, 2011.

[40] K. McCord, Automotive Diagnostic Systems: Understanding OBD-I and OBD-II, CarTech Inc, 2011.

[41] HEM Data Corporation, What's in the CAN? OBD, J1939 and more, http://www.hemdata.com/products/whats-in-the-can, accessed: 2017-05-01.

[42] J. Holle, A. Groll, List of interfaces and specifications of information flow, Deliverable D2.1, OVERSEE Project (2011).

[43] TomTom Telematics, In-vehicle telematics devices, https://telematics.tomtom.com/en_us/webfleet/vehicle-telematics/usage-based-insurance/products/, accessed: 2017-04-05.

[44] Automatic Labs, Automatic: Connect your car to your digital life, https://www.automatic.com/, accessed: 2017-04-05.

[45] AT&T Business, Usage-based insurance and telematics, https://www.business.att.com/enterprise/Service/internet-of-things/vehicle-solutions/usage-based-insurance, accessed: 2017-04-05.

[46] FMS-Standard group, Information about the FMS-Standard, http://www.fms-standard.com, accessed: 2017-04-25.

[47] N. Lu, N. Cheng, N. Zhang, X. Shen, J. W. Mark, Connected vehicles: Solutions and challenges, IEEE Internet of Things Journal 1 (4) (2014) 289–299. doi:10.1109/JIOT.2014.2327587.

[48] Y.-D. Kim, I.-Y. Moon, ZigBee and IEEE 802.15.4 standards, in: ZigBee Network Protocols and Applications, Auerbach Publications, 2014, pp. 31–52.

[49] M. Weyn, G. Ergeerts, L. Wante, C. Vercauteren, P. Hellinckx, Survey of the DASH7 Alliance protocol for 433 MHz wireless sensor communication, International Journal of Distributed Sensor Networks 9 (12). doi:10.1155/2013/870430.

[50] L. Yang, G. B. Giannakis, Ultra-wideband communications: an idea whose time has come, IEEE Signal Processing Magazine 21 (6) (2004) 26–54. doi:10.1109/MSP.2004.1359140.

[51] J.-S. Lee, Y.-W. Su, C.-C. Shen, A comparative study of wireless protocols: Bluetooth, UWB, ZigBee, and Wi-Fi, in: Proceedings of the 33rd Annual Conference of IEEE Industrial Electronics, IEEE, 2007, pp. 46–51. doi:10.1109/IECON.2007.4460126.

[52] J. B. Kenney, Dedicated Short-Range Communications (DSRC) standards in the United States, Proceedings of the IEEE 99 (7) (2011) 1162–1182. doi:10.1109/JPROC.2011.2132790.

[53] European Telecommunications Standards Institute, ETSI EN 302 663 v1.2.1: Intelligent Transport Systems (ITS); Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band (2013).

[54] M. Picone, S. Busanelli, M. Amoretti, F. Zanichelli, G. Ferrari, Communication paradigms and literature analysis, in: Advanced Technologies for Intelligent Transportation Systems, Springer International Publishing, 2015, pp. 21–50. doi:10.1007/978-3-319-10668-7_2.

[55] R. El Hattachi, J. Erfanian, 5G white paper, Final Deliverable, NGMN 5G Initiative (2015).

[56] International Telecommunication Union, ITU-R M.2133 - Requirements, evaluation criteria and submission templates for the evelopment of IMT-Advanced (2008).

[57] International Telecommunication Union, ITU-R M.1822 - Framework for services supported by IMT (2007).

[58] B. Badic, C. Drewes, I. Karls, M. Mueck, Introduction to mobile wireless systems, in: Rolling Out 5G: Use Cases, Applications, and Technology Solutions, Apress, 2016, pp. 1–10.

[59] 3rd Generation Partnership Project, Study on new services and markets technology enablers v14.2.0, http://www.3gpp.org/ftp/Specs/archive/22_series/22.891/22891-e20.zip (2016).

[60] ITU Radiocommunication Sector, IMT vision - Framework and overall objectives of the future development of IMT for 2020 and beyond, https://www.itu.int/rec/R-REC-M.2083-0-201509-I/en (2015).

[61] D. Mohr, N. Müller, A. Krieg, P. Gao, H.-W. Kaas, A. Krieger, R. Hensley, The road to 2020 and beyond, https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-road-to-2020-and-beyond-whats-driving-the-global-automotive-industry, accessed: 2018-07-18 (2013).

[62] M. Alam, The software defined car: Convergence of automotive and Internet of Things, in: Wireless World in 2050 and Beyond: A Window into the Future!, Springer International Publishing, 2016, pp. 83–92. doi:10.1007/978-3-319-42141-4_8.

[63] R. Kroh, A. Kung, F. Kargl, VANETS security requirements final version, Deliverable D1.1, SEVECOM Project (2006).

[64] R. Kala, Advanced driver assistance systems, in: On-Road Intelligent Vehicles, Butterworth-Heinemann, 2016, pp. 59 – 82. doi:10.1016/B978-0-12-803729-4.00004-0.

[65] P. Knoll, Driving assistance systems, in: Brakes, Brake Control and Driver Assistance Systems: Function, Regulation and Components, Springer Fachmedien Wiesbaden, 2014, pp. 180–185. doi:10.1007/978-3-658-03978-3_15.

[66] F. Kargl, Z. Ma, E. Schoch, Security engineering for VANETs, in: Proceedings of the 4th Workshop on Embedded Security in Cars, 2006.

[67] D. Karapiperis, B. Birnbaum, A. Brandenburg, S. Castagna, A. Greenberg, R. Harbage, A. Obersteadt, Usage-based insurance and vehicle telematics: Insurance market and regulatory implications, CIPR Study Series 1 (2015) 1–79.

[68] European Telecommunications Standards Institute, ETSI TR 102 638 v1.1.1 - Intelligent Transport Systems (ITS); Vehicular communications; Basic set of applications; Definitions (2009).

[69] Digital Trends, Ford demonstrates smart home integration at CES 2016, http://www.digitaltrends.com/cars/ford-wants-future-cars-and-smart-homes-to-get-along/, accessed: 2017-04-05 (2016).

[70] F. Gil-Castiñeira, D. Chaves-Diéguez, F. J. González-Castaño, Integration of nomadic devices with automotive user interfaces, IEEE Transactions on Consumer Electronics 55 (1). doi:10.1109/TCE.2009.4814411.

[71] J. Piao, M. McDonald, Advanced driver assistance systems from autonomous to cooperative approach, Transport Reviews 28 (5) (2008) 659–684. doi:10.1080/01441640801987825.

[72] C. Grepet, Use case identification, Deliverable D1.1, OVERSEE Project (2011).

[73] J. Harding, G. Powell, R. Yoon, J. Fikentscher, C. Doyle, D. Sade, M. Lukuc, J. Simons, J. Wang, Vehicle-to-Vehicle communications: Readiness of V2V technology for application, Report DOT HS 812 014, National Highway Traffic Safety Administration - U.S. Department of Transportation (2014).

[74] Y. Dong, Z. Hu, K. Uchimura, N. Murayama, Driver inattention monitoring system for intelligent vehicles: A review, IEEE Transactions on Intelligent Transportation Systems 12 (2) (2011) 596–614. doi:10.1109/TITS.2010.2092770.

[75] J. Hernandez, D. McDuff, X. Benavides, J. Amores, P. Maes, R. Picard, AutoEmotive: Bringing empathy to the driving experience to manage stress, in: Proceedings of the Companion Publication on Designing Interactive Systems, ACM, 2014, pp. 53–56. doi:10.1145/2598784.2602780.

[76] K. Schreiner, Night vision: Infrared takes to the road, IEEE Computer Graphics and Applications 19 (5) (1999) 6–10. doi:10.1109/38.788791.

[77] ERIKA Enterprise, http://erika.tuxfamily.org/drupal/, accessed: 2017-04-05.

[78] Vector, osCAN, https://vector.com/vi_oscan_en.html, accessed: 2017-04-05.

[79] ARCCORE, Products, https://www.arccore.com/products, accessed: 2017-04-05.

[80] Vector, AUTOSAR - a choice for the future!, https://vector.com/vi_autosar_solutions_en.html, accessed: 2017-04-05.

[81] QNX Software Systems Limited, QNX CAR Platform for Infotainment, http://www.qnx.com/content/qnx/en/products/qnxcar/index.html, accessed: 04-10-2017.

[82] Microsoft, A technical companion to Windows Embedded Automotive 7, http://download.microsoft.com/download/0/A/1/0A1E07D6-7562-4566-AACF-E04DF4FF8879/ATechnicalCompaniontoWindowsEmbeddedAutomotive7(final).pdf, accessed: 2018-07-18 (2010).

[83] GENIVI Alliance, About GENIVI, https://www.genivi.org/about-genivi, accessed: 04-10-2017.

[84] The Linux Foundation, Automotive Grade Linux, https://www.automotivelinux.org/, accessed: 04-10-2017.

[85] Apertis, Apertis developer portal, https://wiki.apertis.org/Main_Page, accessed: 04-10-2017.

[86] AGA Project, Automotive Grade Android, https://developer.lindholmen.se/redmine/projects/aga/wiki, accessed: 04-10-2017.

[87] QNX Software Systems Limited, Advanced driver assistance systems, http://www.qnx.com/content/qnx/en/products/adas/index.html, accessed: 2017-04-17.

[88] Green Hills Software, Green Hills Platform for Advanced Driver Assistance Systems, http://www.ghs.com/products/auto_adas.html, accessed: 2017-04-16.

[89] Wind River Systems, Inc., Wind River Helix Drive, https://www.windriver.com/products/chassis/drive/, accessed: 2017-04-16.

[90] A. Groll, J. Holle, C. Ruland, M. Wolf, T. Wollinger, F. Zweers, OVERSEE - a secure and open communication and runtime platform for innovative automotive applications, in: Proceedings of the 7th Embedded Security in Cars Conference, 2009.

[91] Apple Inc., CarPlay, https://www.apple.com/ios/carplay/, accessed: 04-10-2017.

[92] Google Inc., Android Auto, https://www.android.com/auto/, accessed: 04-10-2017.

[93] Car Connectivity Consortium, MirrorLink, http://mirrorlink.com, accessed: 04-10-2017.

[94] Ford Motor Company, AppLink, https://developer.ford.com/pages/applink, accessed: 10-04-2017.

[95] ITEA3, APPSTACLE – open standard APplication Platform for carS and TrAnsportation vehiCLEs, https://itea3.org/project/appstacle.html, accessed: 2018-02-26.

[96] OSEK Group, OSEK/VDX Binding specification version 1.4.2 (2004).

[97] AUTOSAR, Classic Platform, https://www.autosar.org/standards/classic-platform/, accessed: 2017-04-05.

[98] AUTOSAR, Classic Platform release overview - release 4.3.0 (2016).

[99] AUTOSAR, Specification of operating system - release 4.3.0 (2016).

[100] AUTOSAR, Specification of communication - release 4.3.0 (2016).

[101] ETAS, RTA-OS, https://www.etas.com/en/products/rta_os.php, accessed: 2017-04-05.

[102] Mentor, AUTOSAR products, https://www.mentor.com/products/vnd/autosar-products/, accessed: 2017-04-05.

[103] OSEK Group, OSEK/VDX operating system - v2.2.3 (2005).

[104] AUTOSAR, Adaptive Platform release overview - release 17-03 (2017).

[105] A. Hergenhan, G. Heiser, Operating systems technology for converged ECUs, in: Proceedings of the 6th Embedded Security in Cars Conference, 2008.

[106] OpenSynergy GmbH, COQOS SDK, http://www.opensynergy.com/en/products/coqos/, accessed: 2017-04-23.

[107] S. Diewald, A. Möller, L. Roalter, M. Kranz, et al., Mobile device integration and interaction in the automotive domain, in: Proceedings of Automotive Natural User Interfaces Workshop at the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications, 2011, pp. 166–169.

[108] Ford Motor Company, API reference, https://developer.ford.com/pages/api-reference-android, accessed: 2017-04-19.

[109] Android Developers, Building Apps for Auto, https://developer.android.com/auto/index.html, accessed: 04-10-2017.

[110] Apple Inc., CarPlay for developers, https://developer.apple.com/carplay/, accessed: 04-10-2017.

[111] Apple Inc., MFi - frequently asked questions, https://mfi.apple.com/MFiWeb/getFAQ, accessed: 04-10-2017.

[112] E. J. Markey, Tracking & hacking: Security & privacy gaps put American drivers at risk, Congressional Reporthttp://www.markey.senate.gov/imo/media/doc/2015-02-06_MarkeyReport-Tracking_Hacking_CarSecurity2.pdf, accessed: 2018-07-18.

[113] European Commission, C-ITS platform - final report, https://ec.europa.eu/transport/sites/transport/files/themes/its/doc/c-its-platform-final-report-january-2016.pdf, accessed: 2018-07-18 (2016).

[114] R. Kissel, Glossary of key information security terms, National Institute of Standards and Technology (2013). doi:10.6028/NIST.IR.7298r2.

[115] G. Elahi, E. Yu, N. Zannone, A vulnerability-centric requirements engineering framework: Analyzing security attacks, countermeasures, and requirements based on vulnerabilities, Requirements Engineering 15 (1) (2010) 41–62. doi:10.1007/s00766-009-0090-z.

[116] A. Ruddle, D. Ward, B. Weyl, S. Idrees, Y. Roudier, M. Friedewald, T. Leimbach, A. Fuchs, S. Grgens, O. Henniger, Security requirements for automotive on-board networks based on dark-side scenarios, Deliverable D2.3, EVITA Project (2009).

[117] B. Guttman, E. A. Roback, An Introduction to Computer Security: The NIST Handbook, National Institute of Standards and Technology, 1995.

[118] European Telecommunications Standards Institute, ETSI TR 102 893 V1.2.1 - Intelligent Transport Systems (ITS); Security; vulnerability and risk analysis (TVRA) (2017).

[119] M. Wolf, Attackers and attacks in the automotive domain, in: Security Engineering for Vehicular IT Systems: Improving the Trustworthiness and Dependability of Automotive IT Applications, Vieweg + Teubner, 2009, pp. 77–89. doi:10.1007/978-3-8348-9581-3_5.

[120] P. Papadimitratos, V. Gligor, J.-P. Hubaux, Securing vehicular communications - assumptions, requirements, and principles, in: Proceedings of the Workshop on Embedded Security in Cars, 2006.

[121] WikiLeaks, Vault-7: CIA hacking tools revealed, https://wikileaks.org/ciav7p1/cms/page_13763790.html, accessed: 2017-05-15.

[122] B. J. Czerny, System security and system safety engineering: Differences and similarities and a system security engineering process based on the ISO 26262 process framework, SAE International Journal of Passenger Cars - Electronic and Electrical Systems 6 (1) (2013) 349–359. doi:10.4271/2013-01-1419.

[123] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, M. T. M. Shalmani, On the power of power analysis in the real world: A complete break of the KeeLoq code hopping scheme, in: Proceedings of the 28th Annual International Cryptology Conference, Springer Berlin Heidelberg, 2008, pp. 203–220. doi:10.1007/978-3-540-85174-5_12.

[124] D. K. Nilsson, U. E. Larson, F. Picasso, E. Jonsson, A first simulation of attacks in the automotive network communications protocol FlexRay, in: Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems, Springer Berlin Heidelberg, 2009, pp. 84–91. doi:10.1007/978-3-540-88181-0_11.

[125] M. Contag, G. Li, A. Pawlowski, F. Domke, K. Levchenko, T. Holz, S. Savage, How they did it: An analysis of emission defeat devices in modern automobiles, in: Proceedings of the IEEE Symposium on Security and Privacy, IEEE, 2017, pp. 231–250. doi:10.1109/SP.2017.66.

[126] T. Eisenbarth, T. Kasper, C. Paar, S. Indesteege, KeeLoq, in: Encyclopedia of Cryptography and Security, Springer US, 2011, pp. 671–673. doi:10.1007/978-1-4419-5906-5_587.

[127] D. Klinedinst, C. King, On board diagnostics: Risks and vulnerabilities of the connected vehicle, White Paper, Carnegie Mellon University (2016).

[128] A. Kovelman, A remote attack on the Bosch Drivelog connector dongle, Argus blog, Argus Cyber Security, https://argus-sec.com/remote-attack-bosch-drivelog-connector-dongle/ Accessed: 2018-07-21.

[129] I. A. Sumra, I. Ahmad, H. Hasbullah, J. lail bin Ab Manan, Classes of attacks in VANET, in: Proceedings of the Saudi International Electronics, Communications and Photonics Conference, IEEE, 2011, pp. 1–5. doi:10.1109/SIECPC.2011.5876939.

[130] M. S. Al-kahtani, Survey on security attacks in vehicular ad hoc networks (VANETs), in: Proceedings of the 6th International Conference on Signal Processing and Communication Systems, IEEE, 2012, pp. 1–9. doi:10.1109/ICSPCS.2012.6507953.

[131] J. R. Douceur, The Sybil Attack, in: Proceedings of International Workshop on Peer-to-Peer Systems, LNCS 2429, Springer, 2002, pp. 251–260. doi:10.1007/3-540-45748-8_24.

[132] D. K. Nilsson, U. Larson, A defense-in-depth approach to securing the wireless vehicle infrastructure, Journal of Networks 4 (2009) 552–564. doi:10.4304/jnw.4.7.552-564.

[133] L. Apvrille, R. El Khayari, O. Henniger, Y. Roudier, H. Schweppe, H. Seudié, B. Weyl, M. Wolf, Secure automotive on-board electronics network architecture, in: Proceedings of the FISITA World Automotive Congress, 2010.

[134] M. S. Idrees, H. Schweppe, Y. Roudier, M. Wolf, D. Scheuermann, O. Henniger, Secure automotive on-board protocols: A case of over-the-air firmware updates, in: Communication Technologies for Vehicles, Springer Berlin Heidelberg, 2011, pp. 224–238. doi:10.1007/978-3-642-19786-4_20.

[135] H. Mansor, K. Markantonakis, R. N. Akram, K. Mayes, Let's get mobile: Secure FOTA for automotive system, in: Proceedings of the 9th International Conference on Network and System Security, Springer International Publishing, 2015, pp. 503–510. doi:10.1007/978-3-319-25645-0_38.

[136] F. Stumpf, C. Meves, B. Weyl, M. Wolf, A security architecture for multipurpose ECUs in vehicles, in: Proceedings of the 25th Joint VDI/VW Automotive Security Conference, 2009.

[137] D. Adam, S. Tverdyshev, C. Rolfes, T. Sandmann, Two architecture approaches for MILS systems in mobility domains (automobile, railway and avionik), in: Proceedings of International Workshop on MILS: Architecture and Assurance for Secure System, EURO-MILS, 2015. doi:10.5281/zenodo.47991.

[138] B. Glas, J. Guajardo, H. Hacioglu, M. Ihle, K. Wehefritz, A. Yavuz, Signal-based automotive communication security and its interplay with safety requirements, in: Proceedings of the Embedded Security in Cars Conference, 2012.

[139] A. Van Herrewege, D. Singelee, I. Verbauwhede, CANAuth - a simple, backward compatible broadcast authentication protocol for CAN bus, in: Proceedings of the ECRYPT Workshop on Lightweight Cryptography, 2011.

[140] B. Groza, S. Murvay, A. Van Herrewege, I. Verbauwhede, LiBrA-CAN: a lightweight broadcast authentication protocol for controller area networks, in: Proceedings of the International Conference on Cryptology and Network Security, Springer, 2012, pp. 185–200. doi:10.1007/978-3-642-35404-5_15.

[141] A.-I. Radu, F. D. Garcia, LeiA: A lightweight authentication protocol for CAN, in: Proceedings of the 21st European Symposium on Research in Computer Security, Springer International Publishing, 2016, pp. 283–300. doi:10.1007/978-3-319-45741-3_15.

[142] Q. Zou, W. K. Chan, K. C. Gui, Q. Chen, K. Scheibert, L. Heidt, E. Seow, The study of secure CAN communication for automotive applications (2017). doi:10.4271/2017-01-1658.

[143] P. Mundhenk, A. Paverd, A. Mrowca, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, S. Chakraborty, Security in automotive networks: Lightweight authentication and authorization, ACM Transactions on Design Automation of Electronic Systems 22 (2) (2017) 25:1–25:27. doi:10.1145/2960407.

[144] K. Schmidt, H. Zweck, U. Dannebaum, Hardware and software constraints for automotive firewall systems?, in: Proceedings of SAE World Congress and Exhibition, SAE International, 2016. doi:10.4271/2016-01-0063.

[145] M. D. Pesé, K. Schmidt, H. Zweck, Hardware/software co-design of an automotive embedded firewall, in: Proceedings of SAE World Congress Experience, SAE International, 2017. doi:10.4271/2017-01-1659.

[146] V. Verendel, D. K. Nilsson, U. E. Larson, E. Jonsson, An approach to using honeypots in in-vehicle networks, in: Proceedings of the IEEE 68th Vehicular Technology Conference, IEEE, 2008, pp. 1–5. doi:10.1109/VETECF.2008.260.

[147] T. Hoppe, S. Kiltz, J. Dittmann, Applying intrusion detection to automotive it-early insights and remaining challenges, Journal of Information Assurance and Security 4 (6) (2009) 226–235.

[148] A. Taylor, N. Japkowicz, S. Leblanc, Frequency-based anomaly detection for the automotive CAN bus, in: Proceeding of the World Congress on Industrial Control Systems Security, IEEE, 2015, pp. 45–49. doi:10.1109/WCICSS.2015.7420322.

[149] H. M. Song, H. R. Kim, H. K. Kim, Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network, in: Proceedings of the International Conference on Information Networking, IEEE, 2016, pp. 63–68. doi:10.1109/ICOIN.2016.7427089.

[150] K.-T. Cho, K. G. Shin, Fingerprinting electronic control units for vehicle intrusion detection, in: Proceedings of the 25th USENIX Security Symposium, USENIX Association, 2016, pp. 911–927.

[151] M. Marchetti, D. Stabili, A. Guido, M. Colajanni, Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms, in: Proceedings of the 2nd IEEE International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow, IEEE, 2016, pp. 1–6. doi:10.1109/RTSI.2016.7740627.

[152] E. Caberto, S. Graham, A method of securing a vehicle's controller area network, in: Proceedings of the 12th International Conference on Cyber Warfare and Security, Academic Conferences and Publishing International Limited, 2017, pp. 461–468.

[153] C. Hammerschmidt, German car industry plans to close OBD interface, http://www.smart2zero.com/news/german-car-industry-plans-close-obd-interface, accessed: 2017-05-25.

[154] R. Bose, J. Brakensiek, K.-Y. Park, Terminal mode: Transforming mobile devices into automotive application platforms, in: Proceedings of the 2nd International Conference on Automotive User Interfaces and Interactive Vehicular Applications, ACM, 2010, pp. 148–155. doi:10.1145/1969773.1969801.

[155] IEEE Vehicular Technology Society, IEEE standard for wireless access in vehicular environments–security services for applications and management messages, IEEE Std 1609.2-2016, IEEE (2016). doi:10.1109/IEEESTD.2016.7426684.

[156] European Telecommunications Standards Institute, ETSI TS 102 940 V1.1.1 - Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management (2012).

[157] European Telecommunications Standards Institute, ETSI TS 102 941 V1.1.1 - Intelligent Transport Systems (ITS); Security; trust and privacy management (2012).

[158] F. Kargl, E. Schoch, B. Wiedersheim, T. Leinmüller, Secure and efficient beaconing for vehicular networks, in: Proceedings of the 5th ACM International Workshop on VehiculAr Inter-NETworking, ACM, 2008, pp. 82–83. doi:10.1145/1410043.1410060.

[159] E. Schoch, F. Kargl, On the efficiency of secure beaconing in VANETs, in: Proceedings of the 3rd ACM Conference on Wireless Network Security, ACM, 2010, pp. 111–116. doi:10.1145/1741866.1741885.

[160] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, A. Lioy, On the performance of secure vehicular communication systems, IEEE Transactions on Dependable and Secure Computing 8 (6) (2011) 898–912. doi:10.1109/TDSC.2010.58.

[161] M. Feiri, J. Petit, F. Kargl, Evaluation of congestion-based certificate omission in VANETs, in: Proceedings of the IEEE Vehicular Networking Conference, IEEE, 2012, pp. 101–108. doi:10.1109/VNC.2012.6407417.

[162] M. Feiri, R. Pielage, J. Petit, N. Zannone, F. Kargl, Pre-Distribution of Certificates for Pseudonymous Broadcast Authentication in VANET, in: Proceedings of the IEEE 81st Vehicular Technology Conference, IEEE, 2015, pp. 1–5. doi:10.1109/VTCSpring.2015.7146029.

[163] H. Krishnan, A. Weimerskirch, "Verify-on-demand" - a practical and scalable approach for broadcast authentication in vehicle-to-vehicle communication, SAE International Journal of Passenger Cars - Mechanical Systems 4 (1) (2011) 536–546. doi:10.4271/2011-01-0584.

[164] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, K. Sezaki, CARAVAN: Providing location privacy for VANET, in: Proceedings of the Embedded Security in Cars Conference, 2005.

[165] D. Eckhoff, C. Sommer, T. Gansen, R. German, F. Dressler, Strong and affordable location privacy in VANETs: Identity diffusion using time-slots and swapping, in: Proceedings of the IEEE Vehicular Networking Conference, IEEE, 2010, pp. 174–181. doi:10.1109/VNC.2010.5698239.

[166] Y. Zhang, S. Gjessing, H. Liu, H. Ning, L. Yang, M. Guizani, Securing vehicle-to-grid communications in the smart grid, IEEE Wireless Communications 20 (6) (2013) 66–73. doi:10.1109/MWC.2013.6704476.

[167] A. Studer, F. Bai, B. Bellur, A. Perrig, Flexible, extensible, and efficient VANET authentication, Journal of Communications and Networks 11 (6) (2009) 574–588. doi:10.1109/JCN.2009.6388411.

[168] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho, X. Shen, TSVC: Timed efficient and secure vehicular communications with privacy preserving, IEEE Transactions on Wireless Communications 7 (12) (2008) 4987–4998. doi:10.1109/T-WC.2008.070773.

[169] V. Paruchuri, A. Durresi, PAAVE: Protocol for anonymous authentication in vehicular networks using smart cards, in: Proceedings of the IEEE Global Telecommunications Conference, IEEE, 2010, pp. 1–5. doi:10.1109/GLOCOM.2010.5683087.

[170] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, A. Lioy, Efficient and robust pseudonymous authentication in VANET, in: Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks, ACM, 2007, pp. 19–28. doi:10.1145/1287748.1287752.

[171] J. Guo, J. P. Baugh, S. Wang, A group signature based secure and privacy-preserving vehicular communication framework, in: Proceedings of the Mobile Networking for Vehicular Environments Workshop, IEEE, 2007, pp. 103–108. doi:10.1109/MOVE.2007.4300813.

[172] B. K. Chaurasia, S. Verma, S. M. Bhasker, Message broadcast in VANETs using group signature, in: Proceedings of the 4th International Conference on Wireless Communication and Sensor Networks, IEEE, 2008, pp. 131–136. doi:10.1109/WCSN.2008.4772697.

[173] X. Zhu, S. Jiang, L. Wang, H. Li, W. Zhang, Z. Li, Privacy-preserving authentication based on group signature for VANETs, in: Proceedings of the IEEE Global Communications Conference, IEEE, 2013, pp. 4609–4614. doi:10.1109/GLOCOMW.2013.6855678.

[174] X. Lin, X. Sun, P.-H. Ho, X. Shen, GSIS: A secure and privacy-preserving protocol for vehicular communications, IEEE Transactions on Vehicular Technology 56 (6) (2007) 3442–3456. doi:10.1109/TVT.2007.906878.

[175] P. Kamat, A. Baliga, W. Trappe, An identity-based security framework for VANETs, in: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks, ACM, 2006, pp. 94–95. doi:10.1145/1161064.1161083.

[176] J. Sun, C. Zhang, Y. Fang, An ID-based framework achieving privacy and non-repudiation in vehicular ad hoc networks, in: Processings of IEEE Military Communications Conference, IEEE, 2007, pp. 1–7. doi:10.1109/MILCOM.2007.4454834.

[177] P. Golle, D. Greene, J. Staddon, Detecting and correcting malicious data in VANETs, in: Proceedings of the 1st ACM International Workshop on Vehicular Ad Hoc Networks, ACM, 2004, pp. 29–37. doi:10.1145/1023875.1023881.

[178] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, J.-P. Hubaux, Eviction of misbehaving and faulty nodes in vehicular networks, IEEE Journal on Selected Areas in Communications 25 (2007) 1557–1568. doi:10.1109/JSAC.2007.071006.

[179] T. Zhou, R. R. Choudhury, P. Ning, K. Chakrabarty, Privacy-preserving detection of Sybil attacks in vehicular ad hoc networks, in: Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Networking Services, IEEE, 2007, pp. 1–8. doi:10.1109/MOBIQ.2007.4451013.

[180] N.-W. Lo, H.-C. Tsai, Illusion attack on VANET applications - a message plausibility problem, in: Proceedings of Globecom Workshops, 2007, pp. 1–8. doi:10.1109/GLOCOMW.2007.4437823.

[181] S. Park, B. Aslam, D. Turgut, C. C. Zou, Defense against Sybil attack in vehicular ad hoc network based on roadside unit support, in: Proceedings of the IEEE Military Communications Conference, IEEE, 2009, pp. 1–7. doi:10.1109/MILCOM.2009.5379844.

[182] S. Ruj, M. A. Cavenaghi, Z. Huang, A. Nayak, I. Stojmenovic, On data-centric misbehavior detection in VANETs, in: Proceedings of the IEEE Vehicular Technology Conference, IEEE, 2011, pp. 1–5. doi:10.1109/VETECF.2011.6093096.

[183] R. W. van der Heijden, S. Dietzel, T. Leinmüller, F. Kargl, Survey on misbehavior detection in cooperative intelligent transportation systems, ArXiv e-prints abs/1610.06810 (2016).

[184] V. H. Le, J. den Hartog, N. Zannone, Feature Selection for Anomaly Detection in Vehicular Ad Hoc Networks, in: Proceedings of the 15th International Joint Conference on e-Business and Telecommunications, SciTePress, 2018, pp. 481–491. doi:10.5220/0006946804810491.

[185] T. Moore, M. Raya, J. Clulow, P. Papadimitratos, R. Anderson, J.-P. Hubaux, Fast exclusion of errant devices from vehicular networks, in: Proceedings of the 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, IEEE, 2008, pp. 135–143. doi:10.1109/SAHCN.2008.26.

[186] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, J.-P. Hubaux, Certificate revocation in vehicular networks (2006).

[187] K. P. Laberteaux, J. J. Haas, Y.-C. Hu, Security certificate revocation list distribution for VANET, in: Proceedings of the 5th ACM International Workshop on VehiculAr Inter-NETworking, ACM, 2008, pp. 88–89. doi:10.1145/1410043.1410063.

[188] P. P. Papadimitratos, G. Mezzour, J.-P. Hubaux, Certificate revocation list distribution in vehicular communication systems, in: Proceedings of the 5th ACM International Workshop on VehiculAr Inter-NETworking, ACM, 2008, pp. 86–87. doi:10.1145/1410043.1410062.

[189] M. Shavit, A. Gryc, R. Miucic, Firmware update over the air (FOTA) for automotive industry, in: Proceedings of the Asia Pacific Automotive Engineering Conference, SAE International, 2007. doi:10.4271/2007-01-3523.

[190] S. M. Bellovin, W. R. Cheswick, Network firewalls, IEEE communications magazine 32 (9) (1994) 50–57. doi:10.1109/35.312843.

[191] T. Chothia, F. D. Garcia, J. de Ruiter, J. van den Breekel, M. Thompson, Relay cost bounding for contactless emv payments, in: Financial Cryptography and Data Security, Springer Berlin Heidelberg, 2015, pp. 189–206. doi:10.1007/978-3-662-47854-7_11.

[192] F. Kargl, J. Petit, Security and privacy in vehicular networks, in: Vehicular Communications and Networks, Woodhead Publishing, 2015, pp. 171–190. doi:10.1016/B978-1-78242-211-2.00009-X.

[193] B. Wiedersheim, Z. Ma, F. Kargl, P. Papadimitratos, Privacy in inter-vehicular networks: Why simple pseudonym change is not enough, in: Proceedings of the 7th International Conference on Wireless On-demand Network Systems and Services, IEEE, 2010, pp. 176–183. doi:10.1109/WONS.2010.5437115.

[194] C. Zhang, X. Lin, R. Lu, P.-H. Ho, RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks, in: Proceedings of the IEEE International Conference on Communications, IEEE, 2008, pp. 1451–1457. doi:10.1109/ICC.2008.281.

[195] A. Perrig, J. D. Tygar, TESLA broadcast authentication, in: Secure Broadcast Communication: In Wired and Wireless Networks, Springer US, 2003, pp. 29–53. doi:10.1007/978-1-4615-0229-6_3.

[196] A. Shamir, Identity-based cryptosystems and signature schemes, in: Proceedings of the International Cryptology Conference, Springer Berlin Heidelberg, 1985, pp. 47–53. doi:10.1007/3-540-39568-7_5.

[197] A. Kherani, A. Rao, Performance of node-eviction schemes in vehicular networks, IEEE Transactions on Vehicular Technology 59 (2) (2010) 550–558. doi:10.1109/TVT.2009.2030136.

[198] B. Liu, J. T. Chiang, Y.-C. Hu, Limits on revocation in VANETs, in: Pre-Proceedings of the 8th International Conference on Applied Cryptography and Network Security (ACNS 2010 industry track), 2010.

[199] E. Costante, J. den Hartog, M. Petković, S. Etalle, M. Pechenizkiy, A white-box anomaly-based framework for database leakage detection, Journal of Information Security and Applications 32 (2017) 27–46. doi:10.1016/j.jisa.2016.10.001.

[200] W. Nace, P. Koopman, A product family approach to graceful degradation, in: Proceedings of the IFIP WG10.3/WG10.4/WG10.5 International Workshop on Distributed and Parallel Embedded Systems, Springer US, 2001, pp. 131–140. doi:10.1007/978-0-387-35409-5_13.