RESEARCH

CARAMEL: Results on a Secure Architecture for Connected and Autonomous Vehicles Detecting GPS Spoofing Attacks

Christian Vitale[1*], Nikos Piperigkos[3,4], Christos Laoudias[1], Georgios Ellinas[1,2], Jordi Casademont[5,6], Josep Escrig[5], Andreas Kloukiniotis[4], Aris S Lalos[3,4], Konstantinos Moustakas[4], Rodrigo Diaz Rodriguez[7], Daniel Baños[8], Gemma Roqueta Crusats[8], Petros Kapsalas[9], Klaus-Peter Hofmann[10] and Pouria Sayyad Khodashenas[5]

[*]Correspondence: vitale.christian@ucy.ac.cy
[1]KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, Cyprus
Full list of author information is available at the end of the article

Abstract

The main goal of the H2020-CARAMEL project is to enhance the protection of modern vehicles against cybersecurity threats related to automated driving, smart charging of Electric Vehicles, and communication among vehicles or between vehicles and the roadside infrastructure. This work focuses on the latter and presents the CARAMEL architecture aiming at assessing the integrity of the information transmitted by vehicles, as well as at improving the security and privacy of communication for connected and autonomous driving. The proposed architecture includes: (i) multi-radio access technology capabilities, with simultaneous 802.11p and LTE-Uu support, enabled by the connectivity infrastructure; (ii) a MEC platform, where, among others, algorithms for detecting attacks are implemented; (iii) an intelligent On-Board Unit with anti-hacking features inside the vehicle; (iv) a Public Key Infrastructure that validates in real-time the integrity of vehicle's data transmissions. As an indicative application, the interaction between the entities of the CARAMEL architecture is showcased in case of a GPS spoofing attack scenario.

Keywords: Connected autonomous vehicles; Secure architecture; Attack on V2X communication; GPS spoofing attack

1 Introduction

The damaging effects of cyberattacks to an industry like the Cooperative Connected and Automated Mobility (CCAM) can be tremendous. From the least to the most important one, it is possible to mention the damage in the reputation of vehicle manufacturers, the increased denial of customers to adopt CCAM, the loss of working hours (having direct impact on the countries GDP), increased environmental pollution due to, e.g., traffic jams or malicious modifications in sensors' firmware, and, finally, the great danger for human lives, either they are drivers, passengers or pedestrians. The goal of the H2020-CARAMEL project[1] is to proactively address modern vehicle cybersecurity challenges applying, among others, advanced Artificial Intelligence (AI) and Machine Learning (ML) techniques, and to seek methods to mitigate associated safety risks.

[1]https://www.h2020caramel.eu/

To address cybersecurity considerations for the already here Connected and Autonomous Vehicles (CAVs), well established methodologies originating from the Information and Communications Technology (ICT) sector will be adopted, allowing to assess vulnerabilities and the impacts of potential cyberattacks. Although past initiatives and cybersecurity projects related to the automotive industry have improved security for networked vehicles, several newly introduced technological dimensions like 5G, autopilots, and smart charging of Electric Vehicles (EVs) introduce cybersecurity gaps that have, as of yet, not been addressed satisfactorily [1]. Considering the entire supply chain of automotive operations, CARAMEL aims at delivering commercial anti-hacking Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) for automotive cybersecurity and to demonstrate their value through extensive attack and penetration scenarios. Specifically, CARAMEL focuses on three main types of attacks: (i) attacks on the AI of autonomous vehicles: computer vision and AI techniques are crucial for vehicle self-driving and environment understanding; (ii) attacks on the electric vehicle charging infrastructure: the rise in adoption of EVs is gaining momentum and the misuse of the charging infrastructure could have effects on the national and international energy sustainability; (iii) attacks on the communication infrastructure underlying the CCAM, which could impair the overall system performance.

One of the three pillars of CARAMEL is the focus of this paper, i.e., the CCAM secure connectivity infrastructure. Table 1 summarizes the most used acronyms hereinafter. Section 1.1 overviews the attacks that a connectivity infrastructure may face, with the corresponding state-of-the-art countermeasures. The following sections describe instead the solution adopted in CARAMEL. In Section 2, the secure connectivity architecture envisioned in CARAMEL is presented. The three attacks taken into consideration for demonstration in CARAMEL are outlined in Section 3. Then, in Section 4 the interactions among different entities in the proposed architecture are exemplified in a Global Positioning System (GPS) location spoofing attack scenario and an effective mitigation technique is described. Finally, Section 5 concludes the paper.

AI	Artificial Intelligence	EV	Electic Vehicle
ML	Machine Learning	AV	Autonomous Vehicle
CCAM	Cooperative Connected and Automated Mobility	ITS	Intelligent Transport System
PKI	Public Key Infrastructure	CRL	Certificate Revocation List
RCA	Root Certification Authority	EA	Enrollment Authority
AA	Authorization Authority	VA	Validation Authority
AT	Authorization Tickets	EC	Enrollment Credentials
V2X	Vehicle to Everything	V2V	Vehicle to Vehicle
MEC	Multi-Access Edge Computing	UE	User Equipment
LTE	Long Term Evolution	eNB	evolved NodeB
LTE-Uu	Basic LTE interface UE/eNB	D2D	Device to Device
vEPC	virtual Evolved Packet Core Network	RSU	Road Side Unit
VLAN	Virtual Local Area Network	CAM	Cooperative Awareness Message
DENM	Distributed Event Notification Message	BTP	Basic Transport Protocol
GN	GeoNetworking	OBU	On-Board Unit
HSM	Hardware Security Module	IVN	In-Vehicle Network
CAN	Control Area Network	GPS	Global Positioning System
GNSS	Global Navigation Satellite System	EKF	Extended Kalman Filter
SoO	Signal of Opportunity	MLE	Maximum Likelihood Estimation
RTCL-MLE	Robust Traditional Collaborative Localization - MLE	ROC	Receiver Operating Characteristic
RGCL	Robust Graph-based Collaborative Localization	AUC	Area Under Curve

Table 1 List of acronyms used in the paper.

1.1 Attacks on CCAM Connectivity Infrastructures

In this section, the potential threats and vulnerabilities that may be encountered by a CCAM connectivity infrastructure are presented (with available state-of-the-art countermeasures). Based on [1], the attacks can be classified into four general categories: (i) Authenticity/Identification attacks; (ii) Availability attacks; (iii) Confidentiality/Privacy attacks; and (iv) Data integrity/Data trust attacks.

1.1.1 Authenticity/Identification Attacks

Authenticity and secure entities identification is a prime requirement in Autonomous Vehicles (AV) networking to ensure the protection of the legitimate entities against several attacks.

- Sybil attack: A malicious vehicle pretends to be legitimate by exploiting fake identities. Authenticated nodes consider the malicious messages to be legitimate and cannot detect the attackers. Cryptography schemes can be adopted as a countermeasure [2];
- Location Service Jamming and Spoofing: Global Navigation Satellite Systems (GNSS), e.g., the GPS, are vulnerable to attacks where legitimate satellite signals are either blocked or counterfeited. An effective solution for detecting the location spoofing attack is introduced in [3] and presented in detail in Section 4;

1.1.2 Availability Attacks

Availability is crucial to ensure the safety of the involved drivers and vehicles.

- Denial of Service (DoS) attack: Aims at preventing legitimate entities from accessing the network services and resources. Access control with packet filtering is the recommended mitigation technique [4];
- Timing attack: A transmission is delayed by adding extra timeslots between received messages. Authenticated timing methods are effective against these types of attacks [5];
- Flooding and Jamming attack: Focuses on disrupting the network communication channels. Channel switching is the adopted countermeasure solution [6].

1.1.3 Confidentiality/Privacy Attacks

Contrary to the previous attacks, confidentiality and privacy attacks do not affect safety. Nevertheless, sensitive information exchanged in the network, e.g., locations of the AVs, Intelligent Transportation System (ITS) safety messages, and drivers' personal information should be protected.

- Eavesdropping attack: Attempts to steal information (e.g., location) by snooping on the communication channel. Although it is easy to carry out, secure communication can be used to prevent this attack [7];
- Interception attack: Starts by listening to the network for some time and then trying to analyze the data to extract useful information. Privacy-preserving methods can be adopted to mitigate this attack [8].

1.1.4 Data Integrity/Data Trust Attacks

Data must be intact and unchanged throughout their lifecycle. The attackers could easily alter the data or falsify data exchanged among vehicles and/or the infrastructure.

- Replay attack: Previously generated data are maliciously repeated; as a countermeasure, duplicated data can be prevented by making use of the sequence number, time-stamp and secure communication [9];
- Data alteration/Data injection attack: Intentionally modified data are injected in the network of vehicles. Signature of transmitted packets [10], as well as convex optimization approaches that exploit special structures related to spatio-temporal correlations and sparsity characteristics [11], can be used as a countermeasure.

2 The CARAMEL Architecture

The CARAMEL project's objective is to propose a secure environment for autonomous and connected vehicles. As part of this objective, CARAMEL aims at improving the security, enhancing the privacy, and increasing the resilience of the adopted communication infrastructure. For this task, existing state-of-the-art solutions for Multi-Radio Access Technology (Multi-RAT) Vehicle-to-Everything (V2X) communication infrastructure is improved with novel ML algorithms running both at the vehicle, in the so-called On-Board Unit (OBU), and at the network edge, i.e., at the Multi-access Edge Computing (MEC) platform. The implemented ML algorithms allow CARAMEL to keep track of the integrity of the entities in the system and of the information transmitted. A Public Key Infrastructure (PKI) is used to register and authorize all vehicles' data transmissions and to intervene when problems are detected, e.g., by updating or canceling distributed certificates. The different building blocks constituting the CARAMEL's infrastructure, also shown in Figure 1, are unveiled in the following sections: (i) Section 2.1 presents the entities included in the PKI and their main functionalities; (ii) Section 2.2 showcases the adopted communication infrastructure, with some preliminary implementation details and (iii) Sections 2.3-2.4, respectively, introduce the on-board telecommunication unit, i.e., the OBU, and the device hosting in-vehicle secure ML algorithms, i.e., the anti-hacking device.

2.1 The CARAMEL's PKI

The PKI enables the provision of secure V2X message transmissions and will be the basis to the certificate management of vehicles. It comprises basically of five different entities:

- the Root Certification Authority (RCA): This entity contains the root certificates for the entire PKI. For security reasons, this is an offline entity which must be managed only by authorized personnel;
- the Online Certification Authority (OCA): This is an online entity signed by the RCA. Its main responsibility is to sign the different lower authorities in the PKI;
- the Enrolment Authority (EA): This entity is in charge of providing the necessary credentials at the enrolment phase, which are used afterwards by the car to ask for Authorization Tickets (ATs), also known as pseudonym certificates;

- the Authorization Authority (AA): This entity provides the ATs, which are issued for ensuring privacy of the car communications within the ITS infrastructure;
- the Validation Authority (VA): This entity provides a way to ask about the revoked certificates. It maintains the Certificate Revocation List (CRL) including the revoked certificates, along with an online service that returns the state of a specific certificate in real-time.

The interaction between ITS nodes and the PKI follows two successive phases, namely enrolment and authorization. During the enrolment phase, an ITS node, e.g., an AV, requests Enrolment Credentials (ECs) to an EA such that it can be trusted by other ITS nodes. To obtain the enrolment certificate, the AV sends the Bootstrap Certificate (BC) which is a provisional self-signed certificate containing the Canonical ID and a Public Key. Once validated, the EA generates a unique EC for this ITS node which will be required in the next phase. In the authorization phase, an enrolled ITS node requests the ATs to an AA to get specific permissions, ensuring confidentiality and privacy. This request includes the ECs obtained in the previous phase. Internally, the AA asks the VA to validate the credentials provided to proceed with the authorization. Finally, EA and AA can be trusted by the ITS node through validating their authenticity with the RCA. Now, the ITS node is able to securely communicate with other nodes and/or the MEC server by using the AT obtained as a result of this process. Overall, the PKI enables the provision of secure V2X message transmissions and is the basis of the real-time certificate management of vehicles (see Section 4.2).

2.2 The Multi-RAT V2X Communication Infrastructure with MEC Functionalities

As of the late 2020, there is not a clear radio technology to be used for V2X communications. Up to now, IEEE 802.11p (also known as Direct Short Range Communications - DSRC) has been the de facto wireless technology standard for V2X communications. It is a relatively mature technology and has already been validated by over a decade of field trials. Despite that, IEEE 802.11p, which uses Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA), suffers from a high level of collisions under heavy traffic conditions, mainly due to hidden terminal situations. Long-Term Evolution-based V2X (LTE-V2X), from the Third Generation Partnership Project (3GPP), is a relatively new alternative solution to the IEEE 802.11p-based V2X communications. The first version of LTE-V2X or Cellular-V2X (C-V2X) included numerous enhancements to the existing Device-to-Device (D2D) protocol to accommodate vehicular communications. These enhancements include a new arrangement of the resource grid of the physical layer and two types of D2D channel access mechanisms: (i) a mechanism coordinated by the evolved NodeB (eNB), named Mode 3, and (ii) a distributed mechanism, where User Equipments (UEs) access the channel on their own, named Mode 4. Moreover, LTE-V2X employs different radio interfaces: (i) an interface between the vehicle and the eNB, named LTE-Uu, and (ii) an interface between vehicles, named LTE-PC5. Additionally, both technologies continue being enhanced by IEEE, with the 802.11bd, and by 3GPP, with the 5G New Radio technology (NR-V2X), respectively.

The fact is that, currently, ITS stakeholders do not have a single technological option to choose. Some vehicle manufacturers have already began to distribute

vehicles with IEEE 802.11p and shortly, there will be other cars equipped with LTE-V2X (PC5). Moreover, during this transition period, many vehicles will be equipped only with a basic cellular connection 4G (LTE-Uu) or 5G, mainly used to provide Internet connection to their occupants but that potentially could also be used for V2X messages as well. This raises the problem that vehicles using different radio technologies will not be able to communicate directly between them. Apart from supporting the operations of the PKI architecture to secure the V2X communications, the objective of CARAMEL is to support such interoperability.

Specifically, CARAMEL aims at creating an architecture that allows communication between vehicles equipped with IEEE 802.11p, which works in the Control Channel (CCH) of the ITS-G5 band (5,9 GHz), and vehicles equipped with V2X technologies over a basic LTE-Uu connection working in the operator's band. Barring the possibility of having vehicles equipped with multiple technologies, the solution adopted in CARAMEL is to relay on functions performed by the fixed network to implement V2I2V (Vehicle-to-Infrastructure-to-Vehicle) communications. Due to the strict end-to-end delay restrictions required by some Cooperative Intelligent Transport System (C-ITS) applications, interoperability between technologies is implemented using infrastructure support through the use of MEC. Furthermore, most V2X messages used by ITS applications, as for instance the basic Cooperative Awareness Message (CAM) or Decentralized Environment Notification Message (DENM), are sent in broadcast mode, expecting all neighboring vehicles to receive them. Therefore, the target of CARAMEL is also to enable messages transmitted from one vehicle to reach all other vehicles in the same area and, if necessary, vehicles in nearby areas.

The proposed architecture is shown in Figure 2 . Firstly, it comprises of the OBUs deployed in vehicles, which are equipped with LTE-Uu transceivers, enabling IP connections to the Internet and to the servers hosted in the MEC, and, in some cases, with an additional IEEE 802.11p network card for direct V2V/V2X communication. In CARAMEL, V2X communications are based on the ETSI ITS architecture, with protocols Geonetworking (GN) at the network layer and Basic Transport Protocol (BTP) at the transport layer. Geonetworking traffic is always transmitted over 802.11p if the OBU includes an 802.11p interface, and over LTE-Uu if it does not. CARAMEL implements the ETSI ITS protocol architecture through a modification of the open source framework Vanetza [12]. Therefore, if the transmitter and the receivers use 802.11p and are under coverage, broadcast communications are performed directly thanks to the intrinsic broadcast nature of the 802.11p but, in any other case, meaning that transmitters or receivers need to use LTE-Uu or they are not under coverage, a message forwarding function in the MEC is used.

In order for vehicles to reach the forwarding server and, in general, for all cases where a vehicle needs to reach other servers hosted in the fixed infrastructure, two types of radio access stations are deployed: (i) the so-called Road Side Unit (RSU), similar to a WiFi access point, that acts as a forwarder between an 802.11p radio interface and an Ethernet interface; (ii) the standard LTE eNB of small format, also named Small Cell. Both radio access technologies are connected to the MEC using a Virtual Local Area Network (VLAN) capable Ethernet switch. In the MEC also resides the Virtual Evolved Packet Core (vEPC) used for LTE core cellular network functions. Due to the proposed architecture, CARAMEL adopts the following

conceptual model for V2X communications: (i) a hardware technology-dependent radio transceiver and (ii) a software implementation of the upper layers of the protocol architecture (the modified Vanetza framework in the CARAMEL system). Focusing on the previous mentioned fixed radio stations, these two objects could be implemented in the same physical element or in two different entities. As the CARAMEL's objective is to enable vehicles having only LTE-Uu connections to be able to transmit and receive V2X messages, and, since deploying the ETSI ITS protocol architecture in all eNBs of all cellular operators is unfeasible, this module runs as a software instance directly in the MEC. In addition, it is also reasonable to have the same solution for RSUs, decoupling the 802.11p radio transmitter of the RSUs from its corresponding software for upper protocol layers (Vanetza), which will also run in the MEC. This approach for the RSUs enables to deploy very simple and light RSUs and centralize all computation demanding modules in the MEC. As previously mentioned, in case an OBU is additionally provided with an 802.11p interface, the V2X messages are transmitted and received directly through this second interface. Nevertheless, LTE-Uu only OBUs do not have this option. The solution taken by CARAMEL is to establish a layer 2 tunnel over the IP connection provided by the LTE-Uu interface, that starts in the OBU and finishes in the virtual container of the MEC that hosts the Vanetza associated to the small cells. The endpoints of this tunnel are seen as virtual layer 2 interfaces, and Vanetza modules situated at both ends can directly transmit and receive over it.

Taking all this into consideration, the MEC hosts different virtual containers:

- One Vanetza entity for each RSU.
- One single Vanetza entity for all LTE-Uu only OBUs, which attends the endpoints of the tunnels created by them.
- One vEPC that, altogether with the small cells, constitute an LTE system. This module is connected to the Internet through the Internet interface of the MEC, and provides Internet connectivity to all OBUs.
- The V2X forwarding module that receives all V2X incoming messages, analyzes them, and decides if they need to be forwarded to other vehicles depending on their radio technology, area of interest, age of the messages, content of the message, etc. It also enables to inject V2X messages from external servers to the system of fixed radio stations to be received by vehicles. This module is connected to the Internet through the Internet interface of the MEC.
- One module called *Register* to provide LTE broadcast transmissions. Although the LTE standard defines the evolved Multimedia Broadcast Multicast Services (eMBMS), it is not widely deployed. Therefore, to cope with this issue, CARAMEL deploys the *Register* module that registers all LTE-Uu only OBUs in the system, and each time that one V2X message needs to be broadcasted, it transmits one unicast copy to each one of them. The registration process of a new LTE-Uu only OBU is automatically done whenever it receives a V2X message from a vehicle that enters the region of operation for the fist time. The unregistration process is performed when the *Register* stops receiving V2X messages from a vehicle for some specified amount of time. This process is extremely costly in terms of bandwidth but, if the eMBMS or the LTE-PC5 are not operational, converting one LTE broadcast transmission

in multiple LTE unicast transmissions is the only solution to reach all the intended receivers.
- Applications to remotely connect to RSUs and to small cells in order to manage them. It can be as simple as an *ssh*.
- Other processes, e.g., ML algorithms used to improve the overall security of autonomous and connected vehicles.

The final step of the communications architecture is sharing the physical Ethernet interface of the MEC among the different virtual containers in the MEC, some of which require network interfaces configured as layer 2 and others as layer 3. The chosen solution is to create different virtual Ethernet interfaces, one for each virtual container, associated to the same physical interface and split the Ethernet network in the following VLANs:

- One VLAN to connect each pair formed by one RSU with the virtual container that hosts its associated Vanetza. Both ends of the VLAN are configured as layer 2 interfaces.
- One VLAN that connects all small cells (eNBs) and the vEPC of the LTE system. All interfaces of this VLAN are configured as IP interfaces forming an IP network. This network constitutes the interface S1-U of the LTE system and small cells can be reached through this network to control them.
- One VLAN that connects all RSUs with the MEC to be able to access and control them. All interfaces of this VLAN are configured as IP interfaces forming an IP network.

All the remaining containers can reach the OBUs only through the corresponding network radio access stations, hence through the containers in the MEC implementing the dedicated communication protocol stacks (either LTE-Uu or 802.11p).

2.3 The Vehicle's On-Board Unit

An OBU is the telecommunication unit embedded in the standard cooperative vehicles and provides secure communication functionalities. One of the goals of CARAMEL is to develop a completely functional and secure OBU that provides the hardware for secure V2X communications. The OBU security features are enhanced by the so-called "Anti-Hacking Device" that is in charge of detecting malicious attacks and functional misbehavior using pre-trained ML models. The OBU architecture, shown in Figure 3, includes the following main elements:

- A Hardware Security Module (HSM): One of the possible attack vectors to V2X infrastructure is to steal sensitive data or cryptographic keys from a vehicle's OBU. To counter this attack, trustworthy, unforgeable, and non-copyable identities must be established. This is achieved by integrating an HSM into the OBU that serves as a secure storage for private key data, security certificates, and even generic sensitive data. The HSM is responsible for enabling secure communication of V2X applications by protecting the integrity of exchanged safety messages and managing authentication of V2X participants. The HSM, among others, also manages private key generation, derivation, and deletion in case of attack.
- Security applications: This element contains all software functions to interact with the PKI and manage the registration and authorization procedures, as

well as to obtain the pseudonymous ATs and store them into the HSM according to [13]. Additionally, this element also controls in real-time the CRL, so as to account for unreliable message reception.

- ITS Applications: This element represents any ITS application running on the vehicle. The CARAMEL testbed foresees applications for sending and receiving CAMs and DENMs.
- A V2X Communication Protocol Architecture: This element contains the software package that enables the OBU (and the MEC) to generate Facilities layer messages encapsulated on the BTP and the GN protocol. CARAMEL will use the open source framework Vanetza [12], properly extended to perform all security and privacy related functionalities.
- Network Radio interfaces (IEEE 802.11p and LTE-Uu): Radio interfaces are used in CARAMEL for three purposes: i) for connecting OBUs to the PKI servers to obtain the pseudonymous ATs before being able to transmit ITS messages and for real-time management of certificates (for this purpose, LTE-Uu is used); ii) to notify the management center or the MEC when the anti-hacking device detects that the vehicle is under attack (also for this purpose, LTE-Uu is used); iii) for data transmission between vehicles. To reduce latency during ITS message transmission, these communications preferably use direct V2V connections through the 802.11p interface. Nevertheless, as previously mentioned, in the first stages of ITS adoption, not all vehicles will be equipped with this technology. Some cars will only have the LTE-Uu interface and forwarding/message broadcasting will be performed with the assistance of the MEC.
- In-Vehicle Network (IVN) Interfaces: The OBU is equipped with several communication interfaces that enable networking capabilities within the vehicle. This is part of the IVN interface and includes: a 1000Base-T1 Ethernet interface, which defines Gigabit Ethernet over a single twisted pair for automotive and industrial applications; a WiFi interface, compliant with IEEE802.11a/b/g/n/ac, 5G MIMO and Real Simultaneous Dual Band (RSDB); a Controller Area Network (CAN) bus interface.
- Hardware Secure Elements: These elements are included to protect the OBU from tamper attacks, through box opening detection, active hardware protection of susceptible signals, and environmental sensors to prevent fault injection attacks. When the Hardware Secure Elements detect an attack, there is a tamper response and the system is enabled to protect sensible data. Logical methods are also used to prevent firmware manipulation. In order to comply with these security functional requirements, several tamper protection layers have been applied on the different OBU interfaces (Figure 4) based on hardware actuations. More insight about this is given in Section 3.2.3.
- An Anti-hacking Device: This is a a physical controller that is integrated into the car and acts as an attack detection device extending the security capabilities of the OBU. The device passively listens to the internal buses (e.g., CAN or Automotive Ethernet) and extracts the raw sensor data, which is used by pre-trained ML algorithms to detect anomalies that might point out malicious attacks. The device receives ITS messages sent by the OBU

and performs the functions for, e.g., countering potential location spoofing attacks or renewing used ATs to minimize the possibility of being tracked by attackers. For further details see Section 2.4 below.

2.4 The Anti-hacking Device

The anti-hacking device, which represents an important part of the CARAMEL's innovation, is a physical controller integrated into a vehicle that acts as an attack detection device. Even if its role is crucial when validating the vehicle's message transmissions, the objective of the CARAMEL's anti-hacking device is broader. Indeed, its task is to run pre-trained ML models that are also able to detect anomalies on sensor data.

The anti-hacking device is connected to the busses in the car carrying the sensor data. It passively monitors the bus traffic and extracts the raw sensor data. Figure 5 shows the ML pipeline where raw data, e.g.,from the CAN bus, is pre-filtered and aggregated to make it suitable for the following machine learning stage that detects threats and attacks. Any security-relevant event is then forwarded to the visualization and mitigation components in the car. The ML knowledge base (the models) is pre-loaded into the anti-hacking device after being created offline on a more powerful system based on simulated and real-world training data.

Figure 6 shows an overview of the software and hardware architecture of the anti-hacking device. While initially the anti-hacking device is implemented using a Coral Dev Board hardware (together with a solution for development and simulation – the USB Accelerator), more powerful hardware solutions, such as the NVidia Jetson AGX board, are also considered. From bottom-up the following components make up the anti-hacking device:

- HW Interfaces: The anti-hacking device is connected to the in-car systems via appropriate interfaces used in the automotive industry, including the CAN bus, Automotive Ethernet connections, and also Wireless connectivity (Wi-Fi and Bluetooth). For integration into development and simulation frameworks standard Ethernet is also supported.
- ML hardware: Since the anti-hacking device is based on the Coral Dev Board, the Tensorflow Lite Processing Unit (TPU) is the hardware element utilized to support ML. The integrated Edge TPU processor performs 4 trillion operations (tera-operations) per second (TOPS), using 0.5 watts for each TOPS. For a development and simulation configuration the Coral USB Accelerator is also supported.
- HSM: Similarly to the OBU, to provide security-related functions, the anti-hacking device hardware also integrates an HSM. Indeed, a Telekom Card Operating System (TCOS) embedded smartcard module is integrated in the anti-hacking device, supporting secure storage of private keys and different cryptographic operations, e.g., authentication of the anti-hacking device for remote provisioning and updates or for central event reporting and alerting.
- NXP Freescale i.MX8 processor: The adopted processor supports security functions such as High-Assurance Boot (HAB) and Cryptographic Accelerator and Assurance Module.

- Yocto-based firmware layer (a Linux embedded meta distribution): The firmware for the i.MX8 SOC is created using the Yocto environment which is an industry-standard toolkit to create custom embedded firmware images in a reproducible manner. The anti-hacking device build process supports signed bootloaders and a Linux kernel in order to prevent tampering with the anti-hacking device software and configuration.
- Docker-based application-specific containers: Out-of-the box, crypto containers supporting the security functions of the anti-hacking device are present. ML workloads are implemented as containers that have access to the underlying ML hardware as well as to the crypto functions exported by the crypto containers.
- The anti-hacking device could also act as a secure run-time environment for other functions as needed by the different use cases.

3 Overview of CARAMEL Connectivity Attacks

While the potential threats and vulnerabilities that may be encountered by a generic connectivity infrastructure have been introduced in Section 1.1, herein a subset of the possible security enhancements considered within the CARAMEL project is presented: (i) Section 3.1 presents an overview on the privacy mechanisms and on the high secure communication enabled; (ii) Section 3.2 showcases the mechanisms in place in CARAMEL to protect the OBU from possible tampering attacks; (iii) Section 3.3 describes a possible attack on one of the vehicle's sensors, i.e., the GPS receiver. More details on how the CARAMEL's architecture copes with attacks on sensors are presented in Section 4.

3.1 Scenario 1 – Attack on the V2X Message Transmission

This scenario has two main objectives. Firstly, to demonstrate the correct coordination between the PKI and vehicles to distribute the pseudonymous ATs, the use of ATs to sign V2X messages, and their verification to detect non authorized/replayed messages or messages signed with revoked certificates. Secondly, to provide a mechanism to improve privacy by minimizing the possibility that vehicles transmitting ITS messages are tracked.

In this scenario, ITS messages transmitted by vehicles are directly signed by their HSM which provides the necessary protection to prevent their private keys from being stolen. The verification of the signature is also performed by the HSM if the receiver is another vehicle, or by the Vanetza software package if the receiver is the MEC. On the other hand, privacy is performed using pseudonymous identifiers in the ATs (instead of real identifiers), and changing the AT at given intervals. However, knowing the position of vehicles and the time interval used to renew ATs, tracking by an attacker becomes trivial. In CARAMEL, an ML-based algorithm running in the Anti-hacking device optimizes the moment when the AT is renewed. Considering the V2X messages sent by the surrounding mobile entities, a time instant that allows hiding in the crowd will be chosen by the vehicle for its AT renewal.An exhaustive search was performed in order to obtain such optimal moment for changing the ATs. First, a dataset of 30 billions V2X messages was generated based on the simulations performed by Uppoor et al. in [14], representing

24 hours of dense traffic in the city of Köln. Then, with this dataset, a ML algorithm capable of tracking vehicles from their V2X transmissions was trained. One of the conclusions was that it was rather easy to track the vehicles when they sent the V2X messages in periods of 100±50 ms (usually, they are sent every 100 ms following the ETSI standards). It was also possible to quantify/score how difficult it was to do such tracking in terms of confidence of the results, computational resources needed, response time, etc. Based on this score, the implemented algorithm decides when to change the AT. In order to do so, the connected vehicle calculates this score at each packet reception and decides if it is the right moment to change the AT looking at previous scores and applying optimal stopping methods [15].

3.2 Scenario 2 – Tamper Attack to a Vehicle's OBU

In hardware tampering attacks, the adversary actively interacts with the device and/or its components by, for instance, inducing deliberate faults into the computation and observing its result at the output. The severity of the tampering can range from just naive manipulation such as breaking a seal, to dangerous manipulation resulting in accessing privileged information. Therefore, tampering attacks are directed to a specific vehicle affecting its privacy and safety, and, potentially, to all vehicles in the surrounding area receiving corrupted information. In order to comply with the security functional requirements of the CARAMEL project, several hardware design techniques have been applied. In the next paragraphs, the OBU interfaces are reviewed, and the potential OBU attacks and counterattacks through the various interfaces are described. Figure 4 summarizes the adopted securization techniques used.

3.2.1 OBU Interfaces

The vehicle's OBU is used for securing the V2X communication between vehicles and between vehicles and their environment in an ITS. Four interfaces are identified as shown in Figure 4:

- ITS interface: The application processor sends messages through the V2X transceiver to establish communication with other ITS stations and the ITS infrastructure.
- HSM interface: Communication channel with HSM for cryptographic and key management functions.
- IVN interface: Communication over In-Vehicle Network towards the vehicle through the Printed Circuit Board (PCB) connector.
- GNSS interface: Positioning data communication interface to the main processor.

3.2.2 Threats for Tamper Attack of the Vehicle's OBU

The potential threats for tamper attack of the vehicle's OBU that have been considered in this project are identified hereafter:

- Tamper attack on the ITS interface: The attacker uses tampered V2X messages to cause safety hazardous situations.
- Software tamper attack on the ITS interface: The attacker uses malicious software on the V2X front end to track ITS stations or to send rogue messages on the ITS network.

- Clock fault injection attack on the ITS interface: The attacker manipulates the front end's clock to generate malfunctions or break security in the ITS interface.
- Software tamper attack on the main processor: The attacker uses malicious software on the main processor to cause safety hazardous situations.
- Clock fault injection attack on the main processor: The attacker manipulates the main processor's clock to generate malfunctions or break security.
- Voltage fault injection: The attacker manipulates the power supply to generate malfunctions or break security.
- Temperature fault injection: The attacker manipulates the environmental temperature to generate malfunctions or break security.
- Eavesdropping main processor data signals: The attacker eavesdrops on the communication of the main processor memory to obtain confidential information (encryption keys, secure certificates, etc).
- Tamper attack on the HSM interface: The attacker uses tampered HSM messages to cause safety hazardous situations and to get privileges.
- Tamper attack on the GNSS interface: The attacker injects malicious geolocation data to cause safety hazardous situations.
- Software tamper attack on the GNSS interface: The attacker uses malicious software on the GNSS to cause safety hazardous situations.

3.2.3 Anti-tamper Hardware Techniques Implemented on OBU

Since anti-tampering techniques are not bullet-proof, an "onion layered" approach becomes necessary on the design of the OBU hardware securization. Overlaid techniques provide more robust protection: the attacker must disable a protection layer before dealing with the next level of protection. Based on the threats explained in Section 3.2.2, a brief description of the different protection layers implemented in CARAMEL is shown in the list hereafter:

- Environmental sensors: Voltage, temperature and clock sensors added to protect against fault injection attacks.
- Opening enclosure detection: Protects against the physical access to the OBU internal environment.
- Coating sensible circuits: Encapsulation of some circuitry with ruggerized epoxy compounds to avoid physical access. If the attacker tries to remove the encapsulation, some components will be broken and an alarm is triggered.
- Mutual authentication: protects against lifting of critical OBU internal devices and using them in an unintended environment by requiring mutual authentication at start-up.
- Data encryption: Ensures integrity and confidentiality of exchanged messages between devices in the OBU.
- Secure boot: Uses a combination of hardware and software together with a public key to protect the system from executing unauthorized programs.
- Trusted execution environment: Is a secure area on the main processor. Software running in this environment is protected against attacks from potentially compromised platform software.

Table 2 relates the above mentioned countermeasure layers with the most relevant threats in the OBU.

Countermeasure Layers \ Threats	Enclosure manipulation	ITS interface tamper attacks	V2X HSM interface tamper attacks	GNSS interface tamper attacks	Eavesdropping data signals	Clock fault injection	Temperature fault injection	Voltage fault injection
Environmental sensors						•	•	•
Opening enclosure detection	•	•	•	•	•			
Coating covering sensible circuits, with self-destructive components to avoid coating removal						•		
Active wire-mesh protection for critical elements and signals		•		•	•	•		
Mutual authentication		•	•					
Data encryption			•		•			
Secure boot		•	•					
Application processor trusted execution environment		•	•					

Table 2 Relationship between countermeasure and possible threats for the OBU.

3.3 Scenario 3 – GPS Spoofing Attack

Even if the vehicle is perfectly secured, as well as the in-vehicle and between vehicles communication, an attacker may carry an attack in the environment where the AV is moving. A possible attack of this kind is represented by the GPS spoofing attack.

In general, civilian GPS signals are unencrypted and unauthenticated, thus a user can arbitrarily generate or change the signals (via Software Defined Radio (SDR) hardware/software). In this attack, the GPS receiver in the AV is deceived by broadcasting fake satellite signals, structured to resemble a set of normal GPS signals. Typically, a viable attack strategy only requires to align spoofed signals with the true signals and, starting at low level, to increase their power of transmission until they capture the receiver's tracking loops. Once the receiver is locked to spoofed signals, an attacker can alter them in order to cause the receiver to estimate its position to be somewhere other than where it actually is.

To carry out a successful GPS spoofing attack, an accurate knowledge of the target receiver position and trajectory is required [16]. Without such precise information, the attack would trigger a large modification of the receiver localization or of the GPS time. Two possible ways can be used to carry such an attack: (i) via portable receiver-spoofer co-located with the target antenna, where this difficulty is overcome by construction; (ii) from distance, with a static station. In the first case, the receiver-spoofer can be made small enough to be co-located with the target antenna. The receiver component draws in genuine GPS signals to estimate its own position, velocity, and time, which also hold for the attacked GPS receiver due to proximity. Then, based on such information, the attacker generates accordingly counterfeited GPS signals to orchestrate the spoofing. When instead the attack is done from some distance, the relative distance between the attacked GPS receiver and the spoofer needs to be estimated and predicted over short-term time windows. This increases the difficulty of the attack if the actual intent is to alter in an orchestrated way the output of the attacked GPS receiver.

The GPS spoofing attack is in general difficult to detect. As described in Section 4 below, CARAMEL is able to detect the location spoofing attack thanks to a parallel stream of vehicle locations that relies on GPS-free signals, e.g., in-vehicle sensor measurements, or thanks to a collaborative approach enabled by the support of the infrastructure. This secondary location stream is compared with the GPS locations and in case their difference exceeds a predefined threshold, an alarm is raised to signify a GPS spoofing attack.

4 The CARAMEL System in Action

In this section, some early results on one of the three scenarios considered in the CARAMEL connectivity architecture is presented, i.e., the attack on one of the sensors of the vehicle – the GPS receiver. First, the framework for identifying the GPS spoofing attacks in CARAMEL is presented. Two possible implementations are envisioned for GPS location integrity check: (i) an approach based on an *in-vehicle* scheme (Section 4.1.1) and (ii) an approach based on a *collaborative* effort among vehicles that exploits infrastructure support (Section 4.1.2). Then, once the attack is identified, the *mitigation technique* used in CARAMEL as a countermeasure, i.e., the vehicle certificate revocation, is showcased (Section 4.2).

4.1 The GPS Spoofing Attack Detection

Nowadays, solutions for location spoofing resilience are under study. For instance, the first satellite system to propose an anti-spoofing service on a civil GNSS signal is Galileo. Indeed, Galileo proposes on its E1 frequency the use of Open Service Navigation Message Authentication (OS-NMA), which enables authentication of the navigation data [17]. However, despite anticipation, no integrated circuit designs for OS-NMA on E1 have been released to date and some experts question the usefulness of such solution if receivers can deliver anti-spoofing protection based on inertial sensors or signal processing [18]. To this end, the CARAMEL project presents two alternative low-cost and fast-to-deploy solutions.

4.1.1 In-vehicle GPS Location Integrity Check

In this approach, the CARAMEL system computes an alternative localization of the vehicle using a Bayesian filtering technique to check the integrity of the GPS measurements. The idea underlying the proposed approach is to obtain a fall-back localization technique for a specific vehicle that does not relay on GPS measurements. The approach is modular, and it is summarized in Figure 7. To achieve the fall-back localization technique, the proposed Bayesian filer is composed of the following two basic steps: (i) the prediction step; and (ii) the update step. For the *prediction* step, the motion of the vehicle is described through the characterization of the underlying physical laws and the future vehicle location is obtained through on-board sensor measurements. For the *update* step, the predicted location of the vehicle is fused with a GPS-free global location measurements obtained by an alternative location system inside the vehicle. The output of the proposed Bayesian filter is then compared with the actual GPS measurements in order to detect substantial localization deviations, hence a possible GPS spoofing attack. Potentially, depending on the quality of the global location measurements used in the update step,

the CARAMEL system could revert to the fall-back location solution to steer temporarily the vehicle, while the attack is in place. Notably, the solution adopted in CARAMEL adapts to the available on-board sensors and to the available GPS-free global location measurements.

For demonstration purposes, the fall-back location stream in CARAMEL is implemented as a container within the anti-hacking device of the vehicles' OBU, as shown previously in Figure 3. The software has access to the CAN bus data and, among others, to the steering angle (α), the yaw rate ($\dot{\phi}$), and the wheel speed (v) sensor data. Exploiting such sensors information, it is possible to build a non-linear model of the vehicle system state following the underlying physical laws. Such non-linear model exploits the basic assumption that the motion of a vehicle can be well approximated by a bicycle, i.e., collapsing the rear and the front axes into a single point. Given the adopted bicycle model, the motion model of the vehicle can be described considering the involved inertial forces, e.g., the friction of the wheels on the pavement. If the body-frame of the vehicle is considered oriented as the x-axis, the one-step prediction of the location and the speed of the vehicle in its body-frame reference system is [19]:

$$\begin{cases} x^u_{k+1} = v\Delta t \\ \dot{x}^u_{k+1} = v \\ y^u_{k+1} = (C_f(\alpha - \frac{l_f\dot{\phi}}{v}) + C_r(\frac{l_r\dot{\phi}}{v}))\frac{\Delta t^2}{2M} \\ \dot{y}^u_{k+1} = (C_f(\alpha - \frac{l_f\dot{\phi}}{v}) + C_r(\frac{l_r\dot{\phi}}{v}))\frac{\Delta t}{M} \end{cases} \tag{1}$$

where l_f and l_r represent the distance of the front wheel and the rear wheel from the mass barycentre, respectively, M is the mass of the vehicle, and C_f and C_r represent the corner stiffness of the front and rear wheels, respectively. Given the prediction of the vehicle movement in its body-frame, a simple coordinate transformation is applied to obtain a one-step prediction in the global geographic reference system. Under the assumption of uncorrelated and Gaussian measurement noise, the associated covariance of the estimated vehicle's system state is computed with a Bayesian Filter, i.e., an Extended Kalman Filter (EKF) approach. The EKF is also used to update the obtained predicted system state and uncertainty with a GPS-free location measurement. In the update step of the EKF, a global location measurement of the vehicle is obtained through Signals of Opportunity (SoO) [20]. In this technique, a passive receiver located at the vehicle scans a predetermined set of frequencies where transmitters are typically active, e.g., LTE and RSU bands. Using the average received power at the selected bandwidths, a local ML-based algorithm estimates the wireless path loss and computes the approximate distance between the vehicle and the corresponding transmitters. Applying simple multi-lateration techniques provides, with some uncertainty, the relative displacement of the vehicle and, thanks to the knowledge of some anchor points, e.g., a transmitter location, an estimation of the global location of the vehicle. If the error of the SoO update step is approximated as Gaussian, as assumed in CARAMEL, the output of the fall-back solution provides an approximation of the vehicle's location that follows a Gaussian distribution as well; i.e., the output of the fall-back solution

is the average of the vehicle's location estimation $[\mu_x, \mu_y]$, and the corresponding covariance matrix $\Sigma_{x,y}$.

In order to identify a possible GPS spoofing attack, the output of the CARAMEL's fall-back solution is compared with the GPS measurements. Indeed, the GPS receiver not only provides an approximated location $[\mu_x^G, \mu_y^G]$ for the vehicle, but also an uncertainty score that can be transformed into a covariance matrix $\Sigma_{x,y}^G$ [21]. If also the GPS measurements are approximated as a Gaussian distribution, then a natural comparison between the two location measurements is represented by the Bhattacharyya distance (the Bhattacharyya distance computes the amount of overlap of two statistical distribution, hence, measuring their similarity). If the Bhattacharyya distance between the two distributions exceeds a predetermined threshold T, then, an alarm is raised. Specifically, at each time slot k where a new GPS measurement is received, the following average Bhattacharyya distance is computed:

$$D = \sum_{n \in [k-W,k]} \frac{1}{8} \mu(n) \left(\frac{\Sigma_{x,y}(n) + \Sigma_{x,y}^G(n)}{2} \right)^{-1} \mu(n)^T + \frac{1}{2} \ln \left(\frac{\det \frac{\Sigma_{x,y}(n) + \Sigma_{x,y}^G(n)}{2}}{\sqrt{\det \Sigma_{x,y}(n) \det \Sigma_{x,y}^G(n)}} \right) \tag{2}$$

where $\mu(n) = [\mu_x(n), \mu_y(n)] - [\mu_x^G(n), \mu_y^G(n)]$, and the Bhattacharyya distance is averaged over the samples collected over a sliding window of W seconds. The sliding window mechanism allows reducing the number of false alarm due to spurious GPS measurement errors. Nevertheless, the trade-off between the length of the sliding window and the ability of the described attack detection mechanism to react to a GPS spoofing attack has to be taken into account when setting W.

In order to assess the ability to notify the CARAMEL infrastructure of an ongoing GPS spoofing attack, the described mechanism has been implemented in the CARLA simulator [22]. Figure 8 showcases an example of the obtained results. Specifically, Figure 8a depicts: (i) the actual trajectory of the vehicle, directly from the ground-truth notified by the CARLA simulator; (ii) the fall-back location solution, where the SoO is simulated as a GPS-free measurement with very large variance, i.e., $\mathcal{N}(0, \text{diag}(225\ m^2, 225\ m^2))$[2]; and (iii) the GPS measurements received by the vehicle (distributed as a $\mathcal{N}(0, \text{diag}(9\ m^2, 9\ m^2))$), attacked by a malicious entity after the first half of the simulation time with a fixed bias equal to 15 m. Figure 8b shows instead the output of the detection approach envisioned in CARAMEL. As expected, the instantaneous Bhattacharyya distance presents high variability, making the detection of a possible attack more difficult. Nevertheless, the average Bhattacharyya distance D, with sliding time window of $W = 4$s, greatly simplifies the process. Considering as the attack threshold detection T the 99-th percentile of the Bhattacharyya distance D under normal circumstances, the approach proposed in CARAMEL is able to detect as attacked 97% of the GPS measurements

[2]Note that the motion model used in CARLA does not follow the adopted bicycle model. Hence, as in reality, such motion model only represents an approximation of the vehicle's mobility.

maliciously modified. Finally, Figure 9 shows the results of a large simulation campaign where both the length of the sliding window W and the module of the bias used to modify the GPS measurements vary over predefined intervals. The detection rate of a tampered GPS measurement reaches up to 98% in some simulation set-ups. Furthermore, a larger sliding window W improves the performance of the proposed approach, especially when the attack bias introduced is smaller than the SoO location uncertainty.

4.1.2 Collaborative GPS Location Integrity Check

A collaborative approach exploiting the CARAMEL infrastructure for GPS integrity check is now presented.

Consider a vehicular network of N interconnected vehicles that are moving in the road network. The location of vehicle i at time t is denoted by $\mathbf{X}_i^{(t)} = \left[x_i^{(t)}, y_i^{(t)}\right]^T$. Based on [23], each vehicle at time t can collect four types of measurements: (i) absolute position measurement $z_p^{(t)}$ from the GPS, (ii) relative distance measurement $z_d^{(t)}$, (iii) relative angle measurement $z_a^{(t)}$ and (iv) relative azimuth angle measurement $z_{az}^{(t)}$ between neighboring vehicles using LIDAR/RADAR. The relative distance at time t between the neighboring vehicles i and j is modeled as $z_d^{(t)}[i,j] = \left\| X_i^{(t)} - X_j^{(t)} \right\|_2 + N_d$, where $\|(.)\|_2$ is the Euclidean distance and N_d is the measurement noise. The relative angle at time t between neighboring vehicles i and j is modeled as $z_a^{(t)}[i,j] = \arctan(y_j^{(t)} - y_i^{(t)})/(x_j^{(t)} - x_i^{(t)}) + N_a$, while the relative azimuth angle is equal to $z_{az}^{(t)}[i,j] = \lambda\pi + \arctan|x_j^{(t)} - x_i^{(t)}|/|y_j^{(t)} - y_i^{(t)}| + N_{az}$, with $\lambda = \{0,1\}$, or $z_{az}^{(t)}[i,j] = \lambda\pi + \arctan|y_j^{(t)} - y_i^{(t)}|/|x_j^{(t)} - x_i^{(t)}| + N_{az}$, with $\lambda = \{\frac{1}{2}, \frac{3}{2}\}$, where N_a and N_{az} are the measurement noises. Obviously, $z_p^{(t)}[i] = \mathbf{X}_i^{(t)} + N_p$, where N_p is the GPS noise. The accuracy of the GPS, as well as the detection rate of possible location attacks, can be improved by fusing these measurements, which is known as the multi-modal fusion method for cooperative localization [24].

All vehicles transmit their measurements, through CAM messages, to an ITS application that runs in the MEC. In the MEC, first the measurement model for the spoofed GPS is modified according to: $\overline{z}_p^{(t)}[i] = z_p^{(t)}[i] + O_p^{(t)}[i]$, where $O_p^{(t)} = [O_p^{(x)}, O_p^{(y)}]$ is a sparse outlier matrix modeling the impact of a location attack. Then, such impact is evaluated through a collaborative location estimation approach. To this end, assuming Gaussian noise measurements, the estimated locations of the N vehicles is obtained with the following minimization problem according to maximum likelihood estimation (MLE) and sparsity constraints:

$$\begin{aligned}
\underset{X^{(t)}, O_p^{(t)}}{\operatorname{argmin}} \; C^{(t)} = & \sum_{i=1}^{N} \sum_{j=1}^{N(i)} \left(z_d^{(t)}[i,j] - \left\| X_i^{(t)} - X_j^{(t)} \right\|_2 \right)^2 \\
& + \sum_{i=1}^{N} \sum_{j=1}^{N(i)} \left(z_a^{(t)}[i,j] - \arctan \frac{y_j^{(t)} - y_i^{(t)}}{x_j^{(t)} - x_i^{(t)}} \right)^2 \\
& + \sum_{i=1}^{N} \left\| \left(\overline{z}_p^{(t)}[i] - O_p^{(t)}[i] \right) - X_i^{(t)} \right\|_2^2 + \lambda \left\| O_p^{(t)} \right\|_1
\end{aligned} \tag{3}$$

The interior point methods provided by off-the-shelf software, e.g., by the CVX solver [25], can be applied in order to minimize the cost function. The output of this approach, named as Robust Traditional Collaborative Localization based on MLE (RTCL-MLE), is compared against the GPS locations to detect attacked vehicles. If the difference exceeds a predefined threshold, then an attack is detected.

An alternative approach is to treat the VANET as an undirected graph, using the connected vehicles as its vertices and the communication links between them as its edges. The associated Extended Laplacian Matrix $\tilde{L}^{(t)}$ of the VANET graph and the differential coordinates $\delta^{(x),(t)} = \frac{1}{d_i^{(t)}} \sum_{j \in N(i)} -z_d^{(t)}[i,j] \sin z_{az}^{(t)}[i,j]$, $\delta^{(y),(t)} = \frac{1}{d_i^{(t)}} \sum_{j \in N(i)} -z_d^{(t)}[i,j] \cos z_{az}^{(t)}[i,j]$ of each vehicle, can be derived according to that graph modeling and the measurement models (see [26] for more details). Note that $d_i^{(t)}$ is the number of connected neighbors to the i-th vehicle and $N(i)$ is the set of its neighbors, at time instant t. Afterwards, vectors $b^{(x),(t)} = [\delta^{(x),(t)}, z_p^{(x),(t)}]$, $b^{(y),(t)} = [\delta^{(y),(t)}, z_p^{(y),(t)}]$ are defined accordingly, where $z_p^{(x),(t)}, z_p^{(y),(t)}$ are the GPS positions of the vehicles in the network, assuming that they act as anchors. The outliers of position, modeled by $O_p^{(x)}, O_p^{(y)}$ matrices, must be removed only from the anchors/GPS part of vectors $b^{(x),(t)}, b^{(y),(t)}$. Thus, two minimization problems have been formulated, based on the graph representation of VANET and sparsity properties, in order to estimate the locations of the N vehicles, hence detecting and mitigating possible attacks on the GPS measurements. Once again, the interior point methods provided by off-the-shelf software can be applied in order to solve the two minimization problems. This approach is named hereinafter Robust Graph-based Collaborative Localization (RGCL).

$$\underset{x^{(t)}, O_p^{(x),(t)}}{\operatorname{argmin}} \left\| \tilde{L}^{(t)} x^{(t)} - \left(b^{(x),(t)} - O_p^{(x),(t)} \right) \right\|_2^2 + \lambda_1 \left\| O_p^{(x),(t)} \right\|_1 \tag{4}$$

$$\underset{y^{(t)}, O_p^{(y),(t)}}{\operatorname{argmin}} \left\| \tilde{L}^{(t)} y^{(t)} - \left(b^{(y),(t)} - O_p^{(y),(t)} \right) \right\|_2^2 + \lambda_2 \left\| O_p^{(y),(t)} \right\|_1 \tag{5}$$

During the detection phase of either the two minimizations, i.e., (4) and (5), a vector containing the distances between the initial GPS locations and the estimated locations is formed. Afterwards, a small threshold equal to 10 is set, implying that distances below 10 m do not correspond to attacked vehicles, while distances greater than 10 m may be indicative of an attack. In the latter case, the k-means clustering algorithm, with k=2, is applied on the corresponding distances, producing two clusters with associated centers. The cluster with the largest center contains, in fact, the distances that correspond to attacks. As such, the IDs of spoofed vehicles can be identified.

As a simple example, a network of 20 moving vehicles/nodes is considered for 100 time instances. Figure 10 depicts some preliminary results. In the simulations, the different measurement errors are as follows: $\sigma_x = 3$ m, $\sigma_y = 2.5$ m, $\sigma_d = 3$ m and $\sigma_a = \sigma_{az} = 4^\circ$. The CDFs of the maximum GPS and cooperative location estimation errors are plotted at each time instance with 2 and 4 attacked vehicles/nodes, respectively. The attack is simulated by adding a bias (sampled uniformly in the

interval of $[5, 40]$) to the attacked nodes, resulting in an average (with respect to attacked vehicles) localization error equal to 34 m. In the event of location spoofing attack to 2 or 4 vehicles, the proposed approaches demonstrate remarkable robustness as the localization error is slightly increased, contrary to the GPS location error. Moreover, RGCL always outperforms RTCL-MLE, highlighting the benefits of exploiting the VANET graph representation and properties. Finally, RGCL achieves not only the reduction of GPS spoofing error, but also the attacked-free GPS error, proving its superior performance and robustness.

In Figure 11, the Receiver Operating Characteristic (ROC) curves of the detection phase of the two schemes is also provided, when 2 and 4 vehicles/nodes are attacked. The performance of the methods is evaluated by the Area Under Curve (AUC) measurement. In Figure 11-(a), AUC with RGCL is 0.97, while in RTCL-MLE is 0.96. In Figure 11-(b), AUC with RGCL is 0.95, while in RTCL-MLE is 0.94. In the latter case, a slight degradation of classification performance is observed, due to the increased number of compromised vehicles. However, the two methods exhibit remarkable classification accuracy performance, as far as the detection of attacked vehicles is concerned. Moreover, RGCL outperforms RTCL-MLE, proving its superiority and robustness, both in collaborative locations estimation and detection of attacks steps.

4.2 Certificate Management: A Candidate Countermeasure

In current systems, certificates are managed over long periods of time and modifications take place after several days. Nevertheless, this approach is not sufficient in general and especially in the case of GPS spoofing attacks. Therefore, a more agile method to distribute the revocations is needed. CARAMEL bridges this gap by incorporating a system to distribute the CRL to vehicles in real time. This scenario comprises of two possible implementations, following the attack detection approaches described before:

- The OBU detects a misbehavior in the GPS receiver, i.e., it detects a GPS spoofing attack by means of the proposed in-vehicle detection solution. An alarm is then sent to the MEC, which takes the decision to revoke its authorization certificate.
- A process running in the MEC that implements the proposed collaborative detection solutions identifies a GPS location spoofing and takes the decision to revoke the authorization certificate of the vehicle under attack.

Subsequently, the MEC will take all the necessary actions to inform, in real time, the PKI servers and all other vehicles of the system about the revoked certificate of the attacked vehicle. In both cases, all entities of the CARAMEL connectivity architecture will be involved: (i) the MEC, to detect dangerous situations, to decide if certificates should be revoked, and to distribute CRL to vehicles and PKI servers; (ii) the OBU, to detect a misbehaving situation in the vehicle and inform the MEC; (iii) the PKI servers, to remove the revoked certificates from their trusted vehicle lists; (iv) the vehicles, to receive and store the revoked certificates and, when an incoming message is processed, check if its certificate is or is not revoked. Whenever a vehicle is under attack or it misbehaves, its certificates will be temporarily or permanently revoked. Finally, note that all communication between the OBUs

and the fixed infrastructure related to certificates, either valid or revoked, will be transmitted using the LTE-Uu channel.

5 Conclusion

The CARAMEL project investigates advanced methods for the detection and mitigation of cybersecurity attacks in CAVs. Specifically, a novel secure architecture enhancing the end-to-end verification of the transmitted data among entities in CAV scenarios is presented. Such architecture includes: (i) a PKI, which distributes and updates the certificates used by all entities to sign their data transmissions; (ii) a multi-RAT communication infrastructure with MEC functionalities, providing computational capabilities close to the end-users and enabling vehicles on-boarding different technologies, e.g., 802.11p and LTE, to communicate with each other; (iii) an OBU resistant to tampering attacks, which integrates an anti-hacking device capable of running ML techniques extending the its security capabilities.

This work focuses on the ability of the connectivity infrastructure introduced by CARAMEL to detect and mitigate GPS location spoofing attacks, which pose a serious threat to all involved actors in the autonomous mobility habitat, including vehicles, infrastructure, drivers, and pedestrians. Two complementary approaches are proposed for detecting such attacks and the development of a future feasible countermeasure, based on revoking the certificates of the attacked vehicles, is outlined.

As a future step, the overhead of CRLs distribution on the network traffic load, as well as scalability with regards to the number of attacked vehicles (and consequently the volume of revoked certificates) will be studied.

Competing interests

The authors declare that they have no competing interests.

Acknowledgment

This work was supported by the European Union's H2020 research and innovation programme under the CARAMEL project (Grant agreement No. 833611). The work of Christian Vitale, Christos Laoudias and Georgios Ellinas was also supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant 739551 (KIOS CoE) and from the Republic of Cyprus through the Directorate General for European Programmes, Coordination, and Development. The work of Jordi Casademont and Pouria Sayyad Khodashenas was also supported by FEDER and Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya through projects Fem IoT and SGR 2017-00376 and by the ERDF and the Spanish Government through projects TEC2016-79988-P and PID2019-106808RA-I00 AEI/FEDER, UE.

Author details

[1]KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, Cyprus. [2]Depart. of Electrical and Computer Engineering, University of Cyprus, Nicosia, Cyprus. [3]Industrial Systems Institute, Athena Research and Innovation Center, Pastras, Greece. [4]Dept. of Electrical and Computer Engineering, University of Patras, Pastras, Greece. [5]i2CAT Foundation, Barcelona, Spain. [6]Universitat Politècnica de Catalunya, Barcelona, Spain. [7]Atos IT Solutions and Services Iberia S.L., MAdrid, Spain. [8]FICOSA, Barcelona, Spain. [9]Panasonic Automotive, Langen, Germany. [10]T-Systems International GmbH, Frankfurt, Germany.

References

1. Cui, J., Liew, L.S., Sabaliauskaite, G., Zhou, F.: A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. Ad Hoc Networks **90**, 101823 (2019)
2. Mejri, M.N., Ben-Othman, J., Hamdi, M.: Survey on VANET security challenges and possible cryptographic solutions. Vehicular Communications **1**(2), 53–66 (2014)
3. Vitale, C., Piperigkos, N., Laoudias, C., Ellinas, G., Casademont, J., Khodashenas, P.S., Kloukiniotis, A., Lalos, A.S., Moustakas, K., Lobato, P.B., *et al.*: The CARAMEL project: a secure architecture for connected and autonomous vehicles. In: 2020 European Conference on Networks and Communications (EuCNC), pp. 133–138 (2020). IEEE
4. Deepa Thilak, K., Amuthan, A.: DoS attack on VANET routing and possible defending solutions-A survey. 2016 International Conference on Information Communication and Embedded Systems, ICICES 2016 (Icices), 1–7 (2016). doi:10.1109/ICICES.2016.7518892

5. Chuang, M.C., Lee, J.F.: TEAM: Trust-extended authentication mechanism for vehicular ad hoc networks. IEEE Systems Journal **8**(3), 749–758 (2014). doi:10.1109/JSYST.2012.2231792
6. Mokdad, L., Ben-Othman, J., Nguyen, A.T.: DJAVAN: Detecting jamming attacks in vehicle ad hoc Networks. Performance Evaluation **87**, 47–59 (2015). doi:10.1016/j.peva.2015.01.003
7. Whyte, W., Weimerskirch, A., Kumar, V., Hehn, T.: A security credential management system for V2V communications. IEEE Vehicular Networking Conference, VNC, 1–8 (2013). doi:10.1109/VNC.2013.6737583
8. Salem, F.M., Ibrahim, M.H., Ibrahim, I.: Non-interactive authentication scheme providing privacy among drivers in vehicle-to-vehicle networks. In: 2010 ICNS, pp. 156–161 (2010). IEEE
9. Wasef, A., Shen, X.S.: EMAP: Expedite message authentication protocol for vehicular ad hoc networks. IEEE Transactions on Mobile Computing **12**(1), 78–89 (2013). doi:10.1109/TMC.2011.246
10. Li, J., Lu, H., Guizani, M.: Acpn: A novel authentication framework with conditional privacy-preservation and non-repudiation for VANETs. IEEE Transactions on Parallel and Distributed Systems **26**(4), 938–948 (2014)
11. Liu, L., Esmalifalak, M., Ding, Q., Emesih, V.A., Han, Z.: Detecting false data injection attacks on power grid by sparse optimization. IEEE Transactions on Smart Grid **5**(2), 612–621 (2014). doi:10.1109/TSG.2013.2284438
12. Riebl, R., Obermaier, C., Neumeier, S., Facchi, C.: Vanetza: Boosting research on inter-vehicle communication. Proceedings of the 5th GI/ITG KuVS Fachgespräch Inter-Vehicle Communication (FG-IVC), 37–40 (2017)
13. ETSI, T.: ETSI TS 102 941 v1. 3.1-intelligent transport systems (ITS); security; trust and privacy management," Standard, TC ITS, (2019)
14. Uppoor, S., Trullols-Cruces, O., Fiore, M., Barcelo-Ordinas, J.M.: Generation and analysis of a large-scale urban vehicular mobility dataset. IEEE Transactions on Mobile Computing **13**(5), 1061–1075 (2013)
15. Hill, T.P.: Knowing when to stop: how to gamble if you must—the mathematics of optimal stopping. American Scientist **97**(2), 126–133 (2009)
16. Tippenhauer, N.O., Pöpper, C., Rasmussen, K.B., Capkun, S.: On the requirements for successful gps spoofing attacks. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, pp. 75–86 (2011)
17. Fernandez-Hernandez, I., Vecchione, G., Díaz-Pulido, F.: Galileo authentication: a programme and policy perspective. In: 69th International Astronautical Congress (2018)
18. Gutierrez, P.: Galileo to transmit open service authentication. Inside GNSS (2020)
19. Rezaei, S., Sengupta, R.: Kalman filter-based integration of DGPS and vehicle sensors for localization. IEEE Transactions on Control Systems Technology **15**(6), 1080–1088 (2007)
20. Souli, N., Kolios, P., Ellinas, G.: Relative positioning of autonomous systems using signals of opportunity. In: 2020 IEEE 91st Vehicular Technology Conference (VTC Spring), pp. 1–6 (2020). IEEE
21. Almagbile, A., Wang, J., Ding, W.: Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration. Journal of Global Positioning Systems **9**(1), 33–40 (2010)
22. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Conference on Robot Learning, pp. 1–16 (2017)
23. Kim, H., Lee, S.H., Kim, S.: Cooperative localization with distributed ADMM over 5G-based VANETs. In: 2018 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–5 (2018). doi:10.1109/WCNC.2018.8377454
24. Bahr, A., Walter, M.R., Leonard, J.J.: Consistent cooperative localization. In: 2009 IEEE International Conference on Robotics and Automation, pp. 3415–3422 (2009). IEEE
25. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 2.1 (2014)
26. Sorkine, O.: Laplacian mesh processing. 26th Annual Conference of the European Association for Computer Graphics, Eurographics 2005 - State of the Art Reports **29**, 53–70 (2005)

Figures

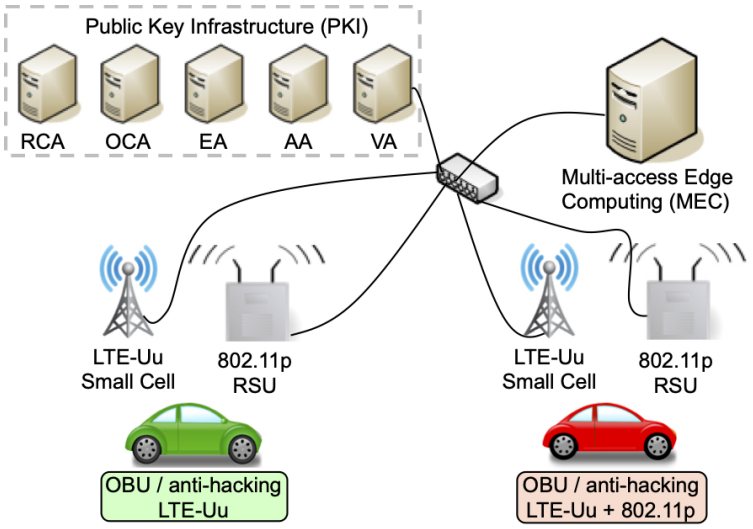


Figure 1 Secure multi-technology V2X telecommunications infrastructure.

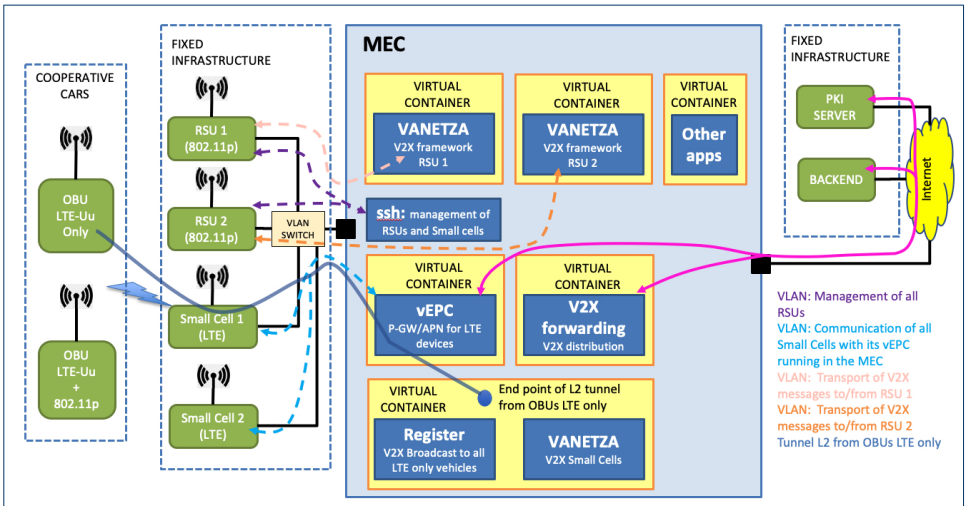


Figure 2 Proposed network architecture.

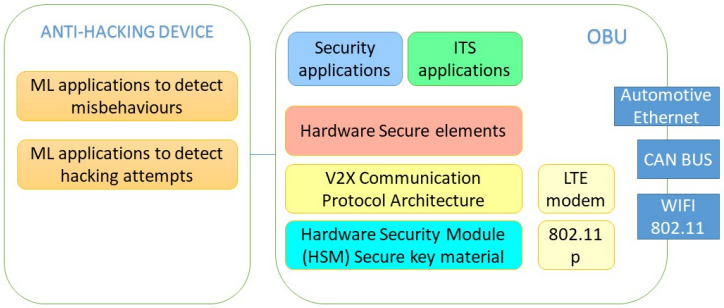


Figure 3 Architecture of OBU and anti-hacking device in a cooperative car.

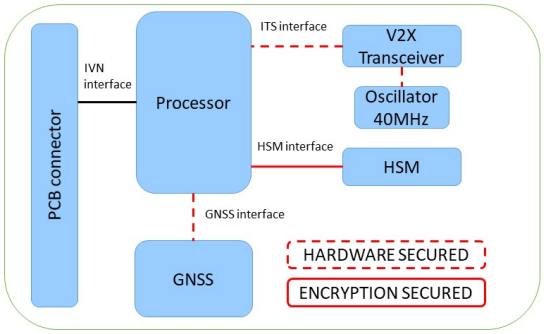


Figure 4 Schematics of hardware securization interfaces in OBU.

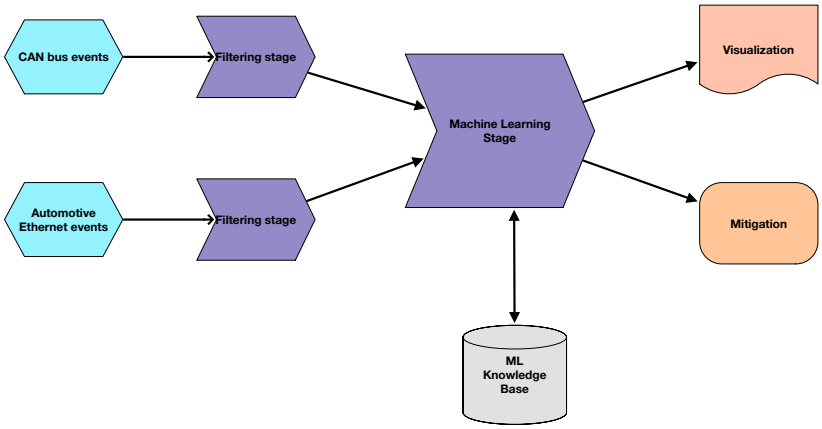


Figure 5 Machine learning pipeline.

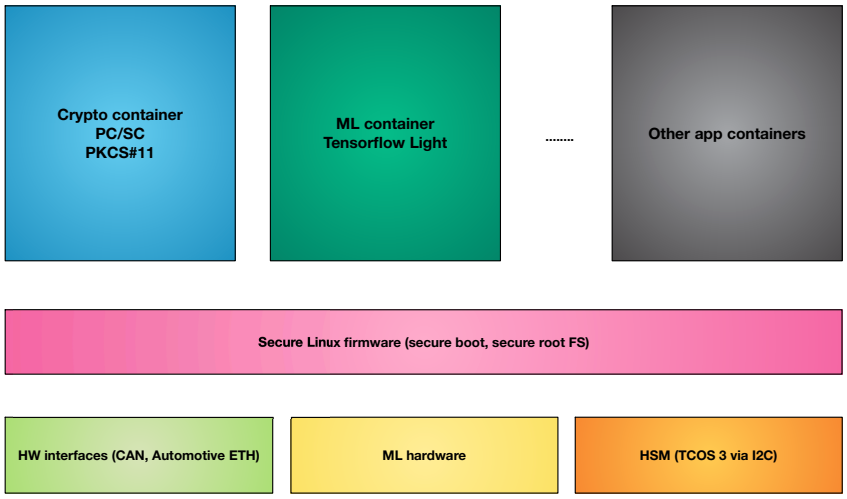


Figure 6 Anti-hacking device software architecture.

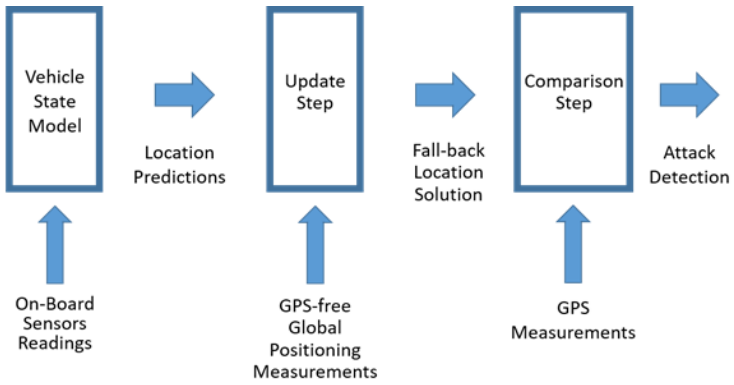
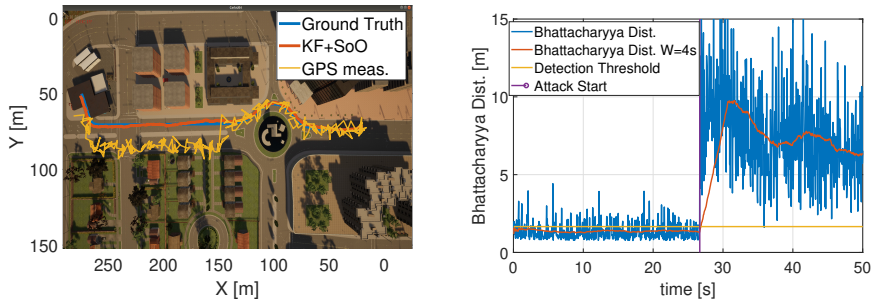


Figure 7 In-vehicle GPS location integrity check.



(a) Attack example

(b) Real Time GPS spoofing attack detection

Figure 8 Example of a GPS spoofing attack and real-time detection in CARLA simulator.

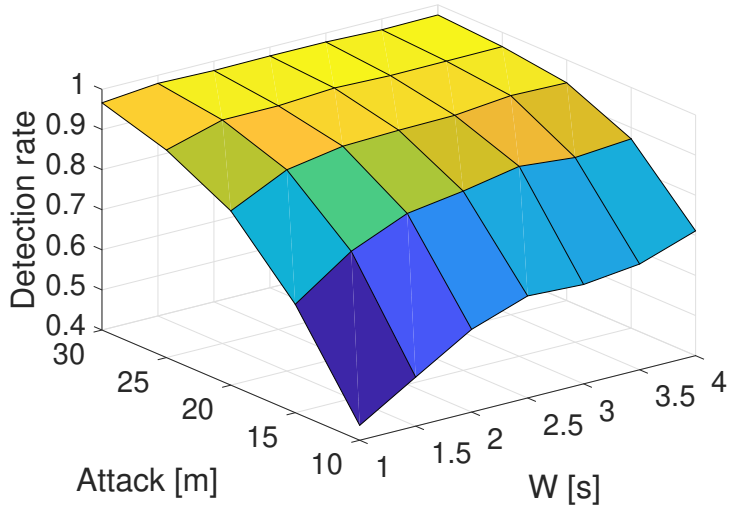
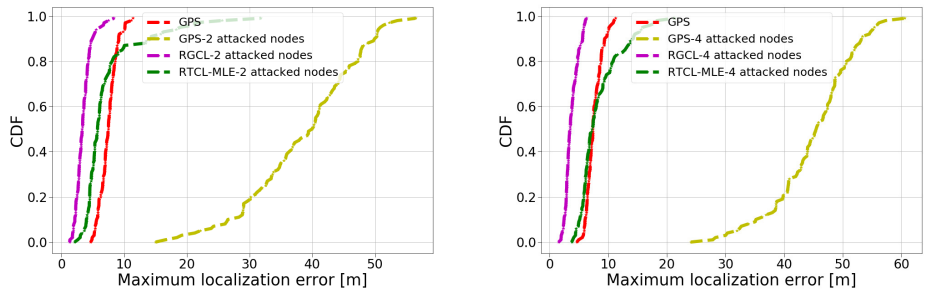
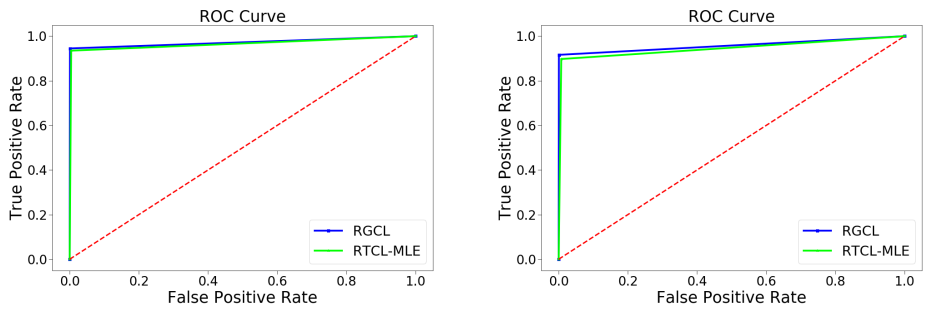


Figure 9 Attack detection rate depending on algorithm parameters.



(a) 2 attacked nodes (b) 4 attacked nodes

Figure 10 CDFs for maximum localization error with 2 and 4 vehicles under GPS location spoofing attack.



(a) 2 attacked nodes (b) 4 attacked nodes

Figure 11 ROC curves with 2 and 4 vehicles under GPS location spoofing attack.