© Copyright Notice

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other non-commercial uses permitted by copyright law.

Security and Privacy in Vehicular Communications: Challenges and Opportunities

Cesar Bernardini[a], Muhammad Rizwan Asghar[b], Bruno Crispo[c,d]

[a]*Department of Computer Science, Aalto University, Finland*
[b]*Department of Computer Science, The University of Auckland, New Zealand*
[c]*Department of Computer Science, KU Leuven, Belgium*
[d]*Department of Information Engineering and Computer Science, University of Trento, Italy*

Abstract

Modern cars have become quite complex and heavily connected. Today, diverse services offer infotainment services, electric power-assisted steering, assisted driving, automated toll payment and traffic-sharing information. Thanks to recent technologies, which made it possible to enable these services. Unfortunately, these technologies also enlarge the attack surface. This survey covers the main security and privacy issues and reviews recent research on these issues. It summarizes requirements of modern cars and classifies threats and solutions based on the underlying technologies. To the best of our knowledge, this is the first survey offering such an overall view.

Keywords: Modern cars, Infotainment, EV, AUTOSAR, On-Board Diagnostic, Intra-vehicle communication, In-vehicle communication, Inter-vehicle communication, Privacy, Security

1. Introduction

The increasing use of electronic components and technology in cars has radically changed the mean of transportation in the last couple of decades. Modern cars became very complex and sophisticated systems after having been equipped with several on-board microcontrollers. The basic unit of computation is the Electronic Control Unit (ECU). Most ECUs are organized and interconnected to monitor and control different subsystems such as the interaction between the braking system, airbags and the Antilock Braking System (ABS). However, the interconnection is not limited to internal car's ECUs; ECUs interconnect to other vehicles' ECUs and microcontrollers available in the roads to provide active safety, infotainment services, electric power-assisted steering, airbags, antilock braking system, assisted driving, cruise controls, automated toll payment and to simplify interaction with the environment.

To provide such a wide range of services, modern cars enable several means of communication. First, different ECUs and subsystems are interconnected to form *intra-vehicle networks*. These subsystems may use different networking technologies depending on the diverse constraints. For instance, a mirror adjusting system has a lower priority compared to the system controlling the braking system. Second, modern cars provide different *gateways* to interact with the external entities. Gateways represent the entry and exit points of the intra-vehicle network and provide a wide range of services. These services include support for self-diagnostics, toll and petrol payment, remote access to various features of the car, the Bluetooth hands-free and the USB media player are few important ones among many others. A communication port, On-Board Diagnostics 2 (OBD2), exchanges test information with intra-vehicle ECUs and reports their status. The petrol and electrical stations could exchange payment information with the car [1]. Third, cars enhance car safety by means of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication that we will call *inter-vehicle* communication.

With such a broader range of communication, it is essential to understand the security and privacy aspects of such a complex Cyber-Physical System (CPS). Indeed, Miller [2] has already demonstrated examples of attacks in modern

Email addresses: mesarpe@gmail.com (Cesar Bernardini), r.asghar@auckland.ac.nz (Muhammad Rizwan Asghar), bruno.crispo@cs.kuleuven.be (Bruno Crispo)

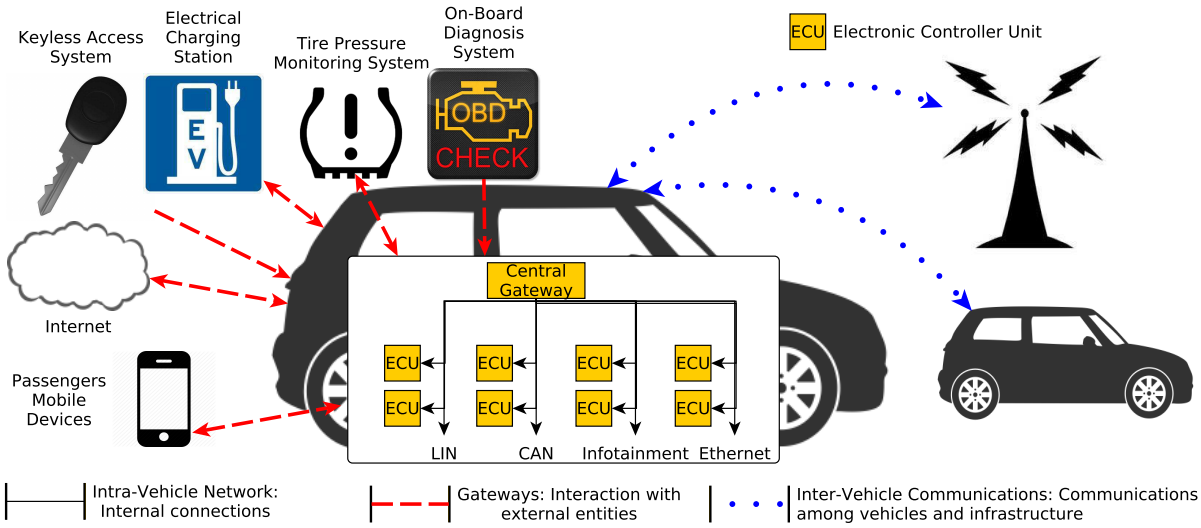

Figure 1: Overview of networking in modern cars, illustrating our proposed classification: interconnection of components in a modern car; inter-vehicle communications; and communication with external entities through gateways.

cars. Petit and Shladover [3] reviewed potential attacks that autonomous vehicles may suffer from. Kleberger *et al.* [4] revisited security aspects of the intra-vehicle network. Nevertheless, existing surveys neither comprehend essential aspects of modern cars nor address potential attacks and proposed solutions in a holistic fashion. This survey reviews the recent technical research on security and privacy in modern cars. It revisits threats and solutions based on the interconnection technologies.

Figure 1 illustrates the classification presented in this survey. A modern car is represented with its intra-vehicle network. The network is composed of multiple ECUs that are interconnected using different communication buses (*i.e.,* LIN, CAN, infotainment – MOST and FlexRay), represented using solid black arrows in Figure 1. The interaction of the car through gateways is represented using dashed red arrows. A modern car may interact with the Electric Charging Stations, the Keyless Access System, the Tire Pressure Monitoring System and the On-Board Diagnosis system. Further, modern cars can also directly connect to the Internet, cloud services and smartphones. The V2V and V2I communications are represented using dotted blue arrows.

The rest of the survey is organized as follows. First, we list security, safety, standardization and architectural requirements for modern cars in Section 2. Section 3 describes security and privacy issues that emerge in intra-vehicle networks. In Sections 4 and 5, we explain challenges caused by the use of gateways and inter-vehicle communication. Section 6 concludes this survey and provides research directions for future work.

2. Requirements for Modern Cars

From the security and privacy perspective, modern cars must respect a set of requirements. In this section, we list this set of requirements that we classify into three categories: security requirements, safety requirements, and standardization and architectural requirements.

2.1. Security Requirements

Security requirements are constraints that emerge from security concerns. Some of these requirements are based on use cases proposed in literature [5, 6]. In the following, we list core security requirements:

- *Authentication.* In vehicular networks, various entities (*e.g.,* drivers, cars and different service providers) interact with each other. If we consider interactions within the car, software and hardware components from different Original Equipment Manufacturers (OEMs) communicate with each other. This openness provides an opportunity to adversaries who may disrupt normal behavior of the car or trigger sophisticated attacks. To overcome this problem, communicating entities must authenticate each other to make sure that they are the ones they claim to be. Further, different car components must establish authenticity of incoming data and their origins.

- *Intellectual Property Protection.* Modern cars must be protected against cloning and reverse engineering. Their software and configuration parameters must remain protected. Moreover, they must be protected even during software updates and upgrades.

- *Confidentiality.* In vehicular networks, confidentiality refers to the protection of information exchanged between or stored by different components and entities. Loss of such protection could lead to leaking sensitive information.

- *Integrity.* In the context of vehicular networks, integrity refers to the property that if the data exchanged between or stored by components and entities is manipulated by an adversary, it must be detected. It must also be possible to detect injection of messages by an adversary. Moreover, it must be possible to detect any modification while the system boots or a piece of software is running. This equally applies to software upgrades and updates.

- *Access Control.* Modern cars must grant selective access to its components. A car must ensure that only authorized components are able to gain access. These components must only be repaired by authorized maintenance partners. Moreover, each component (or subcomponent) must be able to access memory allocated to it. In other words, secure software code must be written to avoid vulnerabilities, such as preventing buffer overflow or string format vulnerability attacks. Besides, entities must get access if they are authorized. The principle of least privilege should be applied while providing access to components or entities.

- *Message Freshness.* Modern cars must ensure that network messages are not being replayed or delayed. Assuring freshness avoids replay attacks, which otherwise could be mounted by adversaries.

- *Privacy.* Modern cars must protect sensitive information that may compromise privacy of users (say drivers or passengers). Tracking the geographical position of the car is a typical example of sensitive information. In this regard, the numerous location services, requiring position of the car, conflict with users' privacy. On top, multiple malicious service providers should not be able to learn more information about users by colluding together.

- *Availability.* A modern car must remain fully responsive at any point in time. Specifically, its components must be functional at all times. In terms of availability, it must be possible to prioritise tasks performed by components (or entities).

2.2. Safety Requirements

Safety requirements in modern cars are of utmost importance. The ISO 26262 standard [7] defines the functional safety requirements that must apply during the complete lifecycle of every automotive electronic/electrical system that is safety-related. In the following, we list the most important safety requirements:

- *Safe Development.* Modern cars must be developed following functional safety aspects including the specification of requirements, design, implementation and integration steps. Development of modern cars must be subject to validation and verification processes. One of the safety concerns is related to car components developed by third parties. To ensure safety, every third party component must be developed following ISO standards. Checkoway *et al.* [8] discovered that most of the security attacks found in automotive emerged from integrated components developed by organizations, other than the car manufacturer. To reduce such an attack surface, software and hardware components from different OEMs must be integrated in a secure manner. The integration must not only address functionalities and flow control but also protect intellectual property rights by carefully analyzing potential consequences.

- *Safety Risks.* The ISO 26262 standard [7] provides the Automotive Safety Integrity Level (ASIL), a risk-based approach to determine potential hazard in a vehicle operating scenario. There exist four levels of ASIL: ASIL-A, ASIL-B, ASIL-C and ASIL-D. The risk increases from ASIL-A to ASIL-D, where the ASIL-A level indicates the lowest risk while the ASIL-D level is the highest one. Besides, there is also one more level: the Quality Management (QM) level that means there are no hazards or safety risks. Every car component is subject to be classified with one of the risk levels, depending on the potential risk that it may imply.

- *Real-Time Constraints.* Modern cars must respect strong real-time requirements. The components must be designed for dealing with real-time operations. Especially, for completing safety-critical tasks, components must react precisely and accurately accordingly to hard deadlines. Indeed, a difference of milliseconds may be crucial to prevent or cause an accident. In [9], Karagiannis *et al.* summarize timing requirements for every component of the car.

- *Maintenance.* During the lifecycle of a modern car, it must be possible to fix potential issues in any of its components. There must exist an infrastructure to diagnose and successively repair any issues with the car. For instance, in case a hardware component fails, it must be possible to replace it. The same applies to software components, where updates and upgrades may be provided during the complete lifecycle of the car.

- *Free from Interference.* During the integration of third party components, it is necessary to ensure that components do not interfere with each other. The interference between software components does not end just with integration. In general, components may disrupt the behavior of other components. Modern cars include isolation mechanisms to prevent failing components interfering with other ones. Interference must be detected and measures must be taken to secure the system. The complete system can be turned into a safe state, or attempt to reboot it or its components. Interference can occur in three domains:

  - *Time Domain.* A non-critical task may be occupying resources while a critical task needs to be executed. Or a schedule may be wrongly assigning priorities to the tasks, which may lead to less important tasks monopolizing the resources. The freedom from interference in the time domain results in preventing all the time-related problems. It is commonly achieved by the use of hardware and software watchdogs.
  - *Communication Domain.* It refers to the networking aspects of the interference. Certain applications may overload the resources from the network, and in case it happens a mechanism must prevent it.
  - *Data Processing Domain.* It occurs when faulty software interferes with the processing of safety-related functions. Indeed, algorithms of high complexity and threads problems are two examples of monopolizing the resources. The solutions depend on the correct software engineering processes.

2.3. Standardization and Architectural Requirements

There are a large number of standards and guidelines modern cars should comply with. Here, we focus only the most important ones related to the adoption of digital communication and processing technologies. Introduced in 2003, a well-known guideline is AUTOSAR (AUTomotive Open System ARchitecture)[1], developed by a strong consortium of key players in the automotive industry including BMW, Bosch, DaimlerChrysler, Volkswagen, Ford, Peugeot and Toyota. AUTOSAR introduces a software architecture for automotive. This architecture aims to assist with the development of vehicular software, user interfaces and their management. AUTOSAR also provides guidelines for updates and upgrades of software and hardware components. There are already different releases of AUTOSAR in the market. As of March 2016, the latest AUTOSAR release is 4.2.2.

AUTOSAR plays a pivotal role in designing modern cars. The AUTOSAR architecture is based on layers. Every AUTOSAR layer has many components and modules. Every component has an associated function and provides services. Every component may use services from other layers or components. The components may be shared by modules. In the following, we list the core modules of the architecture:

[1] `http://www.autosar.org/`

- *Hardware Dependent Modules.* These modules are located in the lower layer of the architecture. They have a direct interaction with the ECUs. The aim of this layer is similar to that of the Hardware Abstraction Layer (HAL) of many operating systems.

- *Operating System (OS).* The OS module manages ECUs and software resources. Its main function is to provide common services for all the applications.

- *Basic Software (BSW).* The BSW performs the role of a middleware between the OS, applications and ECUs. The middleware is in charge of managing communication services, diagnostic capabilities, memory access, triggers, and watchdogs. Communication services are the drivers that interact with network buses. Diagnostic capabilities include software updates and controlling status of electronic components. Memory access deals with managing memory of the software stack. Triggers and watchdogs are responsible for detecting and recovering from malfunctioning of the OS and BSW. The Runtime Environment (RTE) module, explained below, is also considered under the umbrella of BSW.

- *Runtime Environment (RTE).* RTE is a virtual machine process for automotive applications. RTE is responsible for executing automotive applications. It serves as a middleware between applications and BSW. It is responsible for debugging applications, tracking errors and isolating applications from the rest of the system.

- *Software Components (SWC).* We can find a variety of applications for modern cars. These applications are commonly referred to as SWC. SWC is the most important structural element of the AUTOSAR architecture. SWC has ports to interact with the external world.

The AUTOSAR architecture also integrates safety features, some of which are listed below [10]:

- *Memory Protection.* The memory may be subject to corruption of contents or reading and writing by unauthorized modules. To protect the memory, AUTOSAR regulates access to the memory and uses the memory-mapped hardware.

- *Timing Monitoring.* Timing is an important property of intra-vehicular networks. Safety requirements demand that system actions are performed within the pre-defined deadline. AUTOSAR modules cannot guarantee that timing constraints will be enforced. Instead, the AUTOSAR architecture provides support to assess timing constraints for every AUTOSAR module. These modules are prone to execution blockage, deadlocks, incorrect allocation of execution time, and synchronization issues between components. The AUTOSAR architecture provides two timing monitoring mechanisms. First, the OS demands different timing bounds including maximum execution time, maximum locking time and minimum and maximum inter-arrival rate of tasks. The second mechanism is a temporal program flow monitoring that uses the watchdog manager. Basically, the watchdog manager checks if the AUTOSAR modules respect the execution time. In case they do not fulfil the requirements, the watchdog manager triggers alerts.

- *Logic Monitoring.* As AUTOSAR modules are based on software, their correct behavior must be assessed constantly. AUTOSAR includes logical supervision for checking the correct execution of software with a particular focus on control flow errors. An incorrect control flow occurs when program instructions are either processed in the incorrect order or not processed at all. The logical supervision of AUTOSAR includes creation of checkpoints in the source code. These checkpoints are then linked with transitions and an execution graph is created. The execution graph is verified at runtime. When a problem is detected, several recovery mechanisms are triggered. To fix the issue, the watchdog may shut down a memory partition or reboot a faulty ECU. In the worst scenario, the AUTOSAR architecture may reboot an integral part of the system.

- *End-to-End Communication.* As AUTOSAR is based on a modularized architecture, every AUTOSAR module communicates over a transmission channel. These channels must be protected against faults that may incur in individual modules or the overall system. In the following, we identify some potential faults in the communication channel: processes may produce some repetition, loss or corruption of the transmitted information; processes may block access to a communication channel; and processes may inject fault in a channel. To tackle potential

faults, AUTOSAR suggests several protection mechanisms including: Code Redundancy Checks (CRC) to verify the integrity of the messages; sequence counter to organize the processing of messages; timeout detection to assess freshness of elements and prevent outdated messages to flood the AUTOSAR architecture; and the assignment of IDs to every port number used by every process to track message provenance.

- *Execution Modes.* In AUTOSAR, each module runs in one of two modes: trusted group and untrusted group. The trusted group permits the module to have full access to the system. In this group, we can find the OS, BSW, RTE and Hardware Dependent Modules. The untrusted group is used for applications that do not require full access to the system.

From release 3.0, the AUTOSAR architecture suggests the incorporation of the following security modules and services:

- *Crypto Service Manager (CSM).* CSM offers a standardized access to cryptographic services for BSW and applications [11]. Cryptographic services support basic cryptographic primitives (*i.e.,* hash, Message Authentication Code – MAC in short, random number generation, encryption and decryption modules and signatures), key management (*i.e.,* key derivation function, key generation update, export and update) and other services (such as compression and checksum). To provide such services, AUTOSAR defines an internal interface to cryptographic algorithms called *Crypto Library Module (CRY)*.

- *Crypto Abstraction Layer (CAL).* CAL provides other software modules with cryptographic services [12]. CAL offers a C programming interface that can be called directly from BSW modules or complex drivers. As CAL is a library, its services are executed as normal stateless functions.

- *Secure On-Board Communication (SecOC).* SecOC provides secure communication in AUTOSAR [13]. AUTOSAR suggests that SecOC must have minimal impact on resource consumption in order to allow a better protection of vehicles. SecOC provides authentication and resistance against replay attacks [14]. Keyed-Hash Message Authentication Codes (HMACs) are used to protect integrity and authenticity. Confidentiality is achieved by employing encryption, using symmetric key cryptography.

3. Intra-Vehicle Communications

In this section, we first describe components of the intra-vehicle network and then we present the related security and privacy challenges. Note that the intra-vehicle network is formed by the interconnection of the multiple ECUs in a car. The intra-vehicle network is composed of the following components:

- **Electronic Control Unit (ECU).** ECU is the unit of computation in the intra-vehicular network.

- **Communication Media.** The intra-vehicle network is composed of physical wires that interconnect ECUs. These wires interconnect distinct ECUs, forming multiple networks that are linked through a *Central Gateway*. All these networks are operated using different data communication bus-based networks.

As shown in Table 1, the intra-vehicle network is managed using different bus-based networks. In the table, we have categorized them based on the data transfer rate, applications, bus features, nature of real-time and access control. It is important to mention that the nature of real-time could be either hard or soft, which depends on whether deadlines are completely or partially met, respectively. In the following, we briefly describe some of these bus-based networks:

- **Controller Area Network (CAN).** CAN is a data bus standard designed to allow microcontrollers and devices to communicate with each other in applications without requiring a host machine [15, 16]. It is a serial communication protocol. It supports distributed soft real-time control. In CAN, access control to the physical media is regulated using CSMA/CA [17]. A CAN bus allows adding nodes to the network in a plug-and-play manner [18]. An improvement in the initial version of CAN was proposed by Bosch in 2012, which is called CAN with Flexible Data Rate (CAN-FD). Among other changes, CAN-FD improves the data transfer rate, increases the payload from 8 to 64 bytes and includes integrity checks based on CRC codes [19].

Table 1: Summary of characteristics and use of different data communication buses in modern cars.

Bus Name	Transfer Rate	Used for	Bus Features	Hard/Soft Real-Time	Access Control
CAN	1Mbit/s	OBD2, power-train, chassis and body electronics	Multi-master serial bus and Low-cost protocol	Soft real-time	CSMA/CA
CAN-FD	8Mbit/s		Similar to CAN with longer payload		
LIN	20Kbit/s	Body electronics (including mirrors, power seats and accessories)	Broadcast serial bus, master-slave communication and cheaper than CAN	Hard real-time	Polling
FlexRay	10Mbit/s	High-performance power-train and safety (drive by wire, active suspension and adaptive cruise control)	Multi-master serial bus, 1-master; up to 16 slaves, expensive protocol and 2 channels	Hard real-time	TDMA
MOST	150Mbit/s	Rear-view, cameras, infotainment and multi-master bus	Ring topology, supports 64 devices and very high cost	Hard real-time	CSMA/CA TDM
Ethernet (BroadR -Reach)	100Mbit/s	Cameras, infotainment and on-board diagnosis	Cheaper than MOST, more expensive than CAN and lightweight wiring and CSMA/CD	Soft real-time	TDMA TDD

- **Local Interconnect Network (LIN).** LIN is a serial network protocol based on a master-slave model that offers a transfer rate up to 20kb/s. The master node can control up to 16 slave nodes [20, 21]. LIN requires a pre-defined schedule, which specifies at which time each node is allowed to send messages [18].
- **FlexRay.** It is considered as the successor of CAN [22]. It offers a transfer rate up to 10Mb/s. It manages the access to physical media with deterministic time-triggered TDMA behavior [21]. It provides constant latency and jitter using clock synchronization. FlexRay nodes are more expensive than CAN nodes.
- **Media Oriented Systems Transport (MOST).** MOST is a synchronous network protocol used to carry multi-media data via optical fiber. It offers a transfer rate up to 150Mb/s. In MOST, access control is managed with CSMA/CA and TDM. Its main drawback is its very high cost.
- **Ethernet and BroadR-Reach.** Ethernet is a family of computer network protocols used in local and media area networks. It provides a data transfer rate that can reach 100Gb/s. Ethernet is efficient in terms of bandwidth and interoperability. An adaptation of Ethernet, BroadR-Reach, was proposed to replace all the intra-vehicle network protocols [23]. The arguments in favor of BroadR-Reach are: the wiring weight of the car is reduced up to 30%, plenty of technology can be immediately transferred from the telecommunications industry and Ethernet is a scalable and robust protocol with mature solutions [24]. Ethernet provides real-time features for timing and synchronization while bridging network protocols [25]. The Ethernet protocol is also evolving to interrupt existing packets and guarantee latency requirements of the vehicle industry [26].

There exist several network protocols with diverse characteristics. However, due to reduced costs and maturity of the Ethernet technology, we believe that Ethernet will be a key technology in future. With proven success in the aircraft and telecommunications industry and several standards for bridging networks, Ethernet is a mature network protocol likely to replace all the others. The BMW X5 is already using BroadR-Reach and most likely many others will follow[2].

Security and Privacy Issues

The intra-vehicle network provides very attractive features that come with research challenges. In this section, we describe security and privacy challenges and status of state-of-the-art research on this topic.

[2] `http://opensig.org/about/market-forecast-gartner/`

Authentication of ECUs. To prevent impersonation attacks, ECUs have a unique identifier that can be used to certify their identity to other ECUs and sensors. The AUTOSAR architecture includes authentication capabilities: every partner uses a Cipher-based Message Authentication Code (CMAC). The CMAC codes use the AES cipher, which requires that both communication parties share the same key. The AUTOSAR architecture requires that these keys are pre-shared before any communication. Also, other security modules, such as [27], assume pre-sharing of keys. Mundhenk *et al.* [28] present a lightweight authentication framework that does not need the initial sharing of keys. Like SSL/TLS, symmetric keys are negotiated using digital certificates containing asymmetric keys. Afterwards, the communication is protected using negotiated keys. Once a digital certificate is generated, it is broadcasted to the network.

Time-Propagation Errors. As modern cars must respect real-time constraints, the AUTOSAR architecture defines a fixed priority pre-emptive schedule. This schedule specifies a fixed priority rank to every operation. Tasks with higher priority are executed before low priority tasks. The priority rank is determined based on the worst case of the operation. When an operation surpasses its estimated period of time, the task is removed from the schedule and the next task is executed. Safety-related tasks may be re-scheduled when they cannot finish on time. This situation may cause that low-priority is never executed and then components may start failing. Errors may be propagated from high priority to low priority tasks triggering cascade errors on the vehicle.

The AUTOSAR architecture avoids these problems by using watchdogs. Every watchdog monitors a task in the intra-vehicle network. Monitoring every task may actually boost rather than limit the propagation of errors due to two reasons. First, an extra overhead is produced by analyzing the less critical tasks. Second, the time consumed by every task is calculated with worst case estimation. The time may be over-approximated and have an impact on the overall system. In [29], Piper *et al.* propose to limit the monitoring process to only the critical tasks. Then, every unused computation time can be used for non-critical tasks.

Network Monitoring. Miller and Valasek [2] showed that an attacker could take full control of the cars remotely. They hacked the car while a journalist was driving it[3]. Note that such attacks could last for several minutes, say 15 minutes or more. With the terrific impact that an attack may have on a car, it is essential to monitor the network and to detect as soon as possible potential attacks against the car. For this purpose, Intrusion Detection Systems (IDSs) could be used. IDSs are software applications commonly used to monitor a network and aim at detecting malicious activities. For the CAN bus, Matsumoto *et al.* [30] present a solution that continuously monitors messages sent through the network and detects potentially unauthorized messages. Their approach does not consider real-time constraints of the network protocol. Broster and Burns [31] consider a bus guardian architecture that not only detects unauthorized messages but also limits the effect of these messages on the real-time constraints. However, both approaches are strongly coupled with the CAN network and cannot be applied to other buses. In [32], the authors propose a central gateway that controls and logs behavior in intra-vehicle networks. By analyzing error frames in network buses, the central gateway may take pre-emptive measures such as reporting unauthorized messages. The approach requires that CAN buses are adapted to generate error frames, the authors present an example for the CAN bus called CaCAN.

Network monitoring also deals with allowing or denying data traffic in the intra-vehicle network. [33] highlighted the importance of protecting the intra-vehicle network with firewalls. As communication parties are known in advance, firewall rules may be based on the authorization signature. Firewalls must completely disable access to gateways, such as OBD2, during normal operations. However, to defend the network against attacks, firewalls are commonly installed with honeypots. A honeypot is a security mechanism designed to detect or deflect attacks on information systems. Honeypots are usually placed in an easy-to-access position. They are intentionally made vulnerable to certain attacks so that attackers can be tempted to target them thus the intrusion can be detected and monitored. As honeypots are constantly monitored, the behavior of attackers is logged for *a posteriori* analysis. In [34], Verendel *et al.* presented three intra-vehicular designs for honeypots. Their honeypots simulate the full functionality of ECUs and are designed to copy traffic from the real intra-vehicle network to simulate realistic traffic. Although the research has highlighted the importance of honeypots, modern cars are not using them. We believe that one of the potential reasons for not using a honeypot could be its high cost.

Modern cars require the interaction of multiple components produced by third parties. As modern cars are regulated under the restrictive legislation, OEMs are responsible not only for software components they produce but also

[3] `http://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/`

how they use and exchange data. Indeed, OEMs must defend themselves against malfunctioning. To this end, modern cars must include mechanisms to report data provenance. [35] and [36] propose analyzing information flow to track sensitive data. Information flow deals with labeling sensitive information to discover data provenance and leakage at runtime or performing a posteriori analysis. In [35], Schweppe *et al.* propose to instrument application binaries in the intra-vehicle network. The aim is to follow the flow of data between registers and memory regions, and to track potentially suspicious use of data. Instead of instrumenting application binaries, Bouard *et al.* [36] label every network message with the application ID. Messages are appended with signatures and tags to determine provenance and to simplify the filtering process.

Self-Healing. Modern cars must be able to resist attacks and systematic failures of software and hardware components. In every situation, the car must maintain safety functionalities to avoid accidents. To address the problem, AUTOSAR proposes that every vehicle may switch between three states, normal, warning, and safe. The normal state is enabled by default when no failures are detected. The next state is the warning state in which the car activates all the redundancy and monitoring measures to analyze the problem. Finally, when there is no solution, the car switches to the safe state. In this state, only minimal features are available.

A car must switch its state and reconfiguration must be performed on-the-fly, no matter whether it is moving or not. Klopper *et al.* [37] propose a scheduling algorithm that switches states and assures that all the relevant tasks are executed. The authors in [38] extend the idea to multi-core systems. Their proposal focuses on distributing the load across several multi-core systems. Both [37] and [38] focus on functionalities of individual ECUs and do not consider the status of the network. Unfortunately, the network may suffer from congestion problems during updates. For instance, in [39], the authors propose to distribute the tasks around the network by using load balancing.

Self-adaptive Network. Intra-vehicular networks are expected to support self-configuration and to adapt to changing environments. It could happen in special weather conditions such as rainy roads or other situations [40]. One of the issues in literature is the sequence of reconfiguration activities needed to maintain requirements during the update process [37]. [37] proposed a planning model to generate plans to adapt to different scenarios. They demonstrate that using their plans, an intra-vehicle network is shown to self-heal in particular scenarios. In [39], Iannich *et al.* present an alternative approach based on load balancing. They distribute the tasks around the network whiling detect errors and handle the migration of tasks. In [38], Biedermann *et al.* demonstrate how to provide self-healing capabilities in a multi-core system like the AUTOSAR architecture. The self-healing capabilities may permit an automobile to recover from the effects of a faulty-component and internal hijacked data. With the emergence of Software-Defined Networking (SDN), networks adaptation can take place programmatically. The network is subdivided in two: the data plane and the control plane. The data plane delivers content while the control plane manages the network and its configuration. SDN enables the use of security solutions through remote control, giving the possibility to disable features when required. We foresee some innovative applications of SDN in modern cars.

Secure Communication. As every protocol bus provides distinct features (see Table 1), protocols are susceptible to alternative attacks. In the CAN bus, messages are broadcasted to the network and received by all ECUs. Each message has an associated priority decided by the ECU. A high priority message is processed before a low priority message. It means that continuously sending a set of high priority messages may impact performance of ECUs. Constantly postponing low-priority messages however can result in some failure, which may cause an accident. To minimize potential risks, [41] proposed to isolate ECUs with the creation of several CAN buses. There are some other proposals that focus on improving the built-in security of the protocol. For instance, CANAuth provides authentication and resistance against replay attacks [14]. HMACs are used to protect integrity and authenticity while confidentiality is achieved by employing symmetric encryption.However, CANAuth requires 15 bytes of bandwidth overhead, which is quite a problem with the original CAN network. vatiCAN [42] is another solution that allows message recipients to verify message authenticity using HMACs. vatiCAN does not require any change in the CAN legacy. AUTOSAR SecOC was proposed as an authentication protocol for CAN-FD [43]. The mechanism is similar to CANAuth, but it exploits the larger payload of CAN-FD. In [44], Schweppe *et al.* propose an improvement to the approach where Message Authentication Codes (MAC) are used and confidentiality is achieved using symmetric encryption. Hartkopp *et al.* [45] proposed MaCAN, which improved speed and reduced the overhead by truncating MAC. Their approach can not only authenticate messages but also generate group authentication with the use of only 32 bits.

Counterfeiting and Intellectual Property Theft. Counterfeiting and intellectual property theft are two major problems that become important with the underlying software in modern cars. Counterfeits are fake replicas of real products. In the context of vehicular networks, counterfeits describe imitations of original components of the

car. Counterfeits may be installed on a car in order to reduce costs or due to unavailability of original components. Installing counterfeited ECUs may have a critical impact on the car [46]. As counterfeited ECUs have physical access to the network, they may potentially launch attacks against the rest of the system, disrupt behavior of the car or leak sensitive information to the external world. To some extent, the intellectual property theft is also related to counterfeiting. Modern ECUs rely on software to accomplish various tasks. The software can be stolen and potentially used in counterfeited ECUs [47]. As such, to protect against counterfeits and thefts, there are certain security challenges that need to be addressed. First, deployment of unclonable ECUs by protecting its source code and electronic logic. Second, secure identification of original and counterfeited products.

For the deployment of unclonable ECUs, Malipatlolla and Huss [48] suggest to divide software programs into two parts: a section with proprietary information and another without sensitive information. The first part contains all the sensitive information and is protected with the Public Key Infrastructure (PKI). The second part is not encrypted as it does not feature any proprietary information.

In regards to the identification of counterfeiting, Wasicek [49] proposes to continuously monitor presence of ECUs. When the automotive network is installed, every component sends a heartbeat signal to a security module. The security module is aware of identity of all the components and can detect once a component is missing or altered. IBM has started to dig into Blockchains. By using a BlockChains, the car manufacturer and the car's owner can trace the provenance of parts. Detecting therefore all the way through the supply chain to the original manufacturer date and its location.[4]

4. In-Vehicle Network Gateways

Modern cars make use of many services that require interactions with external entities and devices. These external entities and devices include the on-board diagnosis, tire pressure systems, electrical charging/payment systems, Bluetooth devices, USB keys, CD-ROM and keyless system. These external entities and devices require the exchange of data to unlock the car and release payment information to accept charging electricity.

In this section, we first describe the technology used by different gateways in vehicular networks. We then analyze security and privacy challenges faced by in-vehicle gateways (*i.e.,* the keyless entry system, electrical cars and infotainment systems). Table 2 summarizes gateways available in the intra-vehicle network along with their input and output technologies.

Table 2: Gateways available in the intra-vehicle network: input and output technologies.

Gateway	**Input Media/Port**	**Connection to (Network)**
On-Board Diagnostics	OBD2	Any OBD2 device, embedded modem and CAN network
Tire Pressure Monitoring System	Wireless Radio frequency	ECUs
Electric Vehicles	CHAdeMO and IEC 61851	SmartGrid, ZigBee, Ethernet and CAN
Keyless System	Wireless radio	ECUs
Infotainment	Bluetooth, WiFi, USB and radio frequency	Clouds, smartphones, 4G, MOST and Ethernet
Telematics	Wireless radio frequency	4G, GPS, GLONASS, Galileo, DAB, SDARS and RDS

4.1. In-Vehicle Gateways

In this section, we cover gateways used for in-vehicle communication. These gateways include on-board diagnostics, tire pressure monitoring system, electric vehicles, keyless system, infotainment and telematics. In the following, we provide a brief description of each gateway.

[4] https://www.ibm.com/blogs/internet-of-things/iot-blockchain-automotive-industry/

4.1.1. On-Board Diagnostics (OBD)

Modern cars depend on ECUs for providing a set of services. ECUs may be affected by environmental conditions such as overheating, corrosion and electrical problems. Its underlying software may be subject to faults and bugs. As a consequence, it is essential to verify the correct functioning of ECUs and the network. Existing vehicles are equipped with a self-diagnostics system. ISO 14229 [50] and ISO 15765 [51] are two standards that specify the communication protocol to evaluate the status of the ECUs. These standards are two options, among many others, for the Unified Diagnostic Service (UDS). Every ECU has an assigned address used in making queries. ECUs provide an interface for reading the Data Trouble Code (DTC), which identifies a particular problem and is intended to help the technician to find a fault within the car. ECUs must also allow reading and writing of configuration parameters and re-programming of the firmware. Both [50] and [51] specify only the communication protocol. Configuration parameters and DTC are decided by every car manufacturer. Most of the car manufacturers are using the CAN bus, except for those manufacturers that choose Ethernet for using high-speed communications in the upgrading process.

On-Board Diagnostics 2 (OBD2) is a communication port from which fault codes can be read. The OBD2 port allows not only reading and writing data but also installing software on ECUs. The OBD2 communication port permits direct access to the intra-vehicle network. Following regulations, every European car produced after 2008 must have an OBD2 port. There is at least a dozen of OBD2 standards. The ISO 15331 standard [52] specifies the connector OBD2, a minimal communication protocol, configuration parameters and DTC for every vehicle.

In the last few years, remote diagnosis has received great attention. Remote diagnosis is used to control status of ECUs remotely over the Internet. The remote diagnosis device is connected to the OBD2 port and has an embedded modem with access to the Internet. The device acts as a gateway that bridges both networks. The Internet communication protocol is specified in the ISO 13400 standard [53]. The remote diagnosis equipment can be a radio, a telematic box or third party devices (*i.e.,* smartphones).

There are smartphones applications such as Torque Pro[5] that can read fault codes in our network by only using OBD2 technology through wireless technology (Bluetooth or WiFi).

4.1.2. Tire Pressure Monitoring System (TPMS)

TPMS refers to a control system that reports current pressure of car tires. Since 2008, all new cars sold in the United States and in Europe are equipped with TPMS. TPMS works with a pressure sensor located inside the tire or connected directly to the tire valve. The pressure sensor sends data to another ECU located in the intra-vehicular network using a radio frequency transmitter.

4.1.3. Electrical Vehicles (EVs)

EVs are already in the market. Due to their reduced CO_2 emission, EVs are becoming more and more attractive. For charging, there already exists an electricity charger for EVs called the Electric Vehicle Supply Equipment (EVSE), which charges battery of the car [54]. The communication between the EV and the EVSE is handled differently, depending on the country. For instance, ISO/IEC 61851 and 15118 have been adopted in Europe and the United States; while, Japan follows CHAdeMO [18]. SAE has been working with the ZigBee Alliance to plug in vehicles using the Zigbee communication protocol [55]. ZigBee is a wireless protocol for wireless sensor networks.

There are different protocols to connect a car to the electricity network [56, 54]. The Open Charge Point Protocol (OCPP) standardizes the communication between the charge spot and the entity that manages the charge spot. The Open Smart Charging Protocol (OSCP) enables a Distribution System Operator (DSO) to limit the energy available for charging stations. OSCP aims at avoiding congestion problems in the electricity network.

In Figure 2, we summarize the charging infrastructure to charge EVs. An EV is connected to the charging spot using an electrical wire. The communication between the EV and the EVSE follows ISO 15118, ISO 62196 or CHAdeMO. The EVSE communicates with the Charge Spot Operator (CSO) using OCPP. CSO is connected to DSO that handles the distribution of energy.

[5] https://play.google.com/store/apps/details?id=org.prowl.torquefree&hl=en

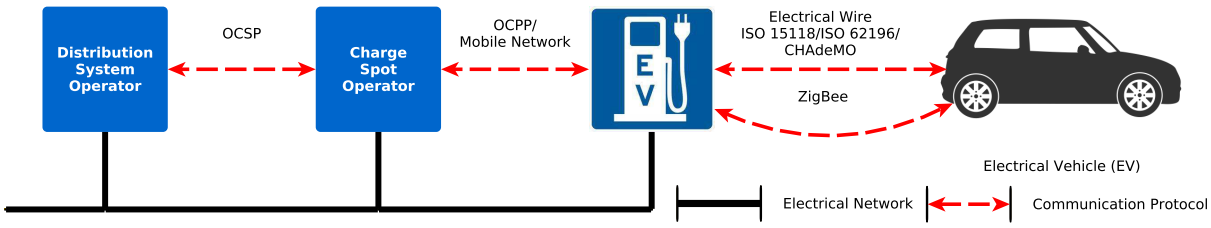

Figure 2: Infrastructure to charge an EV.

4.1.4. Remote Keyless System (RKS)

In old cars, physical keys were used to access the car. That is, by inserting the right key in the car lock, the driver used to access and drive the car. Starting from the 1980s, car manufacturers gradually replaced old physical keys with remote access keys also known as RKS. RKS refers to a lock that employs an electronic remote control to control access to the car[6].

There exist two types of RKS: active and passive. With an active RKS, users push a button on the keypad to unlock the door. With a passive RKS, the key unlocks the door when it approaches the car, so it does not need to be removed from a pocket or a bag. Further, infotainment systems, such as GM OnStar, can unlock the car with a remote signal sent through a smartphone application.

4.1.5. Infotainment and Telematics

Table 3: Typical infotainment systems and their services.

Infotainment System	**Connected to**			**Car Remote Control**	**Apps**
	Cloud	**4G**	**HotSpot WiFi**		
GM OnStar	✓	✓	✓	✓	–
BMW Connected Drive	✓	✓	✓	✓	✓
R-Link	✓	✓	✓	✗	✓
UVO	✓	✓	✓	✗	–
Infinity Connection	✓	✓	✓	✗	✗
MyFordTouch	✓	✓	✓	✓	✗
Citroen Multicity Connect	✓	✓	✓	✓	✓
Cadillac CUE	✓	✓	–	✓	✓
Drive U Connect	✓	✓	–	✓	✓
Mirroring					
Android Auto	✓	✓	✓	✓	Google Apps
Apple CarPlay	✓	✓	✓	✓	✓
Dashlinq	–	✓	✓	–	✓
SYNC AppLink	–	–	–	✓	✓
MirrorLink	✓	✓	✓	–	✓

Modern cars are equipped with multimedia systems that deliver entertainment, information, and emergency services. These systems are commonly classified as infotainment and telematics systems. Telematics systems use long-range mobile networks or Global Navigation Satellite System (GNSS). The long-range mobile networks include 2G, 3G, EDGE+, UTMS and 4G technologies. Examples of GNSS include the American Global Positioning System

[6] http://www.maximintegrated.com/en/app-notes/index.mvp/id/3395

(GPS), the Russian Global Orbiting Navigation Satellite System (GLONASS) and the European Galileo. Telematics systems are devices connected to the Internet or an external network via an integrated modem. Usually, they hold smaller displays and offer fewer services than infotainment systems.

Telematics services are diverse and mostly refer to the protection of drivers and of the car. The emergency call is a service that intends to bring rapid assistance to motorists involved in a collision. The European initiative for emergency calls is eCall while the Russian has Era-Glonass and Americans E911. Some insurance companies calculate the insurance policy based on vehicle information[7].

A telematics box installed in the car keeps track of the time and the vehicle position. The information is transferred to the insurance company that analyzes the data and calculate fees for the insurance policy. A telematics vehicle tracking system is used for the stolen vehicle tracking and remote monitoring.

Infotainment systems are basically media devices that provide useful information and entertainment services. The basic services provided by infotainment systems include, but are not limited to, radio streaming, pairing smartphones through Bluetooth and WiFi, CD-ROM and USB player. In modern cars, it is also possible to watch videos through infotainment systems and interact with smartphone applications. There exist many OSs for infotainment systems in the market, such as QNX (a Linux-based OS for embedded devices) and Windows Embedded Automotive. The Android OS is used as an application container for running vehicle applications. Renault R-Link is a typical example of infotainment based on Android OS.

Advanced infotainment systems are using a new technology called mirroring. Using mirroring, information systems can share the screen with smartphones or vice versa. The mirroring-enabled applications, such as CarPlay and MirrorLink are commonly certified by concerned authorities such as Apple and the Car Connectivity Consortium, respectively. Among others, Android Auto, Apple CarPlay, Dashlinq, Abalta Web Link and MirrorLink are some systems that provide this technology. In Table 3, we summarize typical infotainment systems and their services. In the table, we consider connectivity features that infotainment systems support. Most of the cars offer connectivity to the cloud and HotSpot services. On one hand, it is an opportunity; on the other hand, it can compromise security of the infotainment systems. This is the aspect that we will analyze in the rest of this section.

4.2. Security and Privacy Issues

In this section, we list all the security and privacy issues related to gateways in intra-vehicle networks. These issues are organized based on technologies.

4.2.1. On-Board Diagnostics

ECUs are dependent on software to control critical parts of the car. The ECU software is responsible for various tasks, such as managing the braking system or warning the driver about the low pressure on the tires. Indeed, software bugs may result in accidents and have fatal consequences for the driver and passengers of the car. Car manufacturers are already dealing with this problem and they have the infrastructure to patch the ECU software. Many vehicles have been recalled in the previous years due to software problems. For example, Ford has recalled 700,000 vehicles due to software problems on its airbag system [8]. To minimize the risk of accidents caused by software bugs, the software may be updated regularly. In the following, we highlight what security challenges the OBD technology raise and discuss available solutions:

- **Authentication.** Access keys are held by the devices to grant authorization to monitor and update the ECU software. However, these keys are susceptible to compromise. As such, authorization may be granted only to OEMs and trusted parties. The solution used in the automobile industry is the use of Hardware Security Modules (HSMs) [57]. HSMs include dedicated and a protected memory, a hardware accelerator for cryptographic algorithms and a secure storage for keys [58].

  Kanuparthi *et al.* [59] propose to authenticate parties using Physical Unclonable Function (PUF) sensors attached to ECUs. PUF sensors are electronic devices that provide security features to integrated circuits and

[7] https://www.progressive.com/auto/snapshot/

[8] http://spectrum.ieee.org/cars-that-think/transportation/safety/ford-recalls-695-000-vehicles-for-airbag-transmission-software-updates

have the advantage that the OEM cannot alter PUF. PUF sensors provide authentication using challenges. For a given challenge, a PUF sensor always creates the same unique answer that cannot be guessed or computed by other PUF sensors or other devices. However, PUF sensors do not allow to add new entities and communication parties must be known in advance. Clearly, this is a problem because the garage shops do not remain the same during the lifecycle of a car, some garages appear while others close. Kleberger *et al.* [60] propose an authentication scheme where three parties interact: the diagnostics equipment, the vehicle and a trusted third party. There exists a pre-sharing of keys between the vehicle and the Certification Authority (CA). The trusted third party acts as a CA.

- **Integrity.** The ODB port provides diagnostic codes to detect faults in ECUs. They report internal integrity and defects. However, these codes are not enough to determine if the car is safe and secure after an update. Automotive industry protects integrity of the installed software using digital signatures. Basically, the installed software carries a digital signature that is verified after installation. A periodic check of the signature determines if the software has been modified [58].

  In [61], Drolia *et al.* propose an automotive testbed – AutoPlug – to verify integrity of ECUs. AutoPlug is connected to a car simulator, which provides input for evaluating integrity and performance of ECUs. Drolia *et al.* explain how the ECU responsible for the power-train is evaluated and verified. Instead of using a simulator, Lee *et al.* [62] suggest the inclusion of a master ECU that is in charge of controlling integrity of the vehicle software. ECUs in the vehicle interact with the master ECU by following a challenge-response mechanism. The vehicle creates challenges not only for authenticating the communication parties but also to assess status of the devices.

- **Secure Communication.** For reducing costs, updates are provided using the Firmware Over The Air (FOTA) technology. Shavit, Gryc and Miucic [63] suggest that FOTA does not require any additional infrastructure because it is already used for software updates in the telecommunications industry.

  ISO13400-1 [64] defines a standard vehicle interface for upgrading the software and controlling car emissions. The ISO 13400 standard proposes to use existing communication technology, to separate the technology from the test equipment and the vehicle interface and to adapt the communication protocol to new communication protocols. The network layer services are provided using IP. The standard is also known as the Diagnostics over Internet Protocol (DoIP).

  The work in [65] focuses on problems for the secure communication using DoIP. DoIP's standard does not specify any secure communication protocol. The standard simply suggests that solutions for securing communication, such as SSL/TLS or IPSec, must be considered while implementing DoIP.

- **Attacks on Third Party Interfaces.** The ODB2 port allows third parties to develop software that interacts with the vehicle. As ODB2 is a standard port, third parties may build a device that connects to the vehicle. Unfortunately, such a device becomes an attractive target for attacks. ARGUS Cyber Security[9] explains a remote attack on a third party device sold by Zubie. Zubie's devices allow users to track their driving habits, detect malfunctioning in the vehicle and share locations with their friends. The device includes a debugging port based on UART and a GPRS modem that connects the device to the Internet. By connecting through the UART port, the attackers discovered that the device downloads the software from the Internet based on certain configuration parameters. The attack exploits the DNS address spoofing to download executable code from a malicious server. Once DNS is changed and the software is downloaded, the device executes the python code without checking any signature.

4.2.2. Tire Pressure Monitoring System

In the following, we highlight security and privacy challenges TPMS raises and discuss possible solutions:

- **Authentication and Confidentiality.** Using TPMS, each tire gets an ID. Every 6 minutes, TPMS sensors contact the intra-vehicle network to communicate the pressure of tires. Roufa *et al.* [66] reported serious

[9] `http://argus-sec.com/blog/remote-attack-aftermarket-telematics-service/`

security concerns in TPMS: messages are exchanged without authentication and no input validation takes place. Thus, the authors eavesdrop the TPMS communication and then trigger messages that pop-up warning alerts in a moving vehicle. Xu *et al.* [67] proposed a session-based lightweight encryption scheme. When a car is ignited, a new session begins. At the same time, TPMS sends its ID protected using symmetric encryption. The ID is generated every time by hashing an Initialization Vector (IV) with the sensor ID and a random number. All the communication is encrypted with a master key between every tire and the vehicle.

- **Tracking Vehicles.** The TPMS technology uses identification keys to recognize tires of the car. Every tire communicates with an ECU located within the intra-vehicle network, every 6 minutes its ID and current tire pressure are communicated. As every message includes the tire ID and tires are kept in the car for long periods of time, TPMS messages are easily traceable and may be used to identify the car. Roufa *et al.* [66] demonstrate that a vehicle can be tracked based on the IDs of its two tires. By capturing the TPMS status messages, vehicles on a road can be identified only with prior knowledge of the ID of its tires. Schulz and Junghans [68] explain how road networks can be equipped with wireless sensor networks to calculate traffic density, average speed, and traffic flow.

4.2.3. EV Charging Plug Infrastructure

The charging plug will be used for charging the car but also to send information about payments. Due to the use of electrical batteries, EVs require to be charged more often compared to traditional petrol-based cars. As a consequence, EVs will spend hours connected to EVSE and they will be subject to attacks through the charging plug interface. To protect against such attacks, the ISO 15118 standard has already introduced some security measures [69]. The ISO 15118 standard protects the information using TLS sessions. Once the communication begins, a TLS session is established to ensure mutual authentication and confidentiality between any EV and EVSE. However, many other security and privacy issues are left unanswered:

- **Key Management.** Identification and authorization according to the ISO/IEC 15118 standard could be achieved using private keys. These keys are internally stored in the EV. Lee *et al.* [70] criticize that the ISO 15118 standard does not achieve a secure communication for charging EVs. A static ID is used to authenticate each EV. However, an attacker may impersonate her victim EV by simply setting up the ID of her vehicle. Thus, after a successful charge, the bill will be charged to the victim. Further, an attacker may shut down EVSE by reporting its malfunctioning due to the possibility to report its faulty behaviour.

- **Integrity.** van den Broek, Poll and Vieira [71] describe that MACs are appended to any message to avoid tampering of the data. However, once the TLS tunnel is created, all the integrity measures are stripped. As a consequence, there is no mechanism to prove integrity of the message after the tunnel is established. The authors also discuss changing the communication paradigm from the end-to-end communication to a content-centric approach. In a content-centric approach, the data is secured instead of the communication channel.

- **Connection to the Smart Grid.** Smart grid is an electrical grid that uses cutting-edge technologies to improve the distribution of electrical energies. As shown in Figure 2, EVs are expected to be indirectly connected to the smart grid. In the intermediary process, there will be several agents as the EVSE, Charge Spot Operators (CSOs) and Distribution System Operators (DSOs). We believe that modern cars can be a new attack vector for the smart grid. The intermediary agents such as the EVSE, CSOs and DSOs are more prone to receive attacks than the EV.

- **EV Privacy.** The EV and the EVSE exchange information for mutual authentication. All communications are secured and take place through a mobile operator. That means that the mobile operator can track sensitive information and link EVs with certain EVSEs. If we assume that users tend to charge their cars at the same spot (*i.e.,* EVSE), the mobile operator could eventually track patterns user movements. In regards to payments, the payment system accepts credit cards, pre-paid cards and cash. While pre-paid and cash systems are fully anonymous, credit cards are linked to the identity of a user. Credit cards could be more privacy-invasive when users subscribe for a monthly contract with the mobile operator to pay at the end of the month.

Hofer *et al.* [1] present POPCORN, which improves privacy of the payment model by introducing a dispute resolver and a payment handler. Before starting the charging process, POPCORN requires that a contract is set up between the EV, a mobile operator and the EVSE. Thus, the mobile operator sends anonymous timespan certificates to the EV. Subsequently, the EV uses these certificates to authenticate to the EVSE. All the content exchanged between the parties is encrypted by using proxy encryption schemes.

4.2.4. Remote Keyless System (RKS)

In the following, we address security and privacy challenges in RKS along with state-of-the-art solutions:

- **Security of RKS.** Keyless systems have already been subject to attacks. Both [72] and [73] have hacked the active RKS. Using algebraic attacks, Courtois *et al.* [72] recover the key of the KeeLoq system. They analyzed properties of the encryption algorithm and generated 2^{16} plaintext keys. With these plaintext keys, the secret key can be recovered. [73] continued the algebraic attacks on the encryption system and improved the time to recover the key to few minutes. [74] defeated the passive RKS of the system. With relay attacks, an external box is configured to simulate the passive RKS interaction and to reproduce the previously sent signals from distances up to 8 meters. A security and privacy analysis of keyless car sharing system has been made in [75].

  RKS uses wireless technology and there are two important aspects to highlight. First, all these attacks can be mounted remotely in parking lots. Second, the infrastructure investment for hacking the RKS systems is really low. With Infotainment systems, the car can be unlocked from a smartphone application. The signal to unlock the car is sent through the Infotainment system.

- **Security of Immobilizer.** An immobilizer prevents the vehicle from starting by hot-wiring or forcing the mechanical lock. It is commonly based on passive transponders such as Hitag2, which uses a stream cipher (48-bit) for authentication and integrity. Verdult, Garcia and Balasch [76] describe several weaknesses in the design of the cipher. As the transponder lacks a pseudo-random number generator, they are susceptible to replay attacks. Also, the authors show an algebraic attack to reduce the number of possibilities and hack the system in less than 6 minutes. Busold *et al.* [77] present an immobilizer controlled by a smartphone application. The immobilizer is based on authentication of tokens: every authorized smartphone can grant rights to another car by creating a delegated token. For releasing tokens, the smartphone application uses the keys pre-installed by car manufacturers. The previously presented attack on KeeLoq systems [72] also apply to the immobilizer.

4.2.5. Infotainment and Telematics

The issues raised by the infotainment and telematics systems and possible solutions are discussed below:

- **Secure Interaction with Smartphones.** Modern infotainment systems already include WiFi hotspot capabilities. The radio can stream music from smartphones and even play videos. However, the connection between the infotainment and the smartphones must be secured. Kochanek *et al.* [78] introduce a security layer. Basically, a security tunnel is created: the initial keys are exchanged through the Elliptic Curve Diffie-Hellman (ECDH) and the communications are encrypted using AES-128. Timpner, Schürmann and Wolf [79] suggest to authenticate vehicles with external entities using smartphones. Smartphones receive credentials from a central CA and subsequently deploy the certificate into the vehicle by means of HSM of the infotainment system. For every communication, SSL tunnels are created and the CA identifies smartphones, where OAuth 2.0 is used. Dardanelli *et al.* [80] propose another secure communication layer but it works over Bluetooth. Their authentication scheme is composed of two steps. In the first step, the driver pushes a button located inside the car that sends the car identity using the Bluetooth protocol. The smartphone detects the identity and completes the pairing without the use of any key. In the second step, all the communication is protected with symmetric encryption, which is based on AES with 128-bit.

- **Privacy in Pay-As-You-Drive Insurance.** Connecting the car to the cloud or smartphones offers a new set of applications and business models [81]. Modern infotainment systems are constantly connected to a cloud provider. The cloud provider stores all the information related to the car: acceleration data, GPS coordinates, speed information and inflation of the tires. Some insurance companies calculate the insurance risk based on

this information[10]. Insurance companies gather such information through a black box installed in the car. They can build a database of location data of the car. It was already proven that the identity of consumers can be recovered from these traces [82]. Troncoso *et al.* [83] presented PriPAYD, a privacy-preserving scheme for the pay-as-you-drive insurance policy. Using PriPAYD, the insurance company only receives the minimum information necessary to bill the consumers. All the cost is calculated by the black box in the vehicle. All the calculation of the costs is performed in the vehicle black box by using encryption proxies. Balasch *et al.* [84] propose a privacy-preserving toll payment system. Their scheme computes fees locally and proves to the service provider that they carry out correct calculations. The service provider can challenge the vehicle with proofs about the locations where it has passed by. The vehicle reveals only the location requested by the proofs and the price at that location.

- **Security in Infotainment and Telematics.** A major vulnerability has been discovered in the BMW ConnectedDrive [85]. BMW was using the same symmetric keys in every device. The configuration data was neither tamper-proof nor protected with encryption in any manner.

5. Inter-Vehicle Communication

One of the most promising features of modern cars is its communication with other vehicles, *i.e.,* the inter-vehicle communication. The intercommunication between cars is used for offering innovative services and improving safety conditions. Data is exchanged entirely via wireless communication, using an amendment of *IEEE 802.11*, called *IEEE 802.11p*. *IEEE 802.11p* is placed into the networking layer of the network model. There exists an upper layer, IEEE 1609, that provides security services.

To describe the inter-vehicle network stack, we use the Open Systems Interconnect (OSI) model.

Physical and Data Link Layers: 802.11p. The physical and data link layers are controlled by a variant of the IEEE 802.11 protocol. IEEE 802.11 is a collection of physical layer (PHY) specifications and the media access control. As we already said, IEEE 802.11p is the recently approved amendment to the IEEE 802.11 standard [86]. IEEE 802.11p introduces the Wireless Access in Vehicular Environments (WAVE). It extends the 802.11 standard with the support for communication between vehicles and the roadside infrastructure.

IEEE 1609/WAVE. The networking layer is controlled by IEEE1609/WAVE. IEEE 1609/WAVE is a higher layer standard based on IEEE 802.11p. IEEE 1609/WAVE provides a high-rate and low-latency communication between WAVE devices. In the rest of this survey, we refer to IEEE1609/WAVE as WAVE (or IEEE1609). WAVE represents the network and transport layers in the OSI model. Basically, WAVE provides addressing, data delivery services and controls the communication layer. It specifies channel capabilities for multi-channel operations and management operations for data delivery among WAVE devices. WAVE has an assigned frequency that is divided into one control channel and six service channels [87].

WAVE provides data delivery services and other layers are built on WAVE. These other layers have two main functions: the management of communication channels and the deployment of services. The communication channel is controlled using WAVE control messages, *i.e.,* the WAVE Short Messages Protocol (WSMP). All other messages must be forwarded through service channels [88]. In service channels, the Internet Protocol version 6 (IPv6) is used for communication. In Figure 3, we summarize the inter-vehicle network stack.

Security Features of WAVE. For ensuring security of the network stack, WAVE provides security services. These services include confidentiality, authentication, authorization and integrity. To achieve confidentiality, every Protocol Data Unit (PDU) is sent encrypted. The integrity, authentication and authorization are guaranteed using digital signatures. In WAVE, received signed communications are verified using certificates associated with the signed data. A certificate contains a public key belonging to the certificate holder and the list of permissions associated with the public key. There exist two types of certificates: explicit and implicit. Explicit certificates are those that include the public key. Implicit certificates refer to those certificates that obtain the public key through additional processing. To reduce the size of the certificates, a certificate chain can be built. A certificate chain is a set of certificates where every

[10] `https://www.progressive.com/auto/snapshot/`

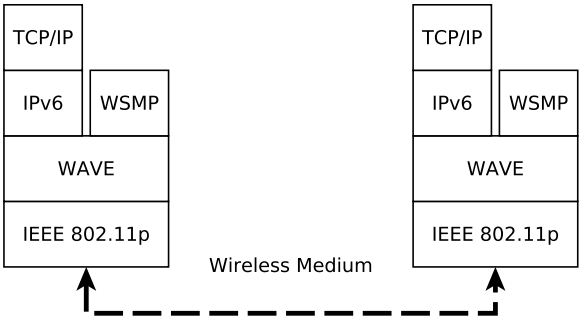

Figure 3: The inter-vehicle network stack.

certificate is the issuing certificate for the subsequent one [89]. The chain of certificates ends on a root CA, which builds trust in the PKI ecosystem.

Security and Privacy Issues

The inter-vehicle network presents security and privacy challenges. In the following, we discuss these challenges as well as existing solutions:

Authentication in V2V and V2I. In the context of vehicular networks, beaconing is the periodic transmission of packets to nearby vehicles or roadside units [90]. Technically, it is a one-hop link-layer broadcast message. These packets usually contain information about location, direction and speed of the vehicle, necessary for safety of vehicles. To avoid the possibility of unauthorized modifications, the exchanged data must be protected. To provide such protection, each vehicle signs the message using its signing key [91]. A receiving party can verify the signature using the corresponding verification key in the certificate. A signing and verification key pair ensures both authenticity and integrity of the data. However, the question is how a receiving party can know whether the verification key belongs to the intended vehicle or not. To make such a binding, digital certificates are used. A digital certificate contains information including, but not limited to, the verification key, the vehicle ID, issue and expiry dates of the certificate. This certificate is typically signed by a CA.

To ensure both authenticity and integrity, each beacon could be signed [91]. This implies that the vehicle has to send the message along with the digital signature. Since a receiving party cannot complete verification of the signed message without the corresponding verification key, the vehicle can also include the digital certificate in every beacon. It is important to note that an adversary can record a signed beacon to replay it later *a.k.a.* replay attack. To withstand against reply attacks, each beacon includes a timestamp, nonce or sequence number. Further, a signing key is stored in a tamper-resistant hardware to prevent extraction for unauthorized use.

There are three main issues with this naive scheme: (i) lack of privacy preservation due to verification using a digital certificate, (ii) increase in beacon size due to digital certificates and signatures and (iii) high computational overheads for verifying digital signatures [90]. Unfortunately, high-frequency beacons and dense traffic situations make the problem even worse.

In [92], Liu *et al.* propose a protocol for authentication in V2I using group communication in VANETs. As we know that an authentication protocol could reveal identity of the car, Yang *et al.* [93] present a solution for anonymous authentication for V2V and V2I settings.

Certificate Omission for Efficiency. Several solutions have been proposed to reduce computational overheads for verifying authenticity. First, in order to speed up cryptographic operations, it is preferred to (i) use the Elliptic Curve Cryptography (ECC) over RSA and (ii) use of dedicated hardware, *i.e.,* ECC ASICs [94]. MAC can speed up performance as compared to asymmetric counterpart; however, it comes up with its own problem of key distribution and sharing. Caching signatures of verified certificates could be used for achieving efficiency. In [90], the authors propose omitting certificates in high-frequency beacons and suggest transferring the certificates at a lower frequency. There are further techniques to improve efficiency by avoiding the periodic omission of certificates. In [95], Feiri,

Petit and Kargl revisit the basic certificate omission and propose the Neighbour-based Certificate Omission (NbCO) and the Congestion-based Certificate Omissions (CbCO) as potential solutions. NbCO includes the certificates only when their neighbours have been changed; whereas, CbCO regulates the number of certificates to verify according to the traffic in the network.

VPKI. Modern cars are authenticated with registration plates, which are assigned by local authorities. In addition, a unique Vehicle Identification Number (VIN) is also marked in certain parts of the car. In inter-vehicle communications, vehicles need to provide their identification in many situations. Registration plates and VINs cannot be sent within the messages, due to the risk of tracking. Indeed, any identification used in the wireless messages can be easily retrieved and associated with the driver and passengers of the car, thus leaking privacy of users. However, the full anonymity is neither acceptable because identity of drivers or cars must be accessible in case of accidents and for forensic purposes. There must exist some sort of accountability for the messages being exchanged.

Cars must be able to communicate without leaking their identification and at the same time still be traceable by enforcement agencies. For instance, a Vehicular Public Key Infrastructure (VPKI) must be built to support authentication, authorization and accountability. However, the construction of such infrastructure imposes several constraints. Multiple identities must be assigned to a car to prevent its de-anonymization. The identification keys should be changed with time to ensure location privacy. VPKI must cope with all the cars in a country. If we consider the case of the United States with 300 million vehicles, at least 300 billion certificates will be generated per year [96]. VPKI must be efficient and fast enough to work properly with cars travelling at high-speed.

VPKI has been proposed in [97, 96, 98]. Nilsson *et al.* [98] focus on a low-cost infrastructure to reduce costs but they consider VPKI as a central entity, which is always available. Their approach is based on negotiation of keys between the vehicle and the infrastructure. Alexiou *et al.* [97] introduce the concept of ticketing for holding and requesting multiple credentials. These tickets can be valid for short or long periods of time and are negotiated in a one-way fashion to avoid linkability of tickets to vehicles. Whyte *et al.* [96] propose the use of a butterfly cryptographic construction. The butterfly keys permit a device to request an arbitrary number of certificates. Every certificate is generated with different keys. The advantage of this mechanism is the fast generation of multiple keys.

For revocation of issued keys, the infrastructure provides a list of revoked pseudonyms, the short-term and the long-term certificates [97, 96]. Then, every vehicle must download the list from a resolution authority. In [96], Whyte *et al.* handle the list of revoked pseudonyms and certificates using different entities to avoid the risks of potential linkage. In [97], a resolution authority generates a digitally signed request to the pseudonym CA. Both CAs exchange data to be able to de-anonymize the vehicle. Whyte *et al.* [96] propose the addition of another certification entity called the linkage authority. This authority generates a linkage of values between the vehicles and the butterfly-generated certificates. Once the authorization has linked to the certificate, the resolution authority de-anonymizes the vehicle.

Pseudonyms (Location Privacy Protection). The broadcast nature of the vehicular communication increases the risk of leaking sensitive information. Position coordinates, distance to objects, acceleration and speed data are exchanged continuously by V2V applications such as carpooling or autonomous driving. As such, an attacker may easily track the car, thus compromising privacy of users. Further, there exists a strong correlation between the driver and her car [99]. To ensure privacy of users, one straightforward solution is to hide position of the vehicle. However, hiding position of the vehicle will render V2V applications (including driverless cars) useless. The alternate approach is to hide IDs vehicles using pseudonyms [100].

A pseudonym is an identifier that differs from the original name of the vehicle and it does not include any identifying information such as driver name or ID of the car [101]. A pseudonym is simply the identity of a node in the V2V network. However, samples provided by the pseudonym can be collected to build profiles. By analyzing these profiles, an attacker may track the vehicle. To harden this tracking, some schemes change periodically the pseudonyms or use a set of pseudonyms per vehicle rather than only one. For example, SLOW [102] is a pseudonym changing scheme for preserving location privacy in vehicular networks. A similar scheme has been proposed in [103] by Gerlach and Guttler. In [104], Freudiger *et al.* suggest the creation of mix-zones at road intersections. They also consider mix-zones as mix-networks to analyze the location privacy in vehicular networks, where a vehicle uses a unique pseudonym (or pseudonym key) in each zone [105]. They assume that each vehicle is equipped with a set of pseudonyms [106]. In order to efficiently communicate within a zone, establishment of symmetric keys has been proposed. The major issue with this scheme is a large set of pseudonyms that are required. More precisely, a recent study shows that a vehicle requires 105,120 pseudonyms in a year, assuming each one is valid for 5.5 minutes [107]. In order to solve this issue, Feiri *et al.* [108] propose a technique to use PUFs for storing only the master key, which could be password-protected.

For protecting regular keys, they analyze two main design choices. First, pseudonym keys could be encrypted under the master key. Second, pseudonym keys could be derived from the master key.

The authors in [99] show that an adversary can significantly weaken the location privacy of vehicles. By linking different pseudonyms, they were able to reconstruct locations of the vehicles within the same area.

Data Fusion. Safety features of the vehicle depend on decisions taken by multiple and diverse sensors. In 2015, Tesla launched a system for automatic driving[11]. Every driving decision is based on multiple sensors: calculation of distance based on the radar and visual information and lane change based on multiple cameras are typical examples. To calculate the distance accurately, the distance measured by a radar must be combined with information extracted from an image. The lane change must take place once the front cameras report that the situation is safe and rear cameras assure that no vehicle is already overtaking. All these sensors have a different perception of reality and a decision must be made by consensus. To complicate further the problem, some sensors may be untrustworthy and provide an incorrect vision of reality, especially when the information is coming from third party sources, such as other cars. The solution for this problem is achieved with data fusion. Data fusion is basically the integration of multiple knowledge representations of situations or objects into consistent, accurate, and useful representations. Data fusion is an enabler technology that combines information from several sources in order to form a unified decision [109].

Confidential Secure Aggregation. Vehicles require the presence of data in plaintext for the aggregation process. Every vehicle or sensor reports a representative value of the measured reality that is confronted against values provided by other vehicles. Next, a consensus is made with a certain procedure, such as an average calculation.

Security and data fusion are achieved in a hop-by-hop fashion. Vehicles decrypt every message they receive and aggregate the messages according to the aggregation function. Subsequently, the vehicles encrypt and forward the aggregated message to the next node. Therefore, the confidentiality of the data presents a problem. Kefayati *et al.* [110] propose BIFF, a mechanism for aggregating information while preserving confidentiality. In this scheme, every node is responsible for transforming data into a common place where encrypted information can be fused without revealing the data or its source.

Data Aggregation. As far as the communication is concerned, the information is disseminated through broadcast messages within vehicular networks. Consequently, each vehicle might need to process a very high volume of data. Unfortunately, a vehicle might not be able to cope with pace of the data generated, transmitted and processed. Even worse, some data cryptographic techniques for establishing a secure channel, say for preserving confidentiality and integrity, could add additional complexity. Thus, scalability is a crucial factor [111]. Data aggregation is one solution to achieve scalability. It semantically combines the information and provides the summarised view of the information. Data aggregation could be used to report traffic information, for instance, rainy road warnings, free parking spots and detection of accidents [112].

Integrity of the Aggregated Data. The fundamental problem is that the integrity of aggregated information can no longer be verified. To address this problem, Raya, Aziz and Hubaux [113] aggregate data based on consensus and protect integrity with signatures. In their proposal, a set of vehicles agrees on a common value. Then, one of them signs the agreed data. The message is re-transmitted to another node that re-signs the data including the signature of the first vehicle. The work in [114] improves the scheme by removing the need of having a common set of vehicles.

Certain structures protect the aggregation process using probabilistic counting solutions, called sketches. Sketches store a summary of a dataset. They can efficiently answer questions about the data with small computational costs at the price of a statistical error. Garofalakis, Hellerstein, and Maniatis [115] store the information in Bloom filters. A Bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set or not. The work in [114] saves storage; however, a central authority must be always available to certify the aggregation process. Dietzel *et al.* [116] improve the previous mechanism but saving complex values of data and using statistical analysis to remove the outliers of the information.

6. Conclusion and Future Directions

Modern cars are equipped with multiple controllers that improve safety and comfort in the vehicle, but they also increase the potential attack surface. The problem is exacerbated by continuous connectivity with roadside

[11] http://www.computerworld.com/article/2993326/telematics/tesla-enables-autosteer-other-autonomous-tweaks-in-its-model-s.html

infrastructures and among cars. In this survey, we review the status of security and privacy issues in modern cars. The existing literature has been classified into three categories according to the underlying technology: internal vehicular network (intra-vehicle), the gateways of the intra-vehicle network with external entities and devices (gateways) and the communication between vehicles (inter-vehicle). For the pedagogical nature of this survey, we have first introduced the underlying technology and then described the security and privacy issues for each category. Every problem includes a description and the solutions adopted into the literature.

Despite the wealth of solutions proposed in the literature, we believe that there are several open problems concerning security and privacy in modern cars. In the following, we highlight some of these exciting open problems.

Effective Network Monitoring. For network monitoring, modern cars could use state-of-the-art IDSs. However, the major challenge associated with deploying an IDS is detecting anomalies and responding in real-time in order to reduce possible damages caused by potential attackers. For discovering a new array of attacks, deploying honeypots could be set up so that attack traffic can be collected. It is unclear how modern cars can fully exploit attack-related information extracted from such traffic. In the context of modern cars, investigating a generic framework for collecting the traffic from honeypots and exploiting extracted information for enriching IDSs would be an interesting research direction.

Secure and Efficient Data Processing. For a secure data exchange, existing solutions use digital certificates, which guarantee a set of security properties including confidentiality and integrity. To achieve efficiency, some solutions propose omitting certificates for instance in case of congestion. Unfortunately, an adversary can generate fake congestion to trigger certificate omission, which can result in compromising security of vehicular networks. To deal with this challenging issue, there is a need to come up with novel mechanisms that can guarantee rigorous security without degrading performance in vehicular networks.

Scalable and Privacy-Preserving Services. Vehicular networks are expected to generate a huge amount of data, *i.e.,* big data. This big data could be used for providing a wide range of safety, entertainment and value-added services. On one hand, such services are becoming essential and useful; on the other hand, they could compromise privacy of users. To ensure privacy, it is required to have not only privacy-preserve solutions but also scalable ones.

Practical and Reliable Data Fusion. For offering a variety of services, modern cars rely on information provided by different entities, say sensors within a car and other cars. Data fusion is required to integrate information provided by different entities to make service-related decisions. Collecting information to do data fusion for providing reliable information, which could be used in practice, is a challenging problem. Without loss of generality, data fusion must be done without compromising confidentiality or integrity.

Acknowledgment

We would like to thank the Editor-in-Chief and reviewers for their useful feedback that helped in improving quality of our work. This work has been partially funded by the project XProbes sponsored by the Provincia Autonoma di Trento, Italy.

References

[1] Höfer et al., "Popcorn: Privacy-preserving charging for emobility," ser. CyCAR '13. New York, NY, USA: ACM, 2013, pp. 37–48.
[2] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *Black Hat USA*, 2014.
[3] J. Petit and S. Shladover, "Potential cyberattacks on automated vehicles," *IEEE TITS*, vol. 16, no. 2, pp. 546–556, April 2015.
[4] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 528–533.
[5] Ruddle et al., "Security requirements for automotive on-board networks based on dark-side scenarios," *EVITA Deliverable D*, vol. 2, p. 3, 2009.
[6] E. Rippel, "Embedded security challenges in automotive designs," Discretix Technologies, Tech. Rep., 2009.
[7] ISO-26262-1, "Road vehicles Functional Safety," International Organization for Standardization, Geneva, Switzerland, ISO 26262-1, 2011.
[8] Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces," in *USENIX*, ser. SEC'11, Berkeley, CA, USA, 2011, pp. 6–6.
[9] Karagiannis et al., "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 4, pp. 584–616, 2011.

[10] AUTOSAR, "AUTOSAR Release 4.2: Overview of Functional Safety Measures in AUTOSAR," 2014.
[11] ——, "AUTOSAR release 4.2.1: Requirements on crypto service manager," 2014.
[12] ——, "AUTOSAR release 4.2.2: Utilization of crypto services," 2014.
[13] ——, "AUTOSAR release 4.2.2: Specification of module secure onboard communication," 2014.
[14] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "CANAuth-a simple, backward compatible broadcast authentication protocol for CAN bus," in *ECRYPT Workshop on Lightweight Cryptography 2011*, 2011.
[15] "Bosch - literature on the CAN bus," 2015, last accessed: May 8, 2017. [Online]. Available: http://archive.is/http://www.bosch-semiconductors.de/en/ubk_semiconductors/ip_modules_3/produkttabelle_ip_modules/can_literature_1/can_literature.html
[16] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks — practical examples and selected short-term countermeasures," ser. SAFECOMP '08. Springer-Verlag, 2008, pp. 235–248.
[17] M. Faezipour, M. Nourani, A. Saeed, and S. Addepalli, "Progress and challenges in intelligent vehicle area networks," *Commun. ACM*, vol. 55, no. 2, pp. 90–100, February 2012.
[18] Sagstetter et al., "Security challenges in automotive hardware/software architecture design," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, March 2013, pp. 458–463.
[19] Bosch CAN-FD, "CAN with Flexible Data-Rate," International Organization for Standardization, Geneva, Switzerland, Specification 1.0 2012, 2012.
[20] "Introduction to the local interconnect network (LIN) bus," November 2003, last accessed: May 8, 2017. [Online]. Available: http://www.ni.com/white-paper/9733/en/
[21] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-vehicle networks: A review," *IEEE TITS*, vol. 16, no. 2, pp. 534–545, 2015.
[22] ISO-17458, "Road vehicles – flexray communications system – part 1," 2013.
[23] Open Alliance - BroadR-Reach, "BroadR-Reach Specifications for Communication Channel," OPEN Alliance SIG, Specification, 2015.
[24] Ixia, "Automotive ethernet: An overview," Ixia, Tech. Rep., May 2014.
[25] "802.1AS-2011/Cor 1-2013 - IEEE standard for local and metropolitan area networks timing and synchronization for time-sensitive applications in bridged local area networks corrigendum 1," 2013.
[26] IEEE-802.1Qbv, "Ieee draft standard for local and metropolitan area networks - Media Access Control (MAC) bridges and virtual bridged local area networks amendment: Enhancements for scheduled traffic," 2015.
[27] M. Wolf and T. Gendrullis, "Design, implementation, and evaluation of a vehicular hardware security module," in *Information Security and Cryptology - ICISC 2011*. Springer, 2012, pp. 302–318.
[28] P. Mundhenk, S. Steinhorst, M. Lukasiewycz, S. A. Fahmy, and S. Chakraborty, "Lightweight authentication for secure automotive networks," ser. DATE '15. San Jose, CA, USA: EDA Consortium, 2015, pp. 285–288.
[29] T. Piper, S. Winter, O. Schwahn, S. Bidarahalli, and N. Suri, "Mitigating timing error propagation in mixed-criticality automotive systems," in *Real-Time Distributed Computing (ISORC), 2015 IEEE 18th International Symposium on*, April 2015, pp. 102–109.
[30] Matsumoto et al., "A method of preventing unauthorized data transmission in controller area network," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*. IEEE, 2012, pp. 1–5.
[31] I. Broster and A. Burns, "An analysable bus-guardian for event-triggered communication," in *Real-Time Systems Symposium, 2003. RTSS 2003. 24th IEEE*. IEEE, 2003, pp. 410–419.
[32] R. Kurachi and H. Takada, "SecGW secure gateway for in-vehicle networks," in *13th escar Embedded Security in Cars Conference, Berlin, Germany*, 2014.
[33] M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars*, 2004.
[34] V. Verendel, D. Nilsson, U. Larson, and E. Jonsson, "An approach to using honeypots in in-vehicle networks," in *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, September 2008, pp. 1–5.
[35] H. Schweppe and Y. Roudier, "Security and privacy for in-vehicle networks," in *IEEE VCSC Workshop*, June 2012, pp. 12–17.
[36] A. Bouard, B. Weyl, and C. Eckert, "Practical information-flow aware middleware for in-car communication," ser. CyCAR '13. New York, NY, USA: ACM, 2013, pp. 3–8.
[37] B. Klopper, S. Honiden, J. Meyer, and M. Tichy, "Planning with utility and state trajectory constraints in self-healing automotive systems," in *IEEE SASO 2010*, September 2010, pp. 74–83.
[38] A. Biedermann, S. A. Huss, and A. Israr, "Safe dynamic reshaping of reconfigurable MPSoC embedded systems for self-healing and self-adaption purposes," *ACM Trans. Reconfigurable Technol. Syst.*, vol. 8, no. 4, pp. 26:1–26:22, September 2015.
[39] I. Jahnich, I. Podolski, and A. Rettberg, "Towards a middleware approach for a self-configurable automotive embedded system," in *Software Technologies for Embedded and Ubiquitous Systems*, ser. Lecture Notes in Computer Science, 2008, vol. 5287, pp. 55–65.
[40] R. Anthony, A. Rettberg, D. Chen, I. Jahnich, G. de Boer, and C. Ekelin, "Towards a dynamically reconfigurable automotive control system architecture," in *Embedded System Design: Topics, Techniques and Trends*. Springer, 2007, pp. 71–84.
[41] R. Kammerer, B. Frömel, and A. Wasicek, "Enhancing security in CAN systems using a star coupling router," in *Industrial Embedded Systems (SIES), 2012 7th IEEE International Symposium on*. IEEE, 2012, pp. 237–246.
[42] S. Nürnberger and C. Rossow, – *vatiCAN – Vetted, Authenticated CAN Bus*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 106–124.
[43] P. Werner, A. Happel, R. Fritz, and S. Keul, "Autosar security modules," in *14th escar Embedded Security in Cars Conference, Berlin, Germany*, 2015.
[44] Schweppe et al., "Securing car2X applications with effective hardware software codesign for vehicular on-board networks," *VDI Automotive Security*, vol. 27, 2011.
[45] O. Hartkopp, C. Reuber, and R. Schilling, "Macan - message authenticated can," in *10th conference for Embedded Security in Cars (Escar'12)*, Berlin, Germany, November 2012.
[46] F. Sagstetter, M. Lukasiewycz, S. Steinhorst, M. Wolf, A. Bouard, W. R. Harris, S. Jha, T. Peyrin, A. Poschmann, and S. Chakraborty, "Security challenges in automotive hardware/software architecture design," in *Proceedings of the Conference on Design, Automation and*

Test in Europe. EDA Consortium, 2013, pp. 458–463.
[47] A. Wasicek, "Protection of intellectual property rights in automotive control units," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 7, no. 2014-01-0338, pp. 201–212, 2014.
[48] S. Malipatlolla, S. Huss *et al.*, "A novel method for secure intellectual property deployment in embedded systems," in *Programmable Logic (SPL), 2011 VII Southern Conference on*. IEEE, 2011, pp. 203–208.
[49] A. Wasicek, "Copy protection for automotive electronic control units using authenticity heartbeat signals," in *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on*, July 2012, pp. 821–826.
[50] ISO-14229-1, "Road vehicles – unified diagnostic services (UDS) – part 1: Specification and requirements," 2013.
[51] ISO-15765-1, "Road vehicles – diagnostic communication over controller area network (DoCAN) – part 1," 2011.
[52] ISO-15331-1, "Road vehicles – communication between vehicle and external equipment for emissions-related diagnostics – part 1," 2010.
[53] ISO-13400-2, "Road vehicles Diagnostic communication over Internet Protocol (DoIP)," International Organization for Standardization, Geneva, Switzerland, ISO 13400-2 2012, 2012.
[54] SAE-J1772, "Electric vehicle and plug in hybrid electric vehicle conductive charge coupler," Society of Automotive Engineers International, SAE Standard J1772:2012, 2012.
[55] "Smart charging standards for plug-in electric vehicles," Society of Automotive Engineers International, SAE Standard 2014-01-1823, 2014.
[56] H. Chaudhry and T. Bohn, "Security concerns of a plug-in vehicle," in *Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES*, January 2012, pp. 1–6.
[57] A. HIS, "Security, she: Secure hardware extension version 1.1," p. 53, 2009. [Online]. Available: www.automotive-his.de
[58] T. E. Project, "Deliverable d3.2: Secure on-board architecture specification," 2008.
[59] A. Kanuparthi, R. Karri, and S. Addepalli, "Hardware and embedded security in the context of internet of things," ser. CyCAR '13. New York, NY, USA: ACM, 2013, pp. 61–64.
[60] P. Kleberger and G. Moulin, "Short paper: Formal verification of an authorization protocol for remote vehicle diagnostics," in *IEEE VNC*, December 2013, pp. 202–205.
[61] U. Drolia, Z. Wang, Y. Pant, and R. Mangharam, "AutoPlug: An automotive test-bed for electronic controller unit testing and verification," in *IEEE ITSC 2011*, October 2011, pp. 1187–1192.
[62] Lee et al., "Formally verifiable features in embedded vehicular security systems," in *IEEE VNC*, October 2009, pp. 1–7.
[63] Shavit et al., "Firmware update over the air (FOTA) for automotive industry," SAE Technical Paper, Tech. Rep., 2007.
[64] ISO-13400-1, "Diagnostic communication over internet protocol (DoIP) – part 1: General information and use case definition," International Organization for Standardization, Geneva, Switzerland, ISO 13400-1 2011, 2011.
[65] J. Lindberg, "Security analysis of vehicle diagnostics using DoIP," 2011.
[66] Roufa et al., "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *19th USENIX*, 2010, pp. 11–13.
[67] M. Xu, W. Xu, J. Walker, and B. Moore, "Lightweight secure communication protocols for in-vehicle sensor networks," ser. CyCAR '13. New York, NY, USA: ACM, 2013, pp. 19–30.
[68] J. Schulz and M. Junghans, "Deployment of tire pressure monitoring systems for traffic monitoring and safety purposes," in *18th World Congress on Intelligent Transportation Systems, 16th-20th October*, 2011.
[69] ISO-15118, "15118-1 road vehicles-vehicle to grid communication interface-part 1: General information and use-case definition." [Online]. Available: http://standardsdevelopment.bsigroup.com/Home
[70] S. Lee, Y. Park, H. Lim, and T. Shon, "Study on analysis of security vulnerabilities and countermeasures in ISO/IEC 15118 based electric vehicle charging technology," in *ICITCS*, October 2014, pp. 1–4.
[71] F. van den Broek, E. Poll, and B. Vieira, "Securing the information infrastructure for EV charging," in *Wireless and Satellite Systems*. Springer, 2015, pp. 61–74.
[72] N. T. Courtois, G. V. Bard, and D. Wagner, "Algebraic and slide attacks on keeloq," in *Fast Software Encryption*. Springer, 2008, pp. 97–115.
[73] Eisenbarth et al., "On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme," in *Advances in Cryptology–CRYPTO 2008*. Springer, 2008, pp. 203–220.
[74] Francillon et al, "Relay attacks on passive keyless entry and start systems in modern cars," in *NDSS*, 2011.
[75] I. Symeonidis, M. A. Mustafa, and B. Preneel, "Keyless car sharing system: A security and privacy analysis," in *Smart Cities Conference (ISC2), 2016 IEEE International*. IEEE, 2016, pp. 1–7.
[76] R. Verdult, F. D. Garcia, and J. Balasch, "Gone in 360 seconds: Hijacking with hitag2," in *USENIX*, 2012, pp. 37–37.
[77] Busold et al., "Smart keys for cyber-cars: secure smartphone-based nfc-enabled car immobilizer," in *ACM conference on Data and application security and privacy*. ACM, 2013, pp. 233–242.
[78] A. Hazem and H. A. Fahmy, "Secure integration of mobile devices for automotive services," in *EsCAR, Germany*, vol. 6, 2012.
[79] J. Timpner, D. Schürmann, and L. Wolf, "Secure smartphone-based registration and key deployment for vehicle-to-cloud communications," ser. CyCAR '13. New York, NY, USA: ACM, 2013, pp. 31–36.
[80] Dardanelli et al., "A security layer for smartphone-to-vehicle communication over bluetooth," *Embedded Systems Letters, IEEE*, vol. 5, no. 3, pp. 34–37, September 2013.
[81] F. Koushanfar, A.-R. Sadeghi, and H. Seudie, "Eda for secure and dependable cybercars: Challenges and opportunities," in *Annual Design Automation Conference*, ser. DAC '12. New York, NY, USA: ACM, 2012, pp. 220–228.
[82] P. Golle and K. Partridge, "On the anonymity of home/work location pairs," in *Pervasive computing*. Springer, 2009, pp. 390–397.
[83] C. Troncoso, G. Danezis, E. Kosta, J. Balasch, and B. Preneel, "Pripayd: Privacy-friendly pay-as-you-drive insurance," *IEEE TDSC*, vol. 8, no. 5, pp. 742–755, 2011.
[84] J. Balasch, A. Rial, C. Troncoso, B. Preneel, I. Verbauwhede, and C. Geuens, "PrETP: Privacy-preserving electronic toll pricing." in *USENIX Security Symposium*, 2010, pp. 63–78.
[85] "Beemer, open thyself! security vulnerabilities in BMW's connecteddrive," February 2015, last accessed: May 8, 2017. [Online].

Available: http://www.heise.de/ct/artikel/Beemer-Open-Thyself-Security-vulnerabilities-in-BMW-s-ConnectedDrive-2540957.html
[86] Abdelgader et al., "The physical layer of the IEEE 802.11 p WAVE communication standard: The specifications and challenges," in *WCECS*, vol. 2, 2014.
[87] "IEEE standard for Wireless Access in Vehicular Environments (WAVE)–multi-channel operation," *IEEE Std 1609.4-2010 (Revision of IEEE Std 1609.4-2006)*, pp. 1–89, February 2011.
[88] S. Cespedes, N. Lu, and X. Shen, "VIP-WAVE: On the feasibility of IP communications in 802.11 p vehicular networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 1, pp. 82–97, 2013.
[89] "IEEE draft standard for wireless access in vehicular environments - security services for applications and management messages," *IEEE P1609.2/D10, October 2015*, pp. 1–233, October 2015.
[90] Schoch et al, "On the efficiency of secure beaconing in VANETs," ser. WiSec '10. New York, NY, USA: ACM, 2010, pp. 111–116.
[91] Papadimitratos et al., "Secure vehicular communication systems: design and architecture," *Communications Magazine, IEEE*, vol. 46, no. 11, pp. 100–109, 2008.
[92] Y. Liu, W. Guo, Q. Zhong, and G. Yao, "LVAP: Lightweight V2I authentication protocol using group communication in VANETs," *International Journal of Communication Systems*, 2017.
[93] Yang et al., "V2X security: A case study of anonymous authentication," *Pervasive and Mobile Computing*, 2017.
[94] S. Peter, P. Langendörfer, and K. Piotrowski, "Flexible hardware reduction for elliptic curve cryptography in GF (2m)," in *DATE'07*. IEEE, 2007, pp. 1–6.
[95] M. Feiri, J. Petit, and F. Kargl, "Evaluation of congestion-based certificate omission in VANETs," in *IEEE VNC*, Nov 2012, pp. 101–108.
[96] W. Whyte, A. Weimerskirch, V. Kumar, and T. Hehn, "A security credential management system for V2V communications," in *IEEE VNC*. IEEE, 2013, pp. 1–8.
[97] Alexiou et al., "VeSPA: Vehicular security and privacy-preserving architecture," ser. HotWiSec '13. New York, NY, USA: ACM, 2013, pp. 19–24.
[98] D. Nilsson, U. Larson, and E. Jonsson, "Low-cost key management for hierarchical wireless vehicle networks," in *IEEE IVS 2008*, June 2008, pp. 476–481.
[99] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough," in *WONS 2010*, Feb 2010, pp. 176–183.
[100] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," ser. VANET '07. New York, NY, USA: ACM, 2007, pp. 19–28.
[101] Papadimitratos et al., "Privacy and identity management for vehicular communication systems : A position paper," in *Workshop on Standards for Privacy in User-Centric Identity Management*, 2006, qC 20120220.
[102] Buttyan et al., "SLOW: A practical pseudonym changing scheme for location privacy in VANETs," in *IEEE VNC 2009*, 2009, pp. 1–8.
[103] M. Gerlach and F. Guttler, "Privacy in VANETs using changing pseudonyms - ideal and real," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th*, April 2007, pp. 2521–2525.
[104] Freudiger et al., "Mix-zones for location privacy in vehicular networks," in *Win-ITS*, 2007, qC 20110707.
[105] A. Beresford and F. Stajano, "Mix zones: user privacy in location-aware services," in *Pervasive Computing and Communications Workshops, 2004*, March 2004, pp. 127–131.
[106] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Commun. ACM*, vol. 24, no. 2, pp. 84–90, 1981.
[107] D. Garcia, A. Waite, R. Walsh, B. Sheppard, L. Frank, and D. Jeffers, "Certificate management entities for connected vehicle environment. public workshop read-ahead document," Tech. Rep., 2012.
[108] M. Feiri, J. Petit, and F. Kargl, "Efficient and secure storage of private keys for pseudonymous vehicular communication," ser. CyCAR '13. New York, NY, USA: ACM, 2013, pp. 9–18.
[109] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: A review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28 – 44, 2013.
[110] Kefayati et al., "On secure consensus information fusion over sensor networks." IEEE, 2007, pp. 108–115.
[111] S. Dietzel, E. Schoch, F. Kargl, B. Könings, and M. Weber, "Resilient secure aggregation for vehicular networks," *IEEE Network*, vol. 24, no. 1, pp. 26–31, January 2010.
[112] A. Polychronopoulos, U. Scheunert, and F. Tango, "Centralized data fusion for obstacle and road borders tracking in a collision warning system," in *International Conference on Information Fusion*, 2004.
[113] M. Raya, A. Aziz, and J.-P. Hubaux, "Efficient secure aggregation in VANETs," in *international workshop on Vehicular ad hoc networks*. ACM, 2006, pp. 67–75.
[114] R. W. van der Heijden, S. Dietzel, and F. Kargl, "SeDyA: secure dynamic aggregation in VANETs," in *ACM conference on Security and privacy in wireless and mobile networks*. ACM, 2013, pp. 131–142.
[115] M. Garofalakis, J. M. Hellerstein, and P. Maniatis, "Proof sketches: Verifiable in-network aggregation," in *Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on*. IEEE, 2007, pp. 996–1005.
[116] S. Dietzel, J. Gurtler, R. van der Heijden, and F. Kargl, "Redundancy-based statistical analysis for insider attack detection in VANET aggregation schemes," in *IEEE VNC*, December 2014, pp. 135–142.

Vitae

Cesar Bernardini is a Post-Doctoral Researcher at Aalto University, Finland. He obtained his Ph.D. at the University of Lorraine, France and pursued his career as a Post-Doctoral Researcher at the University of Trento in Italy and the University of Innsbruck in Austria. His main interests are network security, applied cryptography, and privacy.

Muhammad Rizwan Asghar is a Senior Lecturer in the Department of Computer Science at The University of Auckland in New Zealand. Prior to that, he was a Post-Doctoral Researcher at international research institutes including Saarland University in Germany and CREATE-NET in Trento Italy, where he also served as a Researcher. He received his Ph.D. degree from the University of Trento, Italy in 2013. As part of his Ph.D. program, he was a Visiting Fellow at the Stanford Research Institute (SRI), California, USA. He obtained his M.Sc. degree in Information Security Technology from the Eindhoven University of Technology (TU/e), The Netherlands in 2009. His research interests include access control, applied cryptography, security, privacy, cloud computing, and distributed systems. He is a Member of ACM and IEEE.

Bruno Crispo is Professor in the Department of Computer Science at the KU Leuven in Belgium and at the University of Trento in Italy. His main research interest lies in the area of system and network security. His recent work focuses on smartphone and mobile app security, security and privacy of the Internet of Things (IoT) and behavioral biometrics. He is an Associate Editor of the ACM Transactions on Privacy and Security. He is a Senior Member of IEEE.