A Survey of Security and Privacy in Connected Vehicles

Lotfi ben Othmane[1], Harold Weffers[1], Mohd Murtadha Mohamad[2], and Marko Wolf[3]

[1] Dept. of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, Netherlands
{l.ben.othmane, h.t.g.weffers}@tue.nl
[2] Fac. of Computer Science and Information System
Universiti Teknologi Malaysia
Johor, Malaysia
murtadha@utm.my
[3] ESCRYPT GmbH–Embedded Security
Munich, Germany
marko.wolf@escrypt.com

Summary. Electronic Control Units (ECUs) of a vehicle control the behavior of its devices–e.g., break and engine. They communicate through the in-vehicle network. Vehicles communicate with other vehicles and Road Side Units (RSUs) through Vehicular Ad-hoc Networks (VANets), with personal devices through Wireless Personal Area Networks (WPANs), and with service center systems through cellular networks. A vehicle that uses an external network, in addition to the in-vehicle network, is called connected vehicle.

A connected vehicle could benefit from smart mobility applications: applications that use information generated by vehicles, e.g., cooperative adaptive cruise control. However, connecting in-vehicle network, VANet, WPAN, and cellular network increases the count and complexity of threats to vehicles, which makes developing security and privacy solutions for connected vehicles more challenging.

In this work we provide a taxonomy for security and privacy aspects of connected vehicle. The aspects are: security of communication links, data validity, security of devices, identity and liability, access control, and privacy of drivers and vehicles. We use the taxonomy to classify the main threats to connected vehicles, and existing solutions that address the threats. We also report about the (only) approach for verifying security and privacy architecture of connected vehicle that we found in the literature. The taxonomy and survey could be used by security architects to develop security solutions for smart mobility applications.

1 Introduction

Each vehicle uses a set of sensors and Electronic Control Units (ECUs) to collect data about the vehicle′s behavior and environment, and to control the functionalities of the vehicle. ECUs (of a vehicle) collaborate by exchanging messages; they compose an *in-vehicle network* (a.k.a. on-Board network). Figure 1 depicts an example of architecture of in-vehicle network.

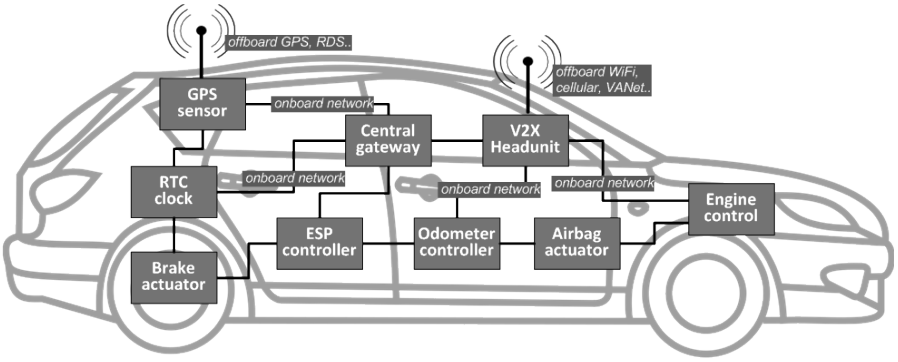

Fig. 1. Example of architecture of in-vehicle network.

Vehicles, in a road, exchange messages with neighboring (close by) vehicles and with Road Side Units (RSUs); they communicate directly–without intermediate node(s), or indirectly–through intermediate node(s). Each vehicle communicates with the neighboring RSUs to inform them about itself (the information may include location, speed, and heading) and to get traffic conditions of the road. Vehicles and RSUs compose *Vehicular Ad-hoc Networks (VANets)* [1](a.k.a. inter-vehicle network). Members of a VANet exchange information, for example, to have a shared knowledge about the traffic.

A Wireless Personal Area Network (WPAN) is composed of personal devices[4] that communicate through short range (from few centimeters to 100 m) wireless technologies, such as, Bluetooth, Near Field Communication (NFC), and Infrared (IR). WPAN gateway, of a vehicle, enables exchange of messages between personal devices (e.g., Personal Digital Assistances (PDAs) and iPods) and in-vehicle ECUs. For example, a driver controls the lights, windshield wipers, air flow, and heat of his vehicle through a Bluetooth-enabled headset [2], or even start remotely its engine and unlock its doors using his PDA.

Vehicles, equipped with mobile devices, exchange messages with Service Centers (SCs) through cellular network (a.k.a. mobile network); they provide data about their locations, behavior, and environment. An *SC* is a remote

[4] Infotainment devices are in most cases connected to the in-vehicle network; they are not members of the WPAN.

office that receives and sends data to vehicles or RSUs in order to assist and provide services to drivers, vehicle owners, and the public community. Example of SCs are fleet management systems. The *cellular network* enables communications of devices that have wireless communication capabilities (e.g., mobile phones) with mobile and land phones (cf. [3]).

Several applications; such as, cooperative adaptive cruise control, remote firmware update, e-call, and remote diagnostic of vehicles use the integration of the four networks of a connected vehicle (Subsection 2.2 describes several applications). These applications are called *smart mobility applications*: applications that use data collected from vehicles to improve the use of vehicles and the safety and comfort of drivers, and to rationalize the use of public infrastructure.

A *connected vehicle* is a vehicle whose ECUs communicate through an in-vehicle network, and it communicates with neighboring vehicles and RSUs through VANets, with personal devices through WPAN, and with Service Providers (SPs) and SCs through cellular network.[5] A connected vehicle is equipped with an on-Board Unit (OBU): a device for communicating a vehicle with other entities through VANets, WPAN, cellular network, and routing messages to/from ECUs of the vehicle.[6]

Connected vehicles enable the use of Intelligent Transportation Systems (ITSs). ITSs support the efficient and safely use of transport infrastructure and means (cars, trains, planes, ships) to facilitate the mobility of human and goods through the use of information and communication technologies [4].[7]

An attacker who aims to change the behavior of a unconnected vehicle needs to be able to physically access to its devices, or access to its communication bus, or be able to install malicious code in a device connected to the in-vehicle network. Vehicle manufacturers have overlooked security of vehicles [5]. There is a common assumption, by the vehicle manufacturers, that it is highly unlikely that potential attackers could acquire one of these capabilities. The assumption is not valid anymore because vehicles become connected to other vehicles, to personal devices, and to SCs.

Connected vehicles offer more capabilities for the attacker to compose complex attacks. An attacker could connect to the in-vehicle network of a target connected vehicle without the need for any of the capabilities listed above. For example, an attacker who has remote access to ECUs of a target

[5] We do not enumerate all (possible) communication mediums–e.g., satellite communication–for vehicles. Instead, we discuss the networks that are commonly used and reported (in the literature) to impact the security and privacy of vehicles.

[6] An OBU in general–e.g., OBU dedicated to VANets–does not act as a gateway to the in-vehicle network.

[7] Terms smart mobility and ITS are often considered similar in the literature. In this work, we use vehicle to refer to car and truck when we discuss smart mobility and all transport means when we discuss ITS.

vehicle could inject in the in-vehicle network messages to increase the speed of the vehicle.

Threats to connected vehicles have financial, privacy, and safety impacts. The description of the impact follows:

- *Financial*: A company could collect the location data of the vehicles of its competitors and extract sensitive information; such as, the address of their customers, and the performance of their fleets. The information gives the company business advantages. Other examples include: stealing pay content–e.g., infotainment, and location-based services; misuse pay-as-you-drive systems–e.g., car insurance, car taxes, car rental, and warranty or maintenance plans; and Intellectual Property (IP) theft of firmware.
- *Privacy*: A company could collect the location and in-vehicle data of drivers and use them to provide services for the drivers, and the public community. Disclosing the information, which is private information, to parties without the consent of their owners is a privacy violation.[8]
- *Safety*: A malicious attacker could remotely control the behavior of a target vehicle and inject messages in its in-vehicle network to lock its break system and make it crash with another vehicle or an obstacle. An easier attack is to mis-inform the driver.

In the last decade, several threat analysis, security solutions, and security and privacy architectures addressing a specific network type (i.e., either in-vehicle network, or VANet, or WPAN, or cellular network) have been proposed in the literature. In this chapter, we propose a taxonomy of security and privacy aspects for connected vehicles. The aspects are: security of communication links, data validity, security of devices, identity and liability, access control, and privacy of drivers and vehicles. We survey the threats to connected vehicles and classify them using the taxonomy. We survey also existing solutions that address the threats[9] and classify them using the taxonomy. We also report on the only solution for verifying security and privacy of connected vehicles that we found in the literature. The taxonomy and survey could be used by security architects to develop security solutions for smart mobility applications.

This chapter is organized as follows. Section 2 describes how connected vehicles collaborate in an ITS and describes a set of smart mobility applications. Section 3 proposes and describes a taxonomy for security and privacy aspects of connected vehicles. Section 4 surveys the security and privacy threats to connected vehicles and classifies them based on our taxonomy. Section 5 surveys the security and privacy solutions for connected vehicles and classifies them based on our taxonomy. Section 6 reports on verifying security and privacy solutions for connected vehicles. Section 7 concludes the chapter.

[8] In USA and Europe privacy violation is prohibited by the law.

[9] We survey security and privacy threats and solutions for *only* connected vehicles, which may not include all security and privacy threats and solutions for smart mobility applications or ITS.

2 Overview of the system architecture and its applications

In this section we describe an example of ITS and show how connected vehicles benefit from the system. We also describe a set of smart mobility applications that could be used by connected vehicles.

2.1 Overview of ITS and smart mobility applications

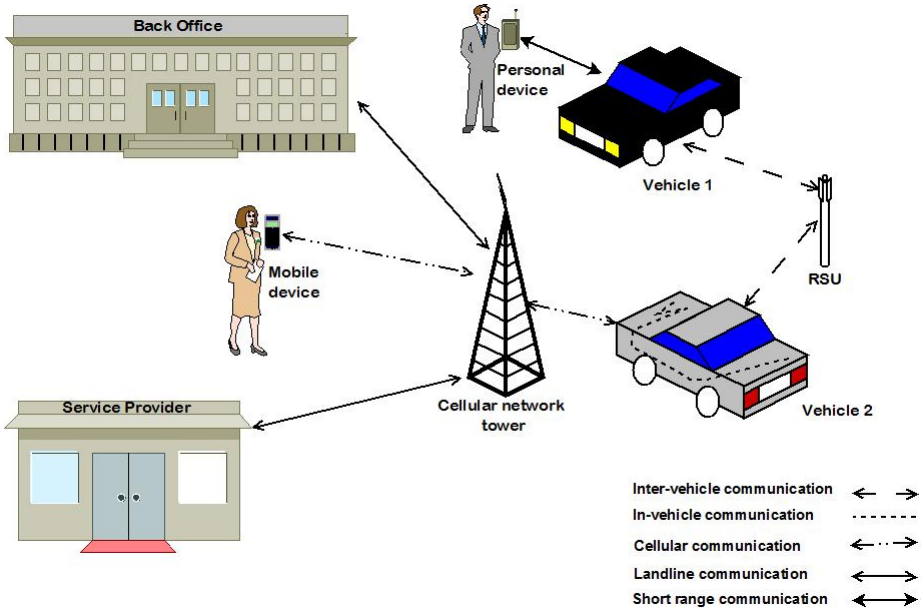

Fig. 2. Example of an Intelligent Transportation System (ITS).

Figure 2 shows an ITS composed of two connected cars, a Back Office (BO)–e.g., a fleet management system, a SP–e.g., e-call service, a personal device, a mobile device, and an RSU. Each car is equipped with ECUs that collaborate to regulate the functionality of the vehicle by exchanging messages through the in-vehicle network. For example, the vehicle dynamics control system (VDCS) of a vehicle assists the driver in over-steering, under-steering and roll-over situations: it uses the steering wheel angle and other relevant sensors data available in the in-vehicle network to decide when to apply the brakes of the individual wheels or adjust the engine torque [6].

Both cars communicate through VANet. Nodes of a VANet communicate using Wireless Access for Vehicular Environments (WAVE) [7]–a standard for inter-vehicle communication. They exchange information in order to, for

example, avoid collision. A vehicle communicates with the neighboring RSUs using WAVE to provide data about its own activities (e.g., location, speed, and heading) and get traffic conditions related to its location.

Vehicle 1 communicates with a personal device through WPAN; the driver uses his PDA to remotely start his vehicle. Vehicle 2 communicates with BO and SP through cellular network. For instance, Vehicle 2 sends data about its location to a fleet owner system, a BO, and communicates with an e-call provider [8], an SP, in case of an emergency. A second example, the car owner of Vehicle 2 uses a mobile device to inquire about the location of her car.

2.2 Smart mobility applications

Data collected from vehicles is useful for individuals, businesses, and public organizations. In the following we describe examples of smart mobility applications.

Connected vehicles could exchange messages with neighboring vehicles through VANets and adjust their speed to keep safe distance with neighboring vehicles. A *Cooperative adaptive cruise control* [9] application–of a vehicle–exchanges messages with the neighboring vehicles and sends appropriate messages to the ECUs of the vehicle to adjust its speed. Cooperative adaptive cruise control application is different from adaptive cruise control application since the latter is limited to the use of radar installed on the vehicle to detect predecessor vehicles, obstacles, and other moving objects. It computes the safe distances between the vehicle and its predecessors, and adjusts the speed of the vehicle accordingly. This application works only when predecessor vehicles are in the line of sight of the radar.

Autonomous vehicles are self-driving vehicles–i.e., robots. They sense their environment and navigate by their own. A human can choose the destination, but may not perform the driving. There are currently several prototypes of autonomous vehicles. For instance, Google [10] reported recently that it is testing a prototype of autonomous vehicle in the traffic–e.g., to drive a blind to the destination he chooses.

Connected vehicles use VANets to share information with their neighboring vehicles; such as, their location and speed. A *Cooperative collision avoidance* [11] application identifies potential collision situations and either informs the driver or acts automatically to avoid the collision. A vehicle involved in a collision, could also send notifications to other vehicles, which otherwise would drive into the crashed vehicle. This helps to avoid cascade accidents (i.e., an accident that involves many vehicles).

Companies who own fleet of vehicles use *fleet management* applications to track the locations and activities of their vehicles. Fleet managers get actual real-time information about their vehicles, including their locations, fuel levels, and speeds. The information is used for scheduling the business activities, and helps to quickly respond to events, such as, accidents.

Owners (or drivers) of vehicles equipped with personal devices; such as, PDAs and mobile phones, could use *remote vehicle control* applications to, for example, start/stop the vehicle engine, lock/unlock the doors, and monitor remotely the speed, mileage, and location of their vehicles.

Currently, there is a growing traffic congestion problem in the main cities in the world. Connected vehicles could send data about their locations and speeds to RSUs through VANets. RSUs could collect the information and use *Traffic management* applications to aggregate them to generate traffic conditions data which it disseminates to neighboring vehicles. Traffic management applications improve the traffic flow on a road by reducing congestion problem and travel time [11].

Smart mobility application	In-vehicle network	VANet	Cellular network	Personal Area Network
Cooperative adaptive cruise control	x	x		
Autonomous vehicles	x	x	x	
Cooperative collision avoidance	x	x		
Fleet management	x		x	
Remote vehicle control	x		x	x
Traffic management		x	x	

Table 1. Classification of the smart mobility applications into networks they use.

Table 1 shows for each smart mobility application, that we describe above, the networks it uses. A connected vehicle can benefit from a specific application only if it can communicate with other parties through the required network(s). (Recall that each vehicle has an in-vehicle network.) As a final note, we refer the interested readers on smart mobility applications to Kargiannis et al.[12].

Note that ITS and smart mobility applications have broader scope than connected vehicles. Figure 2 shows an ITS, which includes connected vehicles and other entities as described above. Connected vehicle is limited to one entity: a single vehicle. This chapter is limited to connected vehicle.

3 Taxonomy for security and privacy aspects of connected vehicles

We use a simple taxonomy of security and privacy aspects of connected vehicles, summarized in Figure 3, which uses 6 classes: security of communication links, data validity, security of devices, identity and liability, access control, and privacy of drivers and vehicles. The description of the aspects follows.

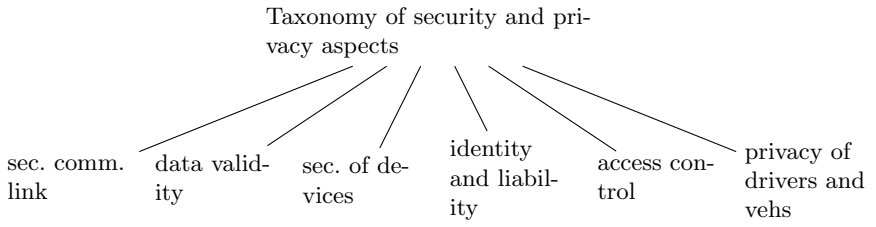

Fig. 3. Taxonomy of security and privacy aspects of connected vehicles. (We use comm. for communication or communicating, vehs. for vehicles, sec. for security, and btw. for between.)

- *Security of communication links*: They are the medium for transmitting and receiving data between two or more entities. There are four types of communications: (1) communication between ECUs, and between ECUs and OBUs through the in-vehicle network; (2) communication between vehicles, and between vehicles and RSUs through VANets; (3) communication of vehicles with personal devices through WPAN; and (4) communication between vehicles and SP and SC through cellular network.
- *Data validity*: Data are information generated, manipulated, transmitted, and received by OBUs (of vehicles), ECUs, RSUs, SP, and SCs. Data include in-vehicle data, location data, and aggregated data. (Aggregated data are computed using aggregation function; such as, average and sum, the aggregation functions use data collected by a vehicle from its neighbors.)
- *Security of devices*: Devices are electronic components (hardware) of vehicles–i.e., ECUs and OBUs. They control the behavior of vehicles. A device has two parts: (1) hardware: the ECU or OBU; and (2) firmware: programs that run on ECUs or OBUs.
- *Identity and liability*: It refers to binding an entity to a specific information or event. Identity refers to a characteristic of an entity, which distinguishes it from other entities. Liability refers to the ability to prove that a specific entity (vehicle, driver, and car owner)–or a set of entities–is responsible for a specific event or a set of events; that is, the entity cannot repudiate the responsibility for a specified event.
- *access control*: It refers to enforcing rules for access or deny to certain functions or data for identified entities.[10]
- *Privacy of drivers and vehicles*: It is the right of an entity to be able to control when, how, to what extent, and for what purpose information about themselves is shared with others (cf. [13]), and to determine the degree to which the entity will interact with its environment (cf. [14]). Privacy of drivers refers to the right of the driver to control the access and

[10] The identity of an unidentified entity is anonymous. Anonymous entities have access to public resources, if any.

use of his personal data. Privacy of vehicles refers to controlling access to the identities of communicating vehicles.

4 Taxonomy of threats to connected vehicles

In the following we describe the assets of, the attackers, and the threats to connected vehicles.

4.1 Assets of the system

An *asset* is a system resource that shall be protected by a security policy or a security mechanism [14]. It is either a component of the system or data consumed or produced by the system. Table 2 describes the assets to connected vehicle.

Asset	Description
In-vehicle data	Data extracted from the in-vehicle network; such as, speed, turning light status, and fuel level.
VANet data	Data of messages exchanged between vehicles, which may include vehicle locations. They could be used for applications, such as, collision warning, and traffic condition warning.
ECU	In-vehicle device that controls the behavior of a component of a vehicle (e.g., break, engine) and collaborate with other ECUs to regulate the behavior of the vehicle. ECUs include Global Positioning System (GPS) device: a device that provides the location of the vehicle at a specific time.
OBU	A device for communicating a vehicle with other entities through VANets, WPAN, or cellular network, and exchanging data with ECUs of the vehicle.
ECU firmware	Programs that controls the behavior of the ECU.
OBU firmware	Programs that controls the behavior of the OBU.

Table 2. Assets of connected vehicle.

A connected vehicle exchanges data with other vehicles, RSUs, personal devices, SPs, and SCs. The data types are: in-vehicle data, VANet data, ECU firmware, and OBU firmware. Table 3 provides a classification of these data types into networks–i.e., in-vehicle network, VANet, WPAN, and cellular network–they use. The goal is to show that the data could be used through several networks. This contrasts the possibility of using the data in the case of unconnected vehicles. For example, in-vehicle data is available only in the in-vehicle network for the case of unconnected vehicle. However, it could be exchanged with other parties through VANet, WPANs, and cellular network in the case of connected vehicles.

Data types	In-vehicle net.	VANet	WPAN	Cellular net.
In-vehicle data	x	x	x	x
VANet data	x	x		x
ECU firmware	x			
OBU firmware		x	x	x

Table 3. Classification of data used by connected vehicle into the networks they use. (We use "net." for network.)

4.2 Attackers and their capabilities

*Attacker*s are entities (e.g., human, robots) who intend to compromise an asset (or many assets) and have capabilities to perform threats to the system. The main attackers for connected vehicle are[15]:

- *Greedy driver*–deviates from the protocol for gains; e.g., sends data about congestion ahead, so other cars change to alternate to long routes, and frees the road.
- *Snoop*–profiles drivers through aggregating data and violating privacy of drivers.
- *Prankster*–seeking fame hacker; e.g., trick two vehicles who use a collision avoidance application in order to slow down one driver and speed up the second. Thus, the vehicles may crash.
- *Insiders*–loads malicious software on a vehicle.
- *Malicious*–attempts to cause harm; e.g., manipulate deceleration warning system to create gridlock before detonating a bomb.

An attacker could be a car owner, a driver, a mechanic, or a government officer; so, they could have one or many attacker capabilities. The capabilities could be used to potentially compromise a target asset. Table 4 lists the capabilities that an attacker uses to perform the threats on assets of connected vehicle.

4.3 Threats

*Threat*s are circumstances and events that may harm an information system through unauthorized access, destruction, disclosure, modification of data, and/or denial of service [14]. Threats are performed by attackers who compromise system's resources.

We classify threats to connected vehicles into 6 aspects: security of communication links, data validity, security of devices, identity and liability, access control, and privacy of drivers and vehicles. The classification uses the taxonomy described in the previous section.

Capability code	Attacker capability
Cap001	Physical access to ECU.
Cap002	Remote access to ECU.
Cap003	Remote control of ECU.
Cap004	Physical access to OBU.
Cap005	Remote access to OBU.
Cap006	Remote control of OBU.
Cap007	Update of ECU firmware.
Cap008	Update of OBU firmware.
Cap009	Access the communication link between ECUs.
Cap010	Control of the communication link between ECUs.
Cap011	Access to the communication link between two vehicles.
Cap012	Control of the communication link between two vehicles.
Cap013	Access to the communication link between a vehicle and the SC.
Cap014	Control of the communication link between a vehicle and the SC.
Cap015	Access to the communication link between a vehicle and a personal device.
Cap016	Control of the communication link between a vehicle and a personal device.

Table 4. Possible capabilities of attacker to a connected vehicle.

Threats to the communication links

Connected vehicles and RSUs exchange messages through VANets; ECUs of a vehicle exchange in-vehicle data through the in-vehicle network; a vehicle exchange data with personal devices through WPAN, and with SPs and SC through cellular network. The threats to messages exchanged between the communicating parties in the four networks are similar. We describe in the following the main threats (cf. [15]).

A snoop could **eavesdrop the communication between two parties** without their consent. The two parties could be, for example, two devices of a same vehicle, two vehicles, or a vehicle and an SP or SC. For example, an attacker installs a mobile phone spy software (e.g., [16]) on the OBU of his victim's vehicle and gets periodically the data available in the OBU.

A vehicle of a greedy driver, who is required to forward messages it receives to other parties, could **drop the messages** (a.k.a. message suppression) it receives. For example, a prankster may receive congestion alerts from neighboring vehicles, but, does not forward them to its neighbors (i.e., close by vehicles) [15]; they will have to wait in the traffic.

A prankster could **fabricate messages** (i.e., create false messages) and send them to other vehicles or devices of the vehicle [5]. For example, a greedy driver makes his car pose as emergency vehicle and sends alert messages to its neighbors. The attack allows the driver to speed up their own trip [15].

A malicious attacker could **alter the messages** being exchanged between two parties (after it intercepts them). The attacker may receive messages from one party, change their content, and forward them to the other communicating party. For example, a greedy vehicle receives messages about the improvement of traffic conditions. It changes the messages to worsening traffic conditions and broadcasts them to its neighbors.

A malicious attacker may receive messages from close by vehicles. They could perform **replay attack** by changing the time stamp of the messages and broadcasting them multiple times to close by vehicles. For example, a malicious attacker could intercept a message broadcasted by a vehicle and replay it to jam the communication channels of close by vehicles.

Threats to data validity

Smart mobility applications rely on the *validity* of the data (the data are correct–not false, and accurate) they use. False data, sent by malicious vehicles, could have severe safety impacts such as accidents. In the following, we describe a set of threats to data validity (a.k.a. data falsification).

The attacker disseminates **bogus data** (i.e., false data) in the network to affect the behavior of other vehicles [17]. For instance, an attacker could send information about bad traffic conditions. Vehicles receiving the messages try to find alternate paths; thus, the attacker frees the path for themself.

An attacker could **cheat in aggregating data** they receive from their neighbors. For example, in traffic management application, a vehicle collects information from its neighbors about the count of vehicles in their proximities–e.g., in 300 meter radius. The data shall be used to compute the count of vehicles in a wider area–e.g., 2 km road length. The attacker sends false aggregate data instead of the data it computes using the data it receives from its neighbors.

An attacker who controls two communicating vehicles can tunnel packets broadcasted in one location to another, thus, disseminating wrong information: they perform a **Wormhole attack** on the VANet [17]. For example the attacker could tunnel traffic information messages from a vehicle in a crowded zone to vehicles in another zone. This misleads traffic management application which collects the information and broadcasts it to other vehicles.

Threats to devices

An attacker who aim to compromise the data used by a smart mobility application (i.e., vehicle location, in-vehicle data, or VANets data) through (unauthorized) altering the behavior of the devices (GPS devices, ECUs, or OBU) of a target vehicle, needs to either have physical access, or remote access to the device. The description of the main threats ECUs and OBUs follows.

A malicious attacker that has physical access to a device can **tamper the device hardware** by modifying or deactivating physical sensors, memories,

and hardware functionalities. For example, an attacker changes circuits of a device to extract sensitive data, such as, cryptographic keys.

A malicious attacker that has physical access to a device can **tamper the device software** by modifying its firmware. For example, an attacker modifies the firmware of a device to send false data to other vehicles.

A malicious attacker who has access to the devices of a target vehicle could **replace a device** of the vehicle with a compromised device to, for example, prevent it from working properly. For example, a mechanic–insider attacker–replaces a night vision device of a military vehicle so it does not work properly [18].

A malicious attacker injects dummy messages or jam the communication channel to stop the communication between a vehicle and other parties; and prevent important messages from reaching the vehicle [17]; that is, perform a **Denial of Service (DoS)** attack. For example, a malicious attacker provokes an accident and performs a DoS attack to prevent the appropriate deceleration warnings from reaching other drivers; they create a cascade crash on the highway [15].

A malicious attacker could inject messages to **remotely control the vehicle**. For example, an attacker installs malicious code on OBU accessible through cellular network and remotely start the vehicle.

An attacker could change the software installed on a device. They change the device behavior by **tampering the device firmware**. For example, an attacker can change the firmware of his OBU, so the vehicle does not send speed exceeding a specified limit. The information is, for example, used by the car insurance of the attacker to evaluate their driving behavior.

An attacker could perform an **unauthorized over the air update of the device** by installing remotely their own software on the device. For instance, a malicious attacker could attack the Over-the-Air (OTA) diagnostic and firmware update procedure; they install their own software on the device [19].

Threats to identity and liability of vehicles

In the following we describe a set of threats to identity and liability of vehicles. Members of VANets: vehicles and RSUs; members of in-vehicle networks: ECUs, GPS, and an OBU; members of WPAN: personal devices and an OBU; members of cellular networks: Handheld Devices (HHD), SPs, and SC, use identity information when communicating. Attackers may exploit the information to perform attacks. In the following we describe the main threats.

A malicious attacker uses identity information of another vehicle to impersonate (i.e., pretend to be) it [17]; that is, perform a **masquerade attack**. For example, a vehicle could impersonate another vehicle and sends information, such as, its own location, to the fleet owner application.

A prankster or a malicious attacker could use multiple identities to deceive the communicating parties; that is, perform a **Sybil attack**. For example, a

malicious vehicle pretends to be multiple other vehicles. Reported data from the vehicle appears to arrive from a large number of distinct vehicles, and hence it is considered credible [20].

An attacker sends information to an entity; but, performs **repudiation attack**: denies responsibility of sending specific messages in the case of a probe. For example, an attacker sends emergency vehicle warnings, so they could bypass other vehicles, but, deny the act in the case of a probe.

Threats to access control

An attacker could modify data or copy code of OBUs or ECUs: they perform **unauthorized access** to data and code. For example, a malicious attacker, who wants to reduce his insurance premium, could modify the aggregated data collected and computed by the OBU of their vehicle, before sending them to pay-as-you-drive insurance service.

Threats to privacy of vehicles and drivers

The main threats for the privacy of vehicles and drivers are location privacy and driving behavior privacy. A snoop could perform a **location privacy attack** through collecting time series data about the location of a vehicle. The data could be obtained from messages the vehicle broadcasts in VANets, or from messages the vehicle sends to SC through cellular network. For instance, a competing company could spy on its competitor by tracking the location of its vehicles. The information allows the company to extract information about the customers of the competitor.

An attacker could violate the **driving behavior privacy** of a driver by collecting time series data about the use of a driver of his vehicle, and use the data to infer and extract private information. For example, the police gets access to a database of in-vehicle data collected by an SC and extracts information about drivers who have speed violations (a.k.a. big brother threat).

5 Taxonomy of security and privacy solutions for connected vehicles

In the following we describe a set of solutions that we found in the literature that address threats to connected vehicles.

5.1 Solutions to assure secure communication between parties

This subsection describes solutions for secure communication in in-vehicle network, secure communication in VANets, secure communication between vehicles and personal devices, and secure communication between vehicles and SPs or SC.

Solutions for secure communication in in-vehicle network

Wolf et al. [21] propose to secure in-vehicle communications between ECUs by a kind of hybrid encryption scheme, which combines symmetric and asymmetric encryption. Thus, for efficiency reasons, communication authenticity and confidentiality between ECUs can be ensured by a symmetric cipher together with a shared secret key. The secure distribution of the shared key to new ECUs can be done by an asymmetric encryption scheme, which checks a new ECU for validate certificates, for instance, before securely handing over the key. In order to extend this scheme to different in-vehicle networks executed in parallel (e.g., Controller Area Network (CAN) and Local Interconnect Network (LIN)), it can make reuse of the central gateway, which already translates the messages between different in-vehicle networks. Thus, the central gateway handles also the decryption and re-encryption of messages of different in-vehicle networks with different shared keys. Moreover, the central gateway can also act as powerful entity for adding and removing new ECUs by running the asymmetric ECU authentication. The security mechanisms they propose–for secure communication–are:

- *ECU authentication*: Each ECU sends its certificate and identity information to the gateway which authenticates the ECU; the gateway verifies whether the certificate of the ECU is signed by an authorized Original Equipment Manufacturer (OEM). An authenticated ECU receives a symmetric key, which it uses to encrypt messages it broadcasts in the in-vehicle network.
- *Message encryption*: An ECU uses the symmetric key it gets from the authentication process to communicate with other ECUs. It is not practical, and not safe, to use asymmetric encryption schemes to encrypt all messages because of the time and capacity constraints of the in-vehicle network. For example, a vehicle could crash with neighboring vehicles, if the duration required for the break engine to receive the request from the driver and decrypt it exceeds a safety delay.
- *Gateway firewalls*: The certificate of an ECU includes an authorization code, used by the gateway of the in-vehicle network, to authorize the ECU to communicate with ECUs from another bus network.

A first practical realization of the proposed scheme has been done by the EVITA project(cf. [22]).

Bar-El [23] developed a framework for security services for in-vehicle network[11], which consists of a set of modules installed on the ECUs and several applications that use the modules. The Framework includes the following components: a tool to generate and distribute session keys[12] for ECUs; a library for encrypting and authenticating messages exchanged between ECUs; a central

[11] The authors use the term intra-vehicle network to refer to in-vehicle network.

[12] Session keys are symmetric keys shared by communicating parties and are valid for a duration of the communication.

secure storage for sensitive information; a library to assure that each ECU, is (upon the boot) about to execute a code that was approved and intended to be executed; and a tool to provide a trusted time. The applications that are included in the framework are: secure ECUs code update enabler, secure logging enabler, authorization enabler, theft prevention enabler, and secure feature activation enabler.

Solutions for secure communication in VANets

In the following, we describe IEEE1609.2 standard [24] and research solutions proposed for securing the communication between nodes of a VANet. IEEE 1609.2 Standard addresses the threats to message eavesdropping, modification, and replay; and minimizes unauthorized identity disclosures. The standard defines the format and processing of secure messages, and a digital certificate (a document binding a public key to an entity described by the identity information, such as, name and address) format to limit the use of bandwidth.

The standard specifies 4 security and privacy services for the network and applications that run over WAVE, which are:

- confidentiality–assuring that only the recipients of the massage can read it;
- authenticity–confirming of origin of the message;
- integrity–assuring that the message has not been changed while in transit between parties;
- anonymity–broadcasted message should not leak information to unauthorized recipients identifying the vehicle originating the messages.[13]

The standard recommends using Elliptic Curve Integrated Encryption Scheme (ECIES) algorithm to encrypt messages exchanged between vehicles and Elliptic Curve Digital Signature Algorithm (ECDSA) to sign the messages.[14] The minimum recommended size of encryption keys is 256 bits, of signature keys is 224, and of the Certification Authority (CA) keys is 256 bits. (CA is an entity that issues digital certificates. A digital certificate certifies the ownership of a public and private key pair to the subject of the certificate) It recommends the use of Advanced Encryption Standard (AES)-Counter with Cipher Block Chaining Message Authentication Code (CBC-MAC) (CCM), as a bulk encryption algorithm, for authenticating-then-encrypting messages they exchange. The standard recommends minimizing exchange of data that uniquely identify the recipient or would allow an unauthorized recipient to identify the sender of the messages.

Now, we summarize some research solutions for secure communication in VANets. Kargl et al. [25], Papadimitratos et al. [26], and Raya and

[13] This definition of anonymity is specific to the standard.

[14] Encryption assures confidentiality of messages. Digital signature assures integrity and authenticity of messages.

Hubaux [27, 17] propose the use of digital certificate to assure integrity of messages and authenticate vehicles in the context of European research project SeVeCom [28]. They propose secure group communication[15], key bootstrapping, and key revocation approaches. Secure communication approach assures confidentiality of messages exchanged between a group of vehicles. Key bootstrapping is the process of creating private keys and related digital certificate for vehicles, and for setting up the vehicles with the keys and certificates. Key revocation is the process of disabling the use of specified certificates.

Papadimitratos et al. [26] evaluate the security properties required by cooperative driving applications[16] and found that most of the applications require authentication, and integrity, but not confidentiality.[17]

Note that The European Telecommunications Standards Institute (ETSI) is currently working on standards for protecting data exchanged between vehicles, and between vehicles and RSUs in VANets[29].

Solution for secure communication between vehicles and personal devices

An in-vehicle WPAN is composed of a set of personal devices (users' short range wireless devices) and an in-Vehicle Access Point (iVAP): a gateway between the in-vehicle network and the personal devices. The iVAP could be the OBU.

Mahmud and Shanker [2] propose an architecture for in-vehicle secure WPAN and a mechanism for generating and distributing secret keys for personal devices. The in-vehicle secure WPAN is composed of a coordinator, which is the iVAP, and personal devices. To join the network, a personal device D_i registers with the iVAP using a trusted link (i.e., a man-in-the-middle attacker cannot eavesdrop the link) which is either a peer-to-peer wired link, e.g., Universal Serial Bus (USB), or Very Short Range (i.e., few centimeters) Wireless (VSRW) link, e.g., IR and NFC. The key distribution approach is adopted by NIST for distributing Bluetooth link keys in its revised guide for Bluetooth security[30].

The iVAP prompts the user for the (authorization) password upon connecting a new personal device D_i to the iVAP. If the authorization succeeds, the iVAP generates n secret keys[18] K_{ij} and sends them to the device, which stores them for future use. The iVAP (resp. personal device D_i) maintains a counter C_i that stores the number of times it communicate with personal device D_i(resp. iVAP); that is, the number of communication sessions. Each

[15] Secure group communication is communication between a group whose members share a secret key, which they use to encrypt messages they exchange.

[16] A cooperative driving application allows a set of vehicles, members of a VANet, to coordinate their actions by exchanging data through a VANet.

[17] Confidentiality is required by few applications; such as, platooning: vehicles follow each other such that they do not collide.

[18] Value of n may depend on the memory of the device.

time D_i communicates with the iVAP, it uses the secret key K_{ij}, where index j is the value C_i; then, it removes the key from its store. iVAP sends new n secret keys to D_i when they reach the n^{th} communication session. To revoke a key k_{ij}, iVAP needs only to remove it from its store.

Devices composing a WPAN, in most of the cases, communicate using the Bluetooth technology. The two encryption algorithms supported by Bluetooth are: E_0 and Advanced Encryption Standard-Counter with Cipher Block Chaining Message Authentication Code (CBC-MAC) (AES-CCM) [31]. Note that, Lu et al. found an attack that can recover the encryption key used by algorithm E_0 in 2^{38} computations [32].

Solutions for secure communication between vehicles and service centers

We describe in this subsection the use of Secure Sockets Layer (SSL) protocol and use of Wireless Transport Layer Security (WTLS) protocol for communicating vehicles with SCs.[19]

We discuss in the following SSL [33] protocol. The protocol is used by applications to assure secure transport communication between two entities. It uses a reliable transport protocol, such as, Transport Control Protocol (TCP) [34]. (A transport protocol is reliable if it assures ordering of received messages, removing of duplicate messages, and preventing loss of messages.) The protocol assures confidentiality, integrity, and authenticity of messages. It has 4 sub-protocols:

- *Handshake protocol*: Two parties use the protocol to create a secure communication session. It is a sequence of steps that allows a server and a client to authenticate each other, to negotiate a cipher-spec., and to share keys before sending or receiving data. The cipher-spec. lists the cryptographic specifications that both parties agree on: key agreement algorithm, symmetric encryption algorithm, Message Authentication Code (MAC) algorithm, and length of the secret keys.[20]
- *Application data protocol*: Application data are fragmented, compressed, and protected using the encryption and MAC algorithms defined in the current cipher-spec.
- *Change cipher-spec. protocol*: It signals a change of the cipher-spec. that both parties will use; both parties could send a change cipher-spec. message to notify the other party that subsequent messages will be protected under the just-negotiated cipher-spec.

[19] In this section we use mobile device to refer to OBU that has mobile device communication and processing capabilities. The reason is that the work that we refer to is about mobile device and not explicitly OBU; but, in general it applies to OBU.

[20] The reader may consult Katz and Lindell [31] for background on cryptography–if needed.

- *Alert protocol*: It signals a warning or error during the handshake and up to the closure of the session. A party that sends a fatal error message to its partner closes immediately the session, after sending this record. The message gives a reason for the session closure. A party receiving a warning message decides whether to close the session or the keep it.

OBU has limited bandwidth (rate of sending data), processing speed, and memory size. The device shows a security processing[21] gap if the processing, memory, and bandwidth capabilities of the OBU are smaller than the security processing requirements [35]. Security processing gap causes loss of data that the OBU is required to send to other parties, but does not.

There are several issues that limit the use of SSL by mobile devices. For instance, SSL requires the use of a reliable transport protocol, such as, TCP; however, mobile devices use also unreliable transport protocol User Datagram Protocol (UDP) for Short Message Service (SMS). Also, mobile devices have limited bandwidth, and limited processing and memory capabilities. Furthermore, there are restrictions on exporting and using strong cryptography outside USA.

WTLS [36] protocol was created to address the issues related to the use of SSL by mobile devices. The protocol is designed to provide confidentiality and integrity of messages, and authentication of communicating applications [36]. The protocol specification is similar to SSL; it provides the same functionalities and has the same sub-protocols: Handshake protocol, Application protocol, Change cipher-spec. protocol, and Alert protocol.

WTLS protocol addresses the issue of the need to support unreliable transport protocol, such as, UDP. WTLS messages include the sequence number mode and key refresh rate, in plain text. The sequence number mode indicates the scheme used to communicate the sequence numbers (e.g., send in plain text). The information is used by the receiving party to enforce reliability of messages, if needed. The refresh rate indicates the frequency of updating the cryptographic specifications used by the communicating parties.

The protocol addresses the issues of bandwidth, processing, and memory limitations through the following measures: compress the application data, use of a digital certificate format shorter than X509.x format [37][22], truncate the MAC of messages [38], use of weak cryptographic algorithms, and use of short secret keys[38].

Most of the changes to SSL made WTLS vulnerable to several attacks [38]. For instance, the use of unauthenticated alert messages (messages sent through the Alert protocol) allows truncation attack: remove arbitrary packets from the data stream without the detection of the receiving party. Saarinen [39] describes 8 possible attacks on WTLS protocol. Note that some of these at-

[21] Security processing is computations for the purpose of security, such as, data encryption and signature.

[22] WTLS supports both its own digital certificate format and X509.x format [37].

tacks are inherent to the protocol, while other are related to the use of weak cryptographic algorithms, or use of short keys.

5.2 Solutions for detecting false data

We discuss in this subsection solutions for detecting malicious data, wormhole attacks, and for secure data aggregation for VANets.

Detecting malicious data

Nodes use sensors to collect in-vehicle data and location data. They share the information with their neighbors, which may propagate it to their neighbors. Nodes collect data from their neighbors which could be somehow redundant.[23] Golle et al. [40] propose a model for detecting malicious data based on: assuming that data obtained from several source about a fact (e.g., the location of a vehicle or his speed) is redundant; assuming that an attack involving few malicious nodes is more likely than an attack that requires collision of a large number of nodes; and use of physical rules, when available; such as, a node can have only one location at a time.

Protection against wormhole attack

A wormhole attack on VANet causes a vehicle, member of a VANet, to use data *of* vehicles that are not its neighbors. Su and Boppana [41] propose Neighbor Verification by Overhearing (NEVO): a protocol for verifying the neighbors of any vehicle. The protocol is a sequence of three messages between two neighbor nodes i and j. First, node i broadcasts a message probe query (PQ). Second, after node j receives the PQ from node i, it rebroadcasts the message as its probe forward (PF). Third, node j sends node i a probe reply (PR) message, which contains the processing delay, δ. Next, node i verifies if node j is its neighbor; it checks if equation $t_{oh} - tx_j - \delta \leq 2R/s_p$ holds, where t_{oh} is the time delay for node i to overhear node j forwarding its own PQ message, tx_j is the transmission time for the forwarded message by node j, R is the radio signal propagation range, and s_p is the radio signal propagation speed.

There are two main issues with using NEVO protocol in VANet. First, the solution assumes that the processing delay, δ, is small compared to propagation and transmission delays. A node j could deceive node i by sending it a false value for δ that passes the test, and not the true delay, which may not pass the test. Second, the protocol does not consider the mobility of vehicles. Vehicles mobility requires considering the time of taking the measurements and the life time of the measurements.

[23] The difference between the reported locations of a vehicle by two other vehicles is smaller than an error margin.

Shokri et al. [42] propose another protocol for verifying the neighbors of any node of a network by verifying consistency of the information collected by neighboring vehicles. The protocol consists of three phases. In the first phase, ranging, every node of the network exchanges messages with its neighbors to evaluate its own distance to its neighbors. It creates a neighbors table that includes the computed distances.

In the second phase, neighbors table exchange, every node shares with its neighbors its own neighbors table. Next, each node creates a table that includes the distances between its neighbors, in addition to its own distance to its neighbors.

In the third phase, link verification, each node A performs three consistency tests to verify whether a reported neighbor is a true neighbor or not. The first test, link symmetry test, uses the integrated neighbors table to test whether the distance from node i to node j is similar to the distance from node j to node i.[24] The second test, maximum range test, checks if the distance between two neighbors is less than the maximum range of the radio transmission of vehicles. The third test, quadrilateral test, checks for every neighbor B of a node A, if there are two nodes C and D, such that, (A, B, C, D) is a 4-clique (the four nodes are neighbors to each other), and is a convex.[25]

This protocol has the same issue as the previous one; it does not consider the mobility of vehicles.

Secure data aggregation for VANets

Several smart mobility applications, such as, co-operative detection of traffic jams, require the collection of information from big set of vehicles, e.g., vehicles running in a distance of 10 km. These applications require that each vehicle aggregates the information it receives from its neighbors (combine the information using an aggregation function; such as, average and sum) and disseminates it in larger area [43]. However, the integrity of aggregated information cannot be easily verified.

Dietzel et al. [43] propose an approach for secure aggregation, where each vehicle strategically selects signed atomic reports (i.e., data that are not the result of an aggregation function) from other vehicles and computes a trust in the correctness of the aggregated data. First, the aggregate area is divided into n squares. Then, the vehicle selects reports of vehicles located at the borders of the aggregate area, and n reports, each is an arbitrary report of a vehicle from each square. (The length of the sides of the squares, which defines

[24] The similarity test considers an error threshold in checking the equality of the computed distances.

[25] Quadrilateral is a polygon with four edges and four vertices or corners. Quadrilateral (A, B, C, D) is convex if the product of the cross products $(\overrightarrow{AB} \times \overrightarrow{BC})(\overrightarrow{BC} \times \overrightarrow{CD})(\overrightarrow{CD} \times \overrightarrow{DA})(\overrightarrow{DA} \times \overrightarrow{AB})$ is positive.

the number of squares n, affects the granularity of the aggregate data.) Next, the reports are aggregated. The vehicle uses fuzzy reasoning [44] to compute a value describing the quality of selected atomic reports. The output value is a crisp value (e.g., very good, good, acceptable, bad, very bad) indicating the trust of the vehicle in the aggregate value.

5.3 Solutions for tampering with devices

In the following we describe four solutions that address devices tampering threats. The solutions are: use of devices that include a Hardware Security Module (HSM), a virtualization platform for devices, a secure OTA firmware update, and a solution for DoS attack on OBU. Note that there are several other work that advocate the use of tamper resistance devices but did not provide solutions, e.g., [45].

Use of devices that include hardware security module

The objective of project EVITA [46] is to design, verify, and prototype architecture for in-vehicle network, such that, security-relevant components are protected against tampering, and sensitive data are protected against compromise. The partners developed three HSMs [47]:

- full HSM protects high-performance ECUs (e.g., OBU, central gateway) by providing a high-performance symmetric and asymmetric cryptographic engine for efficient in-vehicle and VANet communications,
- medium HSM is forseen to protect some central multi-purpose ECUs (e.g., engine control, immobilizer) and hence is similar to full HSM except it has less processing performance and does not have a hardware asymmetric cryptographic engine,
- light HSM secures communication between small but critical ECUs, sensors, and actuators of a vehicle (e.g., pedal sensors, brake actuators, GPS or clock controller) through the use of a symmetric cryptographic engine.

The partners developed also software components for HSM and ECUs and integrated their libraries with AUTOSAR. AUTOSAR is an open and standardized automotive software architecture for ECUs used by automobile manufacturers, suppliers and tool developers [48].

For secure update and regular verification of the device's firmware (i.e., secure boot) and some limited counterfeit protection, there already exits a cost-efficient automotive-capable Security Hardware Extension (SHE). SHE is an official OEM-controlled specification[49] and is already available from several dedicated semiconductor manufacturers.

Over-the-Air firmware update

Companies managing smart mobility applications and vehicle manufacturers use OTA firmware update to fix bugs and improve the functioning of their software installed in OBUs and ECUs. Idrees et al. [50] propose a secure protocol for OTA firmware update for the devices of a vehicle. The main steps of the protocol are:

1. The service station remotely diagnoses the ECUs of the vehicle.
2. The service station sends a request for approving the updates of the firmware of a set of ECUs to the owner of the vehicle human-machine interface.[26]
3. For each ECU, the protocol terminates if the customer denies the request; otherwise, it proceeds with the following steps:
   a) The station switches the ECU from application mode to reprogramming mode.
   b) The service station requests from the OEM the key for decrypting the firmware of the ECU and gets the key encrypted using the public key of the ECU, which it forwards to the ECU.
   c) The ECU downloads the firmware, encrypted with the key that it received from the OEM, from the service station in parts. The size of each part does not exceed the maximum length of the messages of the in-vehicle network.
   d) The ECU verifies the authenticity of the firmware.[27] It proceeds with installing the firmware, deleting the old firmware, and switching the ECU back to application mode if the verification succeeds. It terminates otherwise.

The protocol could be used by, for example, by service stations.

Protection against denial of service attack

IEEE 1609.2 standard proposes the use of ECDSA for signing messages exchanged between vehicles in VANet. However, the verification of a single ECDSA signature requires *7ms* of computation on the typical OBU proposed by the standard [51]. Attacker can create and send invalid signatures in very short time. We call *arrival time*, the average duration between the arrival of two messages to the OBU, and *processing time*, the average time required by the OBU to verify the signature of received messages. A DoS attacker could exploit the difference between processing time and arrival time to compromise the availability of the OBU: flood the OBU with invalid messages.

[26] The human-machine interface could be for example a mobile phone, or a device installed in the vehicle.

[27] It checks if a computed ECU Configuration Register (ECR) (a signature of the firmware) is equal to the reference ECR received from the OEM.

Studer et al. [51] propose a hybrid authentication mechanism named VANET Authentication using Signature and TELSA++ (VAST). The mechanism uses digital signature algorithm ECDSA and TELSA++, an extension for Timed Efficient Stream Loss-tolerant Authentication protocol (TELSA). TELSA++ uses symmetric cryptography with delayed and periodic key disclosure for authenticating messages. The signature could be used, when needed, for non-repudiation of messages, e.g., in case the driver must be alerted about the received message.

5.4 Solutions for vehicle identity and liability

Hubaux et al. propose Electronic License Plates (ELPs) [45] to identify vehicles. An ELP could be a digital certificate granted by authorized governmental agency. Example of use of ELPs includes paying toll roads: vehicles send their ELPs and other required information to an automated toll collection system. ELPs could also be used to authenticate vehicles in parkings.

Wolf [52] describes another general, holistic approach for designing, securing, implementing, and applying an ELP for various existing and upcoming vehicular application scenarios.

5.5 Solution for access control

The project Open VEhiculaR SEcurE platform (OVERSEE) [53] develops an open, standardized in-vehicle software and communication platform, which enables sharing of the platform′s computing resources and its internal (e.g., in-vehicle network) and external communication capabilities (e.g., Wireless fidelity (Wi-fi), cellular, VANet) for vehicular applications, while assuring mutual isolation and strong access control of applications, platform resources, services, communication interfaces, and data.

This approach empowers also third parties to develop, download and install vehicular applications, while OVERSEE ensures that such third party applications cannot harm each other or any other in-vehicle IT system regardless whether an application malfunction is caused by an application failure (IT safety) or an application vulnerability exploited trough a hacker or a computer virus (IT security).

The underlying protection approach is based on virtualization together with a strong access control mechanism and other central security functionalities (e.g., encryption, digital signature), which in turn are protected by an automotive-capable hardware security anchor (i.e., use of EVITA HSM).

5.6 Solutions for privacy protection

We describe in this subsection a set of solutions for protecting privacy of vehicles in VANets and privacy of drivers whose data are collected by SCs.

Privacy of vehicles in VANets

Several smart mobility applications require authenticating messages exchanged between collaborating vehicles; such as, cooperative collision avoidance. Papadimitratos et al. [26] propose message authentication using pseudonyms (fictitious names used to perform a particular role) to protect their privacy, instead of the identities of communicating vehicles. The description of the solution follows. Each vehicle generates a set of key pairs and sends the public keys to a corresponding CA. The CA generates a pseudonym for each key pair, signs each key pair, to produce certificates, and sends the certificates to the vehicle. The vehicle uses the key pairs to authenticate its messages sequentially. It uses one pair for a period of time, discards it, and uses the next key pair. The vehicle, while using a set of certificates, can require the next set of signed certificates from the CA. Changing pseudonyms makes it difficult for an adversary to link messages from the same vehicles and track its movements.

Privacy of drivers whose data are collected by service centers

Project Privacy Enabled Capability in Co-operative Systems and Safety Applications (PRECIOSA)[54] aims to design an architecture for policy enforcement in Cooperative Intelligent Transport Systems (CITSs). The architecture protects privacy of drivers, whose data are collected and used by SCs. Thus, a user associates with his data a set of policies for using the data. The policy includes entities that access the data (i.e., processors), purposes of using the data, retention period of data, and authorized operations–e.g., averaging, summation, and atomic reading. Kargl et al. [55] developed a privacy-enforcing runtime architecture for assuring enforcement of privacy policy in the trusted domain (a domain composed of hosts that are trusted to enforce the policies.

Kung et al. [56] advocate privacy-by-design approach for protecting privacy of drivers and vehicles[28] The principles of privacy by design are:

- data minimization–the collection of personal data should be strict minimum;
- privacy enforcement–the operations of applications that use the personal data should provide maximum protection of data;
- transparency–the application should provide the stakeholders the way it ensure protection of the private data.

The authors discuss also a software engineering process that implements privacy-be-design principles.

Kost et al [57] propose the use of an ITS privacy ontologies[29] to evaluate privacy violations of a model, representing an ITS application. That is, check

[28] The paper addresses ITS applications which, as we discussed in Section-sec:overview includes connected vehicles.

[29] An ontology is a representation of knowledge as a set of concepts and the relationships between them in a specific domain.

the compliance of the model with privacy constraints–e.g., minimization of the use, access, and collection of personal data.

6 Approaches for verifying the security properties of connected vehicles

Manual verification of security and privacy solutions for a connected vehicle is difficult because of the complexity of the network used by the vehicle. An attacker, who could remotely connect to a vehicle–e.g., through a cellular network–can inject messages in the in-vehicle bus (the link connecting the vehicle's ECUs) of the victims′ vehicle. They do not need to physically connect to the in-vehicle bus to inject messages–a capability required to attack an unconnected vehicle [5].

Verifying the security and privacy properties of a connected vehicle requires the use of an automated mechanism. The only work on the subject that we found in the literature is Automated Verification of reAl Time softwARe (AVATAR)[58]: an environment for modeling and verifying real-time embedded systems. It is implemented by TTool [59]. TTool is an open source, general purpose modeling language for systems engineering applications supporting the specification, analysis, design, verification, and validation of a broad range of complex systems, which may include hardware, software, information, processes, personnel, and facilities. TTool supports Systems Modeling Language (SysML) [60].

First, a user models the (to be verified) system using TTool. The model is a set of SysML diagrams extended with annotations describing the implementation of the security solution and the security properties that the system shall assure (e.g., a secret key is confidential and accessible to only communicating parties, not the attacker).

Next, the user activates the security verification of the model–they click the appropriate button of TTool. AVATAR uses the security verification toolkit *ProVerif* [61]. TTool translates SysML diagrams, including the security properties, to the specification language of ProVerif. ProVerif verifies whether the security property is satisfied or not. If a security requirement is not satisfied, TTool provides traces that show how an attacker could exploit the system and compromise the security property.

Pedroza et al.[19] use AVATAR to verify Firmware Update (FU) protocol, developed by Idrees et al. [50]. They modeled the system using SysML diagrams, specified the security properties, and verified several security requirements including the confidentiality of the communication between the OEM, source of firmware update, and the ECU destination of the update. The experiment is a real use case of AVATAR.

Although AVATAR authors claim that it is intended to verify embedded system, TTool has been used only to verify the security of protocols (e.g.,

communication protocol, and key distribution protocol). ProVerif uses Dolev-Yao [62] attack model which focuses on the security of the communication between parties; Dolev-Yao attack model does not include attacks; such as, controlling the behavior of one of the legitimate parties.

7 Summary

A *connected vehicle* is a vehicle whose ECUs communicate through an in-vehicle network; and it communicates with neighboring vehicles and RSUs through VANets, with personal devices through WPAN, and with SCs and SPs through cellular network. An attacker on connected vehicles has more capabilities to perform threats than an attacker on unconnected vehicles. Developing security and privacy solutions for smart mobility applications (applications that use information generated by vehicles; e.g., cooperative adaptive cruise control) requires considering threats to the system that integrates in-vehicle network, VANets, WPANs, and cellular network.

This work proposes a taxonomy for security and privacy aspects of connected vehicles, which are: security of communication links, data validity, security of devices, identity and liability, access control, and privacy of drivers and vehicles. It describes and classifies–based on the taxonomy–the threats to connected vehicles, describes and classifies the solutions proposed in the literature that address the threats, and reports on the only approach (that we found in the literature) for verifying security and privacy in connected vehicles.

The goal of the survey is to provide security architects of smart mobility applications with an initial repository of threats to connected vehicles and solutions that mitigate the threats. We believe that, currently, there are no "strong" solutions for verifying the security and privacy architecture of smart mobility applications.

Acknowledgement

This work is supported by the Dutch national HTAS innovation program; HTAS being an acronym for High Tech Automotive Systems. More information on this innovation program is accessible via the document `http://www.htas.nl/files/pdf%20bestanden/HTAS_Innovatie_Programma_-_september_2007[2].pdf`. Any opinions expressed in this Chapter are those of the authors and do not necessarily reflect those of Dutch national HTAS innovation program.

The authors thank Dr. Arno Spinner, from The Federal Highway Research Institute (BASt), Germany, and Pelin Anguin, from Purdue University, for providing valuable comments on an earlier draft of this book chapter.

References

1. R. Brooks, S. Sander, J. Deng, and J. Taiber, "Automobile security concerns," *IEEE Vehicular Technology Magazine*, vol. 4, no. 2, pp. 52–64, June 2009.
2. S. Mahmud and S. Shanker, "In-vehicle secure wireless personal area network (swpan)," *IEEE Transactions on Vehicular Technology*, vol. 55, no. 3, pp. 1051 –1061, May 2006.
3. J. Zhang and I. Stojmenovic, *Handbook on Security.* Wiley, Dec. 2005, vol. Volume I, Part 2, ch. Cellular networks, pp. 654–663.
4. *Intelligent Transport Systems (ITS) Communications Architecture*, The European Telecommunications Standards Institute (ETSI) Std. ETSI EN 302 665, Rev. V1.1.1, 09 2010. [Online]. Available: http://webapp.etsi.org/workprogram/Report_WorkItem.asp?WKI_ID=28554
5. K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *Proc. of IEEE Symposium on Security and Privacy*, San Diego, CA, May 2010, pp. 447–462.
6. K. H. Johansson, M. Torngren, and L. Nielsen, "Vehicle applications of controller area network," in *Handbook of Networked and Embedded Control Systems.* Springer, 2005, pp. 741–765.
7. R. Uzcategui and G. Acosta-Marum, "Wave: A tutorial," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 126 –133, May 2009.
8. (2011, June) ecall: Time saved = lives saved. [Online]. Available: http://ec.europa.eu/information_society/activities/esafety/ecall/index_en.htm
9. B. van Arem, C. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 429 –436, Dec. 2006.
10. J. Markoff. Google cars drive themselves, in traffic. [Online]. Available: http://www.nytimes.com/2010/10/10/science/10google.html?_r=1&hp=&pagewanted=all
11. M. Kihl, *Vehicular Networks Techniques, Standards, and Applications.* Auerbach Publications, 2009, ch. Vehicular Network Applications and Services, pp. 21–39.
12. G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *Communications Surveys Tutorials, IEEE*, vol. 13, no. 4, pp. 584 –616, quarter 2011.
13. A. Westin, *Privacy and freedom.* New York: Atheneum., 1967.
14. R. Shirey, "Internet Security Glossary, Version 2," RFC 4949 (Informational), Aug. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4949.txt
15. B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Workshop on Hot Topics in Networks (HotNets-IV)*, College Park, MD, Nov. 2005. [Online]. Available: http://sparrow.ece.cmu.edu/~parno/pubs/vehicles.pdf
16. (2012, Apr.) Mobile phone spy cell phone monitoring and tracking system. [Online]. Available: http://www.mobilephonespyx.com/
17. M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, Jan. 2007.

18. M. Wolf, A. Weimerskirch, and T. J. Wollinger, "State of the art: Embedding security in vehicles," *EURASIP Journal of Embedded Systems*, vol. 2007, June 2007.
19. G. Pedroza, M. Idrees, L. Apvrille, and Y. Roudier, "A formal methodology applied to secure over-the-air automotive applications," in *Vehicular Technology Conference (VTC Fall), 2011 IEEE*, Sept. 2011, pp. 1 –5.
20. T. Zhou, R. R. Choudhury, P. Ning, and K. Chakrabarty, "Privacy-preserving detection of sybil attacks in vehicular ad hoc networks," in *The 4th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, Philadelphia, PA, Aug. 2007, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/MOBIQ.2007.4451013
21. M. Wolf, A. Weimerskirch, and C. Paar, "Security in automotive bus systems," in *Workshop on Embedded Security in Cars (escar)04*, Bochum, Germany, Nov. 2004.
22. H. Schweppe, S. Idrees, Y. Roudier, B. Weyl, R. E. Khayari, O. Henniger, D. Scheuermann, G. Pedroza, L. Apvrille, H. Seudie, H. Platzdasch, and M. Sall, "Deliverable d3.3: Secure on-board protocols specification," Tech. Rep., July 2011. [Online]. Available: http://evita-project.org/Deliverables/EVITAD3.3.pdf
23. H. Bar-El, "Intra-vehicle information security framework," in *Proc. of the 7th ESCAR Embedded Security in Cars Conference*, Dsseldorf, Germany, Nov. 2009.
24. *IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages*, IEEE Std., July 2006. [Online]. Available: http://ieeexplore.ieee.org/servlet/opac?punumber=11000
25. F. Kargl, P. Papadimitratos, L. Buttyan, M. Müter, B. Wiedersheim E. Schoch, T.-V. Thong, G. Calandriello, A. Held, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: Implementation, performance, and research challenges," *IEEE Communcations Magazine*, vol. 46, no. 11, pp. 110–118, Nov. 2008.
26. P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: Design and architecture," *IEEE Communcations Magazine*, vol. 46, no. 11, pp. 100–109, Nov. 2008.
27. M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *The Third ACM Workshop on Security of Ad Hoc and Sensor Networks*, ser. SASN '05, Alexandria, VA, Nov. 2005, pp. 11–21.
28. (2011, June) Secure vehicle communication. [Online]. Available: http://www.sevecom.org/Pages/Publications.html
29. S. Randall and S.-H. Houmb, "Experience in developing standards for cooperative systems," June 2012, Workshop Personal Data Protection and Security Aspects Related to its Applications, Brussels, Belgium.
30. J. Padgette, K. Scarfone, and L. Chen, "Guide to bluetooth security : recommendations of the national institute of standards and technology," National Institute of Standards and Technology (U.S.), 2010.
31. J. Katz and Y. Lindell, *Introduction to Modern Cryptography*. Chapman & Hall/CRC, 2007.
32. Y. Lu, W. Meier, and S. Vaudenay, "The conditional correlation attack: a practical attack on bluetooth encryption," in *The 25th annual international*

conference on Advances in Cryptology, ser. CRYPTO'05. Santa Barbara, CA: Springer-Verlag, Aug. 2005, pp. 97–117. [Online]. Available: http://dx.doi.org/10.1007/11535218_7

33. A. Freier, P. Karlton, and P. Kocher, *The Secure Sockets Layer (SSL) Protocol Version 3.0*, Internet Engineering Task Force (IETF) Std., Aug. 2011. [Online]. Available: http://tools.ietf.org/html/rfc6101
34. J. Postel, *Transmission Control Protocol*, Std. RFC793, Sep. 1981. [Online]. Available: http://tools.ietf.org/html/rfc793
35. S. Ravi, A. Raghunathan, P. Kocher, and S. Hattangady, "Security in embedded systems: Design challenges," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 3, pp. 461–491., Aug. 2004.
36. L. Wireless Application Protocol Forum, *Wireless Transport Layer Security*, Std., 2001. [Online]. Available: http://www.openmobilealliance.org/wapdocs/wap-261-wtls-20010406-a.pdf
37. R. Housley, W. Ford, W. Polk, and D. Solo, *Internet X.509 Public Key Infrastructure Certificate and CRL Profile*, Std. rfc2459, Jan. 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2459.txt
38. S. Jormalainen and J. Laine. (1999, Nov.) Security in WTLS. [Online]. Available: http://www.hut.fi/jtlaine2/wtls/
39. M.-J. O. Saarinen, "Attacks against the WAP WTLS protocol," in *The IFIP TC6/TC11 Joint Working Conference on Secure Information Networks: Communications and Multimedia Security*, Leuven, Belgium, Sep. 1999, pp. 209–215. [Online]. Available: http://dl.acm.org/citation.cfm?id=647800.736984
40. P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proc. of the 1st ACM international workshop on Vehicular ad hoc networks*, ser. VANET '04, Philadelphia, PA, oct. 2004, pp. 29–37. [Online]. Available: http://doi.acm.org/10.1145/1023875.1023881
41. X. Su and R. Boppana, "Mitigating wormhole attacks using passive monitoring in mobile ad hoc networks," in *IEEE Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.*, New Orleans, LO, Dec. 2008, pp. 1 –5.
42. R. Shokri, M. Poturalski, G. Ravot, P. Papadimitratos, and J.-P. Hubaux, "A practical secure neighbor verification protocol for wireless sensor networks," in *Proc. of the second ACM conference on Wireless network security*, ser. WiSec '09, New York, NY, USA, 2009, pp. 193–200. [Online]. Available: http://doi.acm.org/10.1145/1514274.1514302
43. S. Dietzel, E. Schoch, B. Könings, M. Weber, and F. Kargl, "Resilient secure aggregation for vehicular networks," *Netwrk. Mag. of Global Internetwkg.*, vol. 24, no. 1, pp. 26–31, Jan. 2010. [Online]. Available: http://dx.doi.org/10.1109/MNET.2010.5395780
44. L. A. Zadeh, "Fuzzy logic and approximate reasoning," *Synthese*, vol. 30, pp. 407–428, 1975, 10.1007/BF00485052. [Online]. Available: http://dx.doi.org/10.1007/BF00485052
45. J. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security Privacy*, vol. 2, no. 3, pp. 49 –55, May-June 2004.
46. (2012, May) Evita project: E-safety vehicle intrusion protected applications, european commission research grant fp7-ict-224275. [Online]. Available: www.evita-project.org
47. L. Apvrille, R. E. Khayari, O. Henniger, Y. Roudier, H. Schweppe, H. Seudié, B. Weyl, and M. Wolf, "Secure automotive on-board electronics network ar-

chitecture," in *FISITA 2010 World Automotive Congress*, Budapest, Hungary, May-June 2010.
48. (2012, May) Autosar. [Online]. Available: http://www.autosar.org/
49. Hersteller Initiative Software - Security working group, "SHE–functional specification v1.1, rev 439," Oct. 2009.
50. M. S. Idrees, H. Schweppe, Y. Roudier, M. Wolf, D. Scheuermann, and O. Henniger, "Secure automotive on-board protocols: a case of over-the-air firmware updates," in *Proc. of the Third international conference on Communication technologies for vehicles*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 224–238. [Online]. Available: http://dl.acm.org/citation.cfm?id=1987310.1987333
51. A. Studer, F. Bai, B. Bellur, and A. Perrig, "Flexible, extensible, and efficient vanet authentication," *Journal of Communications and Networks*, vol. 11, no. 6, pp. 574–588, Dec. 2009.
52. M. Wolf, "A secure and privacy-preserving electronic license plate," in *Automotive Safety & Security*, Stuttgart, Germany, June 21 23 2010.
53. (2012, June) Oversee. [Online]. Available: https://www.oversee-project.com/
54. (2012, May) Preciosa–privacy enabled capability in co-operative systems and safety applications. [Online]. Available: http://www.preciosa-project.org/
55. F. Kargl, F. Schaub, and S. Dietzel, "Mandatory enforcement of privacy policies using trusted computing principles." in *AAAI Spring Symposium: Intelligent Information Privacy Management*, Stanford, CA, Mar. 2010.
56. A. Kung, J. Freytag, and F. Kargl, "Privacy-by-design in ITS applications," in *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM),*, Lucca, Italy, June 2011, pp. 1 –6.
57. M. Kost, J.-C. Freytag, F. Kargl, and A. Kung, "Privacy verification using ontologies," in *First International Workshop on Privacy by Design*, Vienna, Austria, Aug. 2011, pp. 627 – 632.
58. G. Pedroza, L. Apvrille, and D. Knorreck, "AVATAR: A SysML environment for the formal verification of safety and security properties," in *11th Annual International Conference on New Technologies of Distributed Systems (NOTERE)*, Paris, France, Mar. 2011, pp. 1 –10.
59. TTool - an open-source UML and SysML toolkit. [Online]. Available: http://ttool.telecom-paristech.fr/
60. Object Management Group, Inc. (OMG). (2010, June) OMG systems modeling language (OMG SysML). [Online]. Available: http://www.sysml.org/docs/specs/OMGSysML-v1.2-10-06-02.pdf
61. B. Blanchet, "Automatic verification of correspondences for security protocols," *J. Comput. Secur.*, vol. 17, no. 4, pp. 363–434, Dec. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1576303.1576304
62. D. Dolev and A. C. Yao, "On the security of public key protocols," Stanford, CA, USA, Tech. Rep., 1981.