PRISSMA Project
Plateforme de Recherche et d'Investissement pour la Sûreté
et la Sécurité de la Mobilité Autonome
04/2021 - 04/2024

[L5.2] REPPORT ON CYBER-THREAT ANALYSIS IN AUTONOMOUS VEHICULAR ECOSYSTEMS

RAPPORT SUR L'ANALYSE DES MENACES DE CYBERSECURITE DANS LES ECOSYSTEMES VEHICULAIRES AUTONOMES

Main authors: Jean CASSOU-MOUNAT, Virginie DENIAU, Richard DENIS, Christophe GRANSART, Sammy HADDAD, Reda YAICH

Keywords: Asset, Functional Architecture, Threats, Autonomous Driving System

Abstract. This document presents a state of the art of threats applying to Autonomous Driving Systems (ADS) using Artificial Intelligence (AI) software. For this study the first elements that we have define is a reference ADS system architecture. Thanks to this architecture we have been able to identify critical assets (data and functions) to be protected and for which security countermeasure will have to be tested and validated within the project. Then we have identified threats applying to the different component and functions of this architecture to finally analyze and select those threats for which security counter measures will have to be assess and validated.

Résumé. Ce livrable présente l'état de l'art des menaces s'appliquant aux systèmes de transport autonomes (STA) à base d'intelligences artificielles (AI). Pour réaliser cette étude, nous avons commencé par définir une architecture fonctionnelle de référence pour les STA dans laquelle nous avons identifié les données et fonctions critiques à protéger en lien avec l'IA. Pour chacun de ces biens, nous avons alors identifié l'état de l'art des menaces pour finalement fournir une analyse identifiant parmi ces menaces lesquelles impliquent la nécessité de valider les contre-mesures associées dans le cadre du projet.

TABLE OF CONTENT

Table of content 2
Abbreviations 4
1 Introduction 5
2 Reference architecture 6
2.1 System components 6
2.2 Users 8
2.3 Data and functional assets 10
3 State of the art 16
3.1 Communication threats 18
3.1.1. In-Vehicle Networks – CAN BUS 19
3.1.2. External stacks – V2X (ITS-G5 and cellular) 24
3.2 Threats on exteroceptive sensors 31
3.4 Threats to Artificial Intelligence and Machine Learning (IRT) 44
3.4.1. Categories of Adversarial Machine Learning 44
3.4.2. Adversarial Examples Generation 45
3.4.3. Taxonomy of adversarial examples 45
3.4.4. White-box adversarial examples 47
3.4.5. Black-box adversarial examples 50
3.4.6. Defences against adversarial examples 52
3.4.7. Reactive defences 52
3.4.8. Input reconstruction 52
3.4.9. Adversarial examples detection 53
3.4.10. Proactive defences 53
3.4.11. Gradient masking 53
3.4.12. Adversarial training 53
3.4.13. Defensive distillation 54
3.5 Documents with global risk identification for related systems 55
3.5.1. UNCE - R155 – Annex 5 55
3.5.2. ETSI ITS TVRA 55
4 Threats 56
4.1 Threat agents 57
4.2 PRISSMA threat scope analysis 58
References 70
Annex - R 155 threats identification table 73

Annex – ETSI ITS TVRA Tables 8 and 14 79

Abbreviations

For the purposes of the present document, the following abbreviations apply:

Abbreviations	Meaning
AA	Authorization Authority (synonym to PCA)
ADAS	Advanced Driver Assistance System
ADS	Autonomous Driving System
AI	Artificial Intelligence
AT	Authorization Ticket (synonym to PC)
AV	Autonomous Vehicle
CAM	Co-operative Awareness Message
C-ITS	Cooperative ITS
CRL	Certificate Revocation List
DENM	Decentralized Environmental Notification Message
EA	Enrolment Authority (similar to LTCA)
ECU	Electronic Control Unit
GNSS	Global Navigation Satellite System
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technology
ITS	Intelligent Transport System
ITS-S	ITS-Station
IVN	Internal Vehicle Network
OBU	On-Board Unit
PKI	Public Key Infrastructure
RSU	Road Side Unit
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	V2I or V2V

1 INTRODUCTION

This document presents a state of the art of threats applying to Autonomous Driving Systems (ADS) using Artificial Intelligence (AI) software. We first identified as much as possible known threats for ADS, before refining which ones are to be considered in the context of PRISSMA.

In fact, PRISSMA and more specifically the work package 5, aims at defining security validation mechanisms for AI based transport systems. The result of this study and the content of this deliverable will be used as an input for the future analyses of security objective definition to be done in the later task 5.3.

To do that, the first thing we had to define was a reference system architecture (section 2) to identify the critical assets (data and functions) to be protected (section 2.3) and thus for which security countermeasure will have to be tested and validated against known threats.

We first analyzed existing ADS architecture presented in the state of the art and identified the recuring elements composing them. From there, we managed to define generic components (e.g., vehicles, central station, roadside equipment, etc.) constituting such systems (section 2.1), their interactions, as well as users that interact with them (section 2.2), either to benefit their services (e.g., the transport system user or traveler) or manage it (PKI officers, remote vehicle managers, traffic controller, etc.).

With this generic architecture, we managed to define the associated set of assets to be protected (section 2.3), i.e., both critical data and functions necessary to provide the final travelling service safely and securely and performed a threat analysis by identifying the different known attacks and vulnerabilities for the different architecture components and technologies.

Finally, we identify the subset of those threats for which PRISSMA should provide security assessment mechanisms for the associated mandatory countermeasures.

2 REFERENCE ARCHITECTURE

In the context of PRISSMA we do not target one single use case or one specific system architecture. We tackle in a broader way the challenge to assess security of automated or autonomous vehicle systems using AI solutions.
Since we analyse here a generic system architecture, we provide only high-level description of the system users and functionalities, those can be implemented in many ways or decomposed in different manners corresponding to different technical implementation choices, however the presented architecture should still be a good representation or approximation of any such ones.
Our architecture is freely inspired by several reference documents providing different level of description of different systems components (vehicle network, ITS communication architecture, PKI, etc.) such as [1], [2], [3], [4], [5], etc; from which we have identified the main architecture elements freely combined in the architecture described in this section.
The system under study is the complete system handling and providing autonomous driving services on open roads. In this section we start by presenting the main components composing the system under study, then we detail the users connecting to it and using its services, after what, we present the system assets (critical data and function to be protected) and architecture components necessary to provide the full automated driving service to the travellers. We do not limit the scope of this description to AI functions, even though it is the main scope of PRISSMA, but rather we try to be as exhaustive as possible to be able later on to identify potential interaction and impact of other functions threats on AI.
Not all threats applying to the identified elements in this section will be kept in the PRISSMA scope. But this will be analysed and defined in section 4.

2.1 System components

We present here in Figure 1 and Table 1 the different components we identify to be able to run a full autonomous transport system. Based on that list we will latter identified assets, users and functions associated to these components.

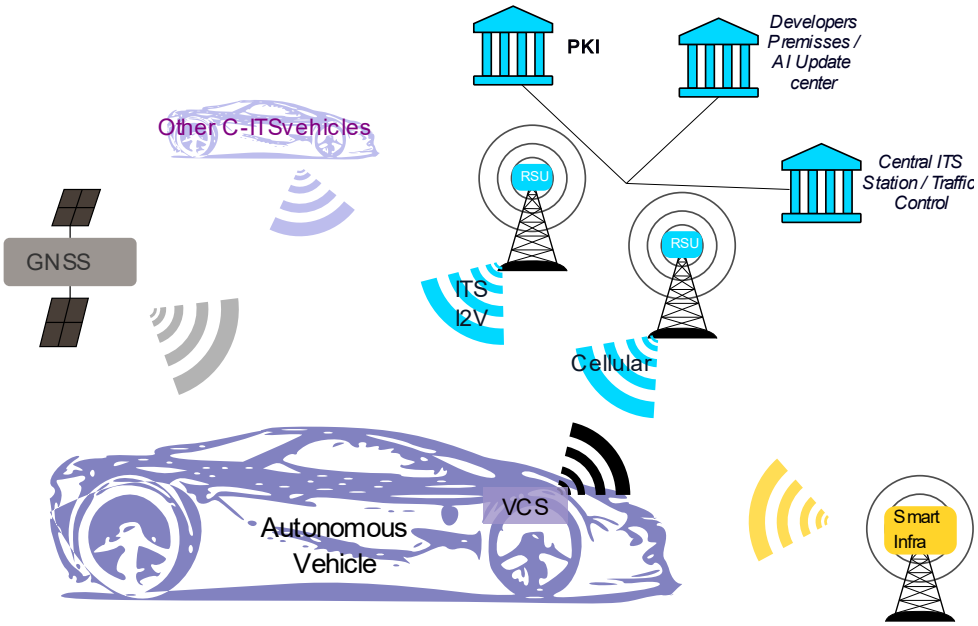


Figure 1 System components

Name	Description
Autonomous Vehicle (AV)	The complex system (vehicle) providing autonomous transportation means to the Passenger. The Internal AV Network (IVN) is composed of several component and/or subsystem. The following examples present the most common vehicle subsystems: • Body control providing equipment's related to passenger's compartment and trunk. • Power train includes ECU and sensors responsible of the transmission of the engine energy to propulsion • Chassis control includes ECU and sensors controlling the car actuators e.g., steering control, airbag control, braking systems. It may also include ADAS, but for the sake of our analysis we consider ADAS as a standalone car functional block. • Autonomous Driving System (ADS) • Global Navigation Satellite System (GNSS) receiver providing time and position information to the vehicle • Vehicle C-ITS Station (VCS) providing cellular, Wi-Fi or also 5G V2X connectivity to enable the vehicle ITS communications (e.g., CAM, DENM, SPAT, MAP, CPM, etc.) with other vehicles or the infrastructure (e.g. traffic and supervision centre, vehicle remote control facilities) to enhance its environment awareness, provide infotainment services and broadcast information. • Sensors allowing the vehicle to perceive its nearby environment e.g.: Lasers, cameras, radars, lidars, acoustic sensors. • IVN global network interconnection within the vehicle. The aforementioned elements or functionalities can be developed and organized in many different ways, but they all need to be interconnected somehow. The IVN contains all the communication connexions and links between the vehicle components as well as potential security countermeasure (e.g., gateways providing firewalling, authentication features, etc.
GNSS	Satellite systems providing time and allowing positioning to the autonomous vehicle.
Road Side Units (RSU)	Roadside equipment providing communication gateway functions to the Autonomous toward the Internet or cellular network thus including connectivity means to central ITS station, PKI, Developers premisses or the Internet in a more general. Also, RSU can include sensor providing data to the ITS network (camera, lidar, radar, connected traffic lights, remote sensors, etc.).
Central ITS station	Central ITS station gathers and provides ITS data to the vehicle and the rest of the infrastructure, this covers:

	• Traffic control centre sending and receiving traffic management data and maintaining a global model of the current traffic status • Remote control centre form which AV can be remotely controlled
Developers premisses / AI update repository	Online and offline IT system used to gather AI's training data, develop, and update AI models, provides AI update repository.

Table 1 PRISSMA reference architecture main components

2.2 Users

As for the previous and following elements of this section, we present here potential system user that may not all be part of all autonomous transport system but that are still most common potential user to be managed by such system. For instance, we know that PRISSMA should cover autonomous delivering system, and obviously in those systems the first identified user "Passenger" will not be part of it but again we present a generic set of users that should be commonly using or connecting to autonomous transports systems. This list is presented here in Table 2 and illustrated in Figure 2.

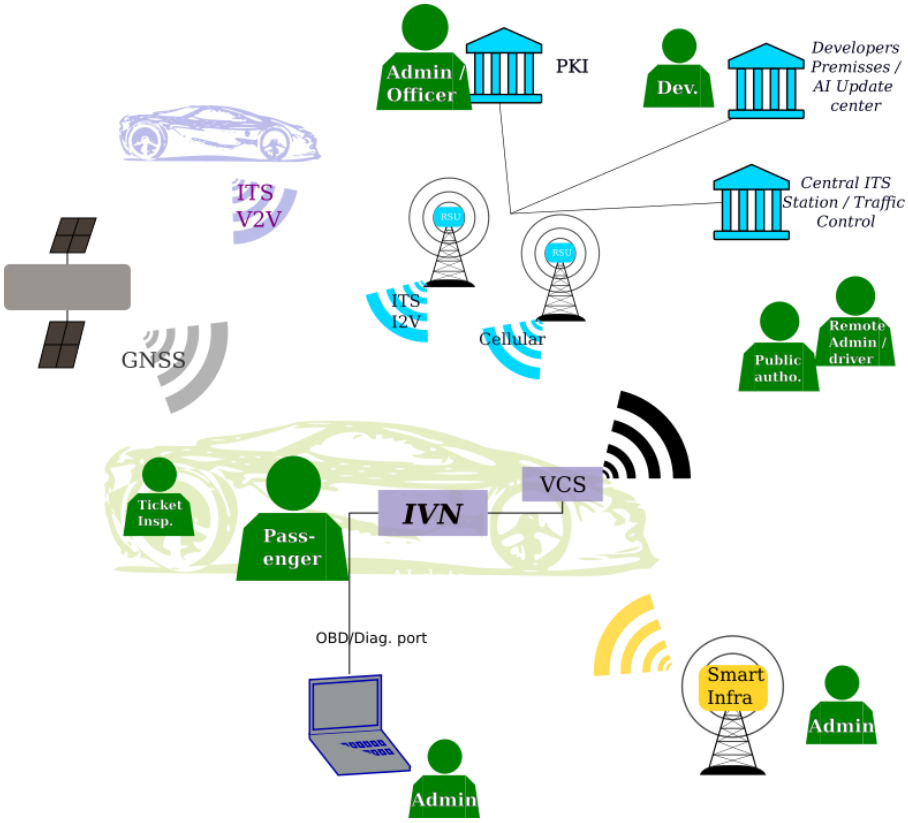


Figure 2 PRISSMA reference architecture users

Name	Description
Transport service	
Passenger	The traveller that uses the autonomous vehicle to perform its journey.

Local vehicle's manufacturer administrator / workshop maintenance	Administrator directly and physically connected to the vehicle managing and maintaining the vehicle software system performing regular administrative activities, e.g.: • Vehicle software configuration • Software update • Log review

Remote vehicle's manufacturer administrator / workshop maintenance	Administrator managing and maintaining the vehicle software system performing regular administrative activities, e.g.:
	• Vehicle software configuration
	  ○ Configure VCS connection to IVN, HSM, sensors, external ITS App
	  ○ Manages the different V2X communication parameters
	  ▪ Revocation/disabling of the VCS communications
	  ▪ Import, creation, update of certificates
	  ▪ Frequency of messages
	• Software update
	• Access and manages the audit traces produced by the VCS
	  ○ Configure, read, modify audit traces (logs)

Driver/operator	Responsible for actively monitoring the journey of the autonomous vehicle and taking driving actions (remotely or not) if required.
Payment system	
Ticket inspector	Personnel in charge of verifying travellers or goods transportation ticket or voucher, connecting having access to the connecter IT validation system through portable validator or other technical mean.
Transport system management	
Public authorities	Personnel representing local or national authorities (e.g., police, emergency services, traffic manager, etc.) providing inputs to the ITS systems (e.g. alerts) or access supervision data to perform their duty (e.g. accident identification for emergency intervention).
Traffic manager	Authority representative that handles traffic management at the global service level. They provide traffic management input (choice route selection, warning/road state information's, programming and re-programming traffic lights, etc.).
Trust system	
PKI administrators	Administrator of the PKI software and hardware, configuring and managing PKI elements (HSM, servers, etc.). Including following PKI configuration, e.g.: • Set cryptographic algorithms • Certificate revocation • Addition new CA certificate • Downloading new CTL or CRLs
PKI officers	Configures CA's policies, e.g., for ETSI standardized PKIs:

	• *'region'* of type Geographic Region as defined by [6] present or absence • *'appPermissions'* indicate message signing permissions, i.e., permissions to sign certificate response messages contained in a EtsiTs103097Data • *'certIssuePermissions':* this component shall be used to indicate issuing permissions, i.e., permissions to sign an enrolment credential / authorization ticket with certain permissions. as defined in [6]
AI lifecycle	
AI Developer	Responsible of the AI development and lifecycle. They gather and provides training data to the AI to generate initial models and updates to be push to the vehicle.
AI lifecycle	
Connected infrastructure manager	User managing the RSU configuration and software either via physical or remote connection.

Table 2 PRISSMA reference architecture users

2.3 Data and functions (assets)

In this section, we present the different elements constituting of the system under study. We identify assets, i.e., data and functions to be protected in the second step of our study against associated threats presented in section 4.2.
We first present the data produced and exchanged within the system and then the system functions. Those data and functions association with system architecture components are illustrated in Figure 3.
For each data identified in the system (Table 3 Data assets), we identify high-level security needs in terms of confidentiality, integrity, availability, and authenticity.

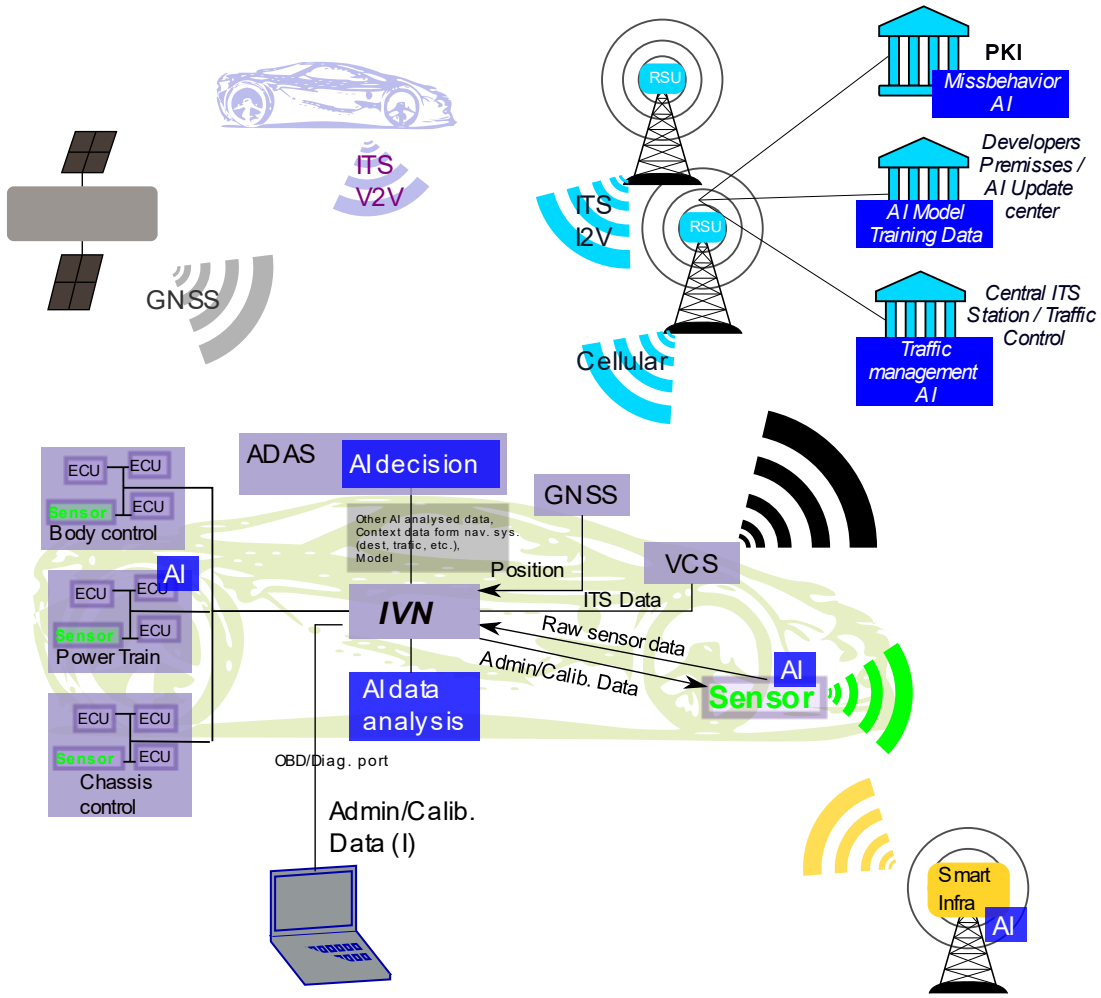


Figure 3 PRISSMA reference architecture functionalities and assets

Name	Description	Security needs
Keys		
Canonical Public Key	Public key generated by ITS-Station and used by the EA to verify EC request signature.	Integrity
Data encryption key	AES key used to encrypt requests and responses messages data.	Integrity, confidentiality
CA private keys	Private keys corresponding to the public keys contained in CA certificates issued by the PKI system. Theses keys are used for signature/encryption mechanisms.	Confidentiality, integrity
Certificates		
CA Certificates	This includes the root CA self-signed, EA, AA and MA certificates generated by the root CA.	Availability

Enrolment Credential (EC)	EC is a certificate that contains a unique name, a public key and other attributes as defined in [6] section 7.2.2 and [7] section 6.2.3.2.1.	Integrity
Authorization Ticket (AT)	AT is a pseudonym certificate that do not contain any identification information but public key(s) and other attributes as defined in [6] section 7.2.1 and [7] section 6.2.3.3.1	Integrity
TLM certificate	Self-signed certificate managed by EU	Availability
Station registration data		
Canonical ID	This information is stored at initial registration of the ITS station under the responsibility of the manufacturer. The canonical ID shall contain a substring identifying the manufacturer or operator to make uniqueness of this identifier possible.	Integrity, confidentiality
ITS-S Profile	The profile information for the ITS-S that may contain an initial list of maximum appPermissions (ITS-AIDs with SSPs), region restrictions etc; which may be modified over time.	Integrity
Tag	HMAC-SHA 256 of the keys to be certified.	Confidentiality, integrity
HMAC key	Key used to compute Tags sent with AT requests.	Confidentiality, integrity, availability
CA Network addresses	URL used to communicate with the CA.	Integrity, availability
DC network address	URL used to communicate with the DC.	Integrity, availability
CPOC Network address	URL used to communicate with the DC.	Integrity, availability
Policies		
Certificate Policy configuration data	Configuration data defining the role and duties of the PKI entities, include definition of parameters for: issuance, publication, archiving, revocation, renewal. This includes the certificate profiles.	Integrity, availability
Trust lists		
CRL	This list contains all information about revoked entities and need to be protected from any malicious change and we need to assure the integrity of this list as defined in [7].	Integrity, availability
CTL	This list contains all information about trusted entity certificates (CA), using	Integrity, availability

	the format and properties as defined in [7].	
ECTL	This list contains all information about root CA certificates (certificates, URL to access to the CPOC, ...) as defined by [7].	Integrity, availability
PKI services		
Software/Execution of the software	Correct execution of the TOE function to provide the correct services.	Integrity
Misbehaviour detection		
Misbehaviour Report (MR)	Reports send by the ITS-S to the MA to provide information regarding a possible misbehaving ITS-S [8].	Integrity, availability
ITS data		
X2V Safety sensitive ITS application data	Data used by **ADAS** as **input** to plan and control functions execution (like adapted CAM, PMM, CPM, MAP, etc.). Also includes **remote control commands.**	Integrity, Authenticity, Availability
X2V Sensitive ITS application data	Application data containing sensitive user information (e.g., credentials, web browsing, etc.) or application configuration data. This includes all communication data with the PKI authorities [7] [9]. This also may include context data for navigation context (destination, traffic density, etc.).	Confidentiality, Integrity, Authenticity.
X2V Informative ITS application data	Informative ITS data related to the vehicle and the road environment, e.g.: vehicle type, speed, emergency braking, road hazard warning, etc. This includes at least CAM [10], DENM [11], CPM.	Integrity, Authenticity.
X2I Safety Sensitive ITS application data	ITS information having an impact on the vehicle behaviour or trajectory e.g., remote vehicle control	Confidentiality, Integrity, Authenticity, availability
X2I Sensitive but not safety critical ITS application data	E.g., user credentials, web browsing, etc.	Confidentiality, Integrity, Authenticity
X2I Informative ITS application data	Informative ITS data related to the vehicle and the road environment, e.g.: vehicle type, speed, emergency braking, road hazard warning, etc.	Integrity, Authenticity.
LDM	The Local Dynamic Map (LDM) is an in-vehicle ITS station's dynamically updated repository of data relating to local driving conditions. It includes	Integrity, Authenticity

	information received from on-board sensors and from CAM and DNM messages.	
Sensor Data	Information gathered and analysis by vehicle sensors or RSU sensors.	Integrity
ITS software	Code of the different ITS elements, including ADAS and AI software.	Integrity
GNSS		
Time and position	Time and position data received by ITS-S from GNSS system	Availability, Integrity
Travelers apps and ticketing system		
User traveling and ticketing information	All data used for communication with the user apps, payment, and ticketing services	Confidentiality, Integrity, Authenticity.
IT management		
Configuration and calibration data	Data provided by administrators to configure IT components of the AV system.	Integrity, Authenticity

Table 3 Data assets

In Table 4, we identify the critical functions performed by ADS and we link them to potential AI software which is PRISSMA focus. This link is not trivial since implementation could potentially use AI for any system function. But we identified most common AI functions from PRISSMA partners developments and available state of the art architecture.

Name	Description	Security needs	AI
PKI			
Certificate request management	Receives certificates requests (EC, AT), generates adequately new certificates, and send them to the requester.	Integrity, availability, confidentiality	No
Trust list management	Updates trust list and provide distribution point for vehicle trust list updates.	Integrity, availability.	No
Misbehaviour management	Reception and analysis of misbehaviour reports and adequate PKI reaction management (e.g., certificates revocation via CRL or CTL update).	Integrity, availability	Yes
Developer servers			
AI software or model update	Provides repository for vehicle AI updates.	Integrity, availability, confidentiality	Yes
Field data collection	Collect vehicle reports on AI behaviour on field to help the developer to update AI models.	Integrity	Yes
Central ITS			

Traffic management	Collect system sensors data (form vehicles, roadside units, human agent, etc.) and send traffic status data or journey management data to the system components or vehicles.	Integrity	Yes
Vehicle remote control	When requires take over autonomous vehicle control to allow human agent to operate remotely the vehicle, especially in emergency cases	Integrity, availability	No
GNSS			
GNSS	Provides time and positioning services to the component of infrastructure.	Integrity, availability	No
Roadside infrastructure			
V2X support	Allows communication between vehicle and the infrastructure including PKI, developers and central ITS station communication with the vehicle.	Integrity, availability	No
Road infrastructure monitoring and environment perception	Data collection by the road infrastructure either based on the roadside component state audit (e.g., red light status) or roadside sensors detection (temperature, cameras footage, etc.).	Integrity, availability	Yes
Vehicle			
Journey	User or goods transportation from one physical point to another by the autonomous vehicle. This service is based on the vehicle motion capabilities (e.g., power train or chassis control systems).	Availability	Yes
LDM	Local dynamic map maintenance allowing the vehicle to maintain a precise enough awareness of its surrounding to enable autonomous driving.	Integrity, availability	No
V2X communication	Communications management with the infrastructure and other road used (e.g., exchange of standardized messages CAM, DENM, SPAT, MAP, CPM, etc.)	Integrity	No
ADS	Control of vehicle driving (or delegated driving) to achieve user journey. This is composed of several different functions depending on architectures: • Localization based on sensor data including lane positioning, road localization	Integrity	Yes

	• Environment perception via road network modelling, traffic flow identification, scenery modelling • Planning and control including mission planning, guidance (situation assessment and behaviour planning: lane crossing, lane changes, driving, etc.)		
Environment perception	Raw sensor data (radar, LiDAR, camera, lasers, etc.) collection or analysis by the sensor itself used to get environment awareness to be forwarded to the ADS.	Integrity, availability	Yes
Audit and diagnostic	Log generation for the vehicle software and hardware main events (start-up/shutdown, errors, logins, etc.), vehicle diagnostic, sensors data recording, etc.	Integrity, availability	Yes
Remote control and management	Remote communication between vehicle managers or drivers and the vehicle or its passengers for: • Communication with passengers • Vehicule evacuation • Delegated driving supervision • Remote driving	Integrity, availability	No
Travelers apps and ticketing system			
Ticketing and payment validation	Journey planification apps, traffic information, ticket validation means, etc.	Integrity, availability	No

Table 4 Functional assets

3 STATE OF THE ART

General threats in V2V (from Vehicular communications) are well presented in [12] which propose a good survey of known attacks in 2020. In this article the authors present among other things threats on privacy and integrity of V2V, general threats in V2I that we will further study in this section.

The authors identify that **privacy and integrity of V2V** communication can be compromised by such attacks as illusion, bogus information, sybil, timing, impersonation, and alteration/replay attacks.

- **Illusion attack**: During an illusion attack, attackers create false traffic events by altering vehicle sensor readings to trigger the sending of false traffic information messages [13]. Since these messages are sent from a legitimate source, other nodes on the network may receive this data and make erroneous decisions. The illusion attack is one of the tougher attacks to detect because forms of authentication, such as node registration or signature verification, will not work, as the data is sent from an authorized user.
- **Bogus information attack**: During a bogus information attack, attackers generate bogus traffic information and make other vehicles choose different paths, freeing up the

road for themselves [14]. The bogus information attack can be performed on various wireless networks at the same time, thus routing the whole path from source to destination for the attacker. The attacker's vehicle sends bogus information to Vehicle A and Vehicle B. The vehicles change their lanes or even their routes assuming that there is heavy traffic ahead of them, thereby freeing up the road for the attacker.

- **Sybil attack**: In a sybil attack, a single intruder node can declare itself as multiple nodes, eventually leading to extensive damage to network topologies and consuming large amounts of bandwidth [15]. The sybil attack is one of the most hurtful and dangerous attacks possible for vehicular ad-hoc networks. Since many vehicular networks are implemented with no certificate authorities or digital signatures, the feasibility of a sybil attack is quite high.
- **Timing attack**: In timing attacks, a malicious vehicle receives a message, adds some time delay, and then forwards the message to other vehicles, thus leading to improper timing information [16]. This attack can be devastating to vehicular networks, which depend upon real-time applications. The attacker had the obligation to communicate Vehicle A's positional information when Vehicle B changed the lane. But the attacker adds a time delay to the information and delivers the information only when the Vehicle B changes its position to B', leading to an accident.
- **Impersonation attack**: Impersonation attacks are carried out by providing a vehicle with a false identity [17]. Impersonation is detrimental to the legitimacy of the overall vehicular network architecture and is specifically hurtful in the case of an accident since the vehicle under investigation becomes untraceable.
- **Alteration/Replay attack**: As the name suggests, an alteration/replay attack occurs when an attacker employs any previously generated frames to send and communicate with other nodes, with or without alteration [18].

The second category of threats of interest for us in that survey is the **"General threats in V2I"** category.
Within V2I environments, vehicle On-Board Units (OBUs) communicate with Roadside-Units (RSU) to relay information about road conditions. RSUs can authenticate OBUs and grant them Internet access [19]. Within a VANET, both OBUs and RSUs are vulnerable to malicious activity. This section discusses known threats to V2I communication and proposed countermeasures.
Islam et al. [20] identify potential attacks on V2I. One attack is the distributed denial of service attack, which is described as the unnecessary transmission of information by an attacker that renders road-side unit software unable to function. Other attacks include impersonation attacks, which enable attackers to pose as RSUs or OBUs; malware attacks, which can infect the roadside unit software; and eavesdropping attacks, which allow attackers to gain access to confidential information.
Kim et al. [21] discuss the Road-Side Unit (RSU) replication attack, which moves an RSU or replicates it at another location to provide incorrect traffic information and perform erroneous services. There is also a discussion on trust authorities, which evaluate the authenticity of nodes within the VANETs, to examine eavesdropping attacks and monitor vehicle locations.
[12] provides a table summing up the different attack and their feasibility that we copy here:

Attack	Property	Ease of attack	Detection probability
Eavesdropping	Confidentiality	High	Low
GPS Spoofing	Authentication, Privacy	High	Low
Alteration/Replay	Integrity, Authentication	High	Low
Magnetic	Privacy, Integrity, Availability, Real-time Constraint	High	Low-Driver, High-System
Identity tracking	Location, Privacy	High	Low-at High Traffic Density
Sybil	Authentication, Availability	High	Moderate
Denial of service	Authentication, Availability	High	High
Timing	Availability, Real-time Constraint	High	High
Bogus information	Integrity, Authentication	Moderate	Low-Driver, Moderate-System
Black hole	Availability, Confidentiality, Integrity	Moderate	Moderate
Man-in-the-middle	Confidentiality, Integrity, Authentication	Moderate	Moderate
Injection	Integrity	Moderate	Moderate-Driver, High-System
Blinding	Privacy, Integrity, Real-Time constraint	Moderate	High
Illusion	Authentication, Integrity	Low	Low-Driver/System
Impersonation	Integrity, Authentication	Low	High

The rest of this section focuses on two main threat types: communication threats and sensors threats and then focuses on 2 important references for which we provide in annex the extensive sources of threats they have identified for ITS systems.

3.1 Communication threats

Communication stack is, by definition, a way for assets to exchange information. In an in-vehicle network, this can lead to severe issues such as critical ECU reprogramming and taking control of the vehicle over the different networks.

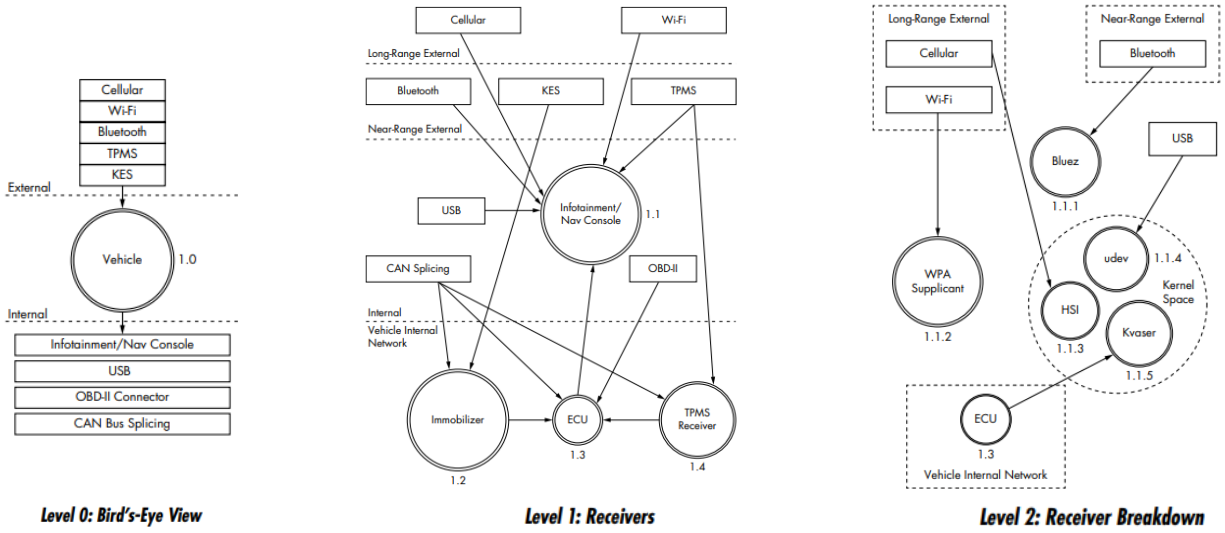


Figure 4: Bird's-Eye view of internal and external connections (left: attack surface at level 0, centre: specifies the receivers of level 0 input, right: focuses communication on Linux based processes) [12]

Attack surfaces are a key element of threat modeling and risk analysis. **Figure 4** above provides a binary representation of ITS vehicle connection: internal communications and external communications. We focus the following state of the art on one technology per stack.

3.1.1. In-Vehicle Networks – CAN BUS

As PRISSMA aims to study the impact of AI in autonomous vehicles, the bus that interconnects the ECUs seems relevant to focus our attention on.
Erreur ! Source du renvoi introuvable. presents a typical In-Vehicle Network (IVN), consisting of ECUs and CAN buses connecting the different IVN subnetwork (e.g., body control, power train, etc.).

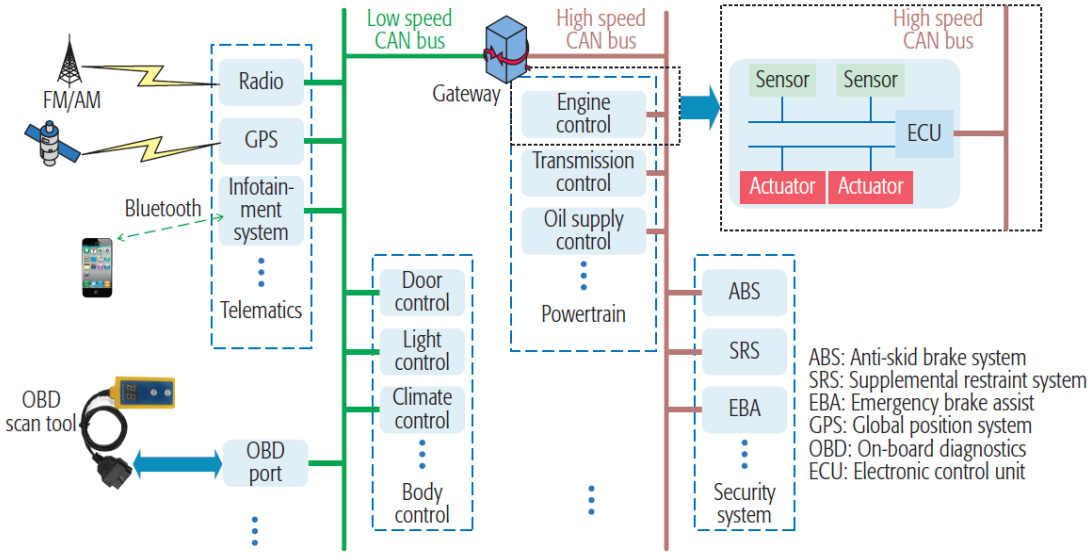
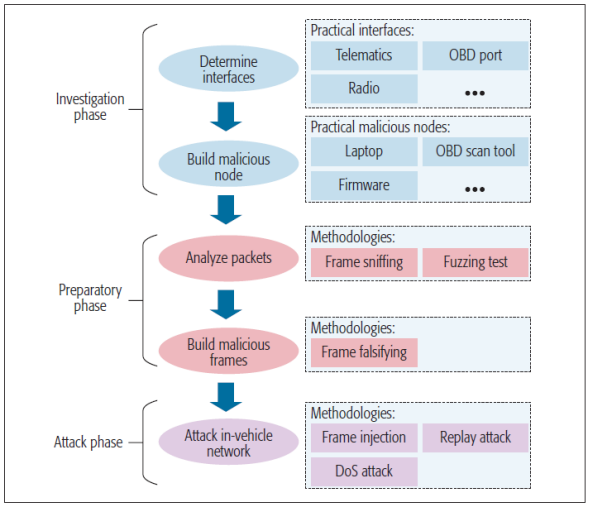
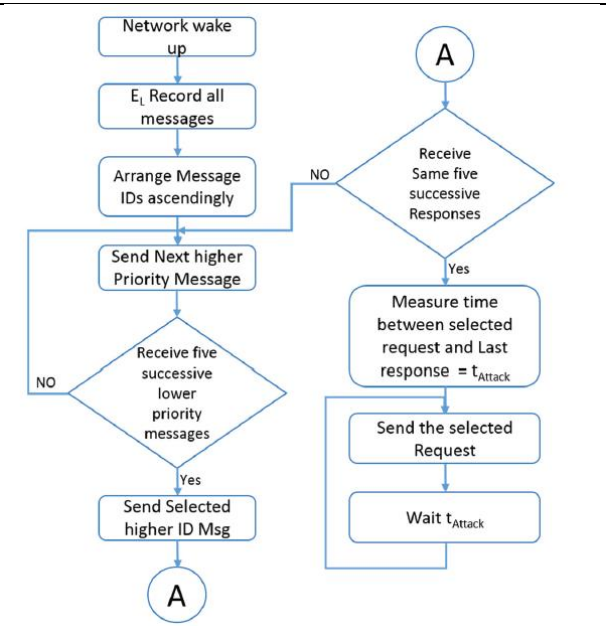


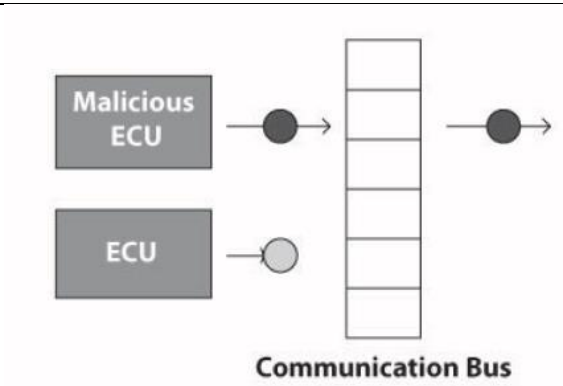
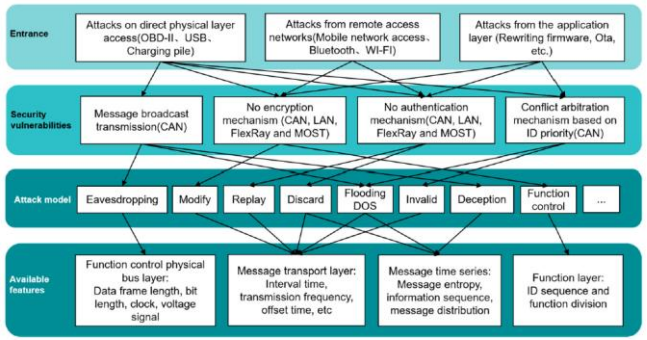
Figure 5 CAN network architecture [23]

CAN was developed in the early 1980s by Robert Bosch GmbH. Because of its high efficiency and low cost, the International Standardization Organization (ISO) established CAN as the international standard in 1993. The CAN protocol has several intrinsic vulnerabilities, such as broadcast transmission, no authentication, no encryption, ID-based priority scheme and available interfaces. These vulnerabilities make IVNs vulnerable to malicious attacks. Mainly when any of the component connected to the CAN bus is corrupted, it can send any information that the other equipment won't be able to verify due to this lack of security mechanisms.

Article/Paper	Target	Type of threat	Impact on AV percetion	Illustration	Security countermeasures (proposed by the authors)[1]
In-Vehicle Network Attacks and Countermeasures: Challenges and Future Directions, 2017 [23]	CAN	- Frame sniffing - Frame falsifying - Frame injection - Replay attack - DoS attack	- Confidentiality - Integrity (false detection of obstacles, add false ECU) - Availability (undetected objects)		- Enhancing In-Vehicle Network Security by Encryption and Authentication - Separating Potential Attacking Interfaces from In-Vehicle Networks

[1] Based on the sensors models under study (with possibly specific characteristics/performance compared to sensors with similar technology)

Replay Attack on Lightweight CAN Authentication Protocol, 2017 [24]	CAN	- Replay attack	- Integrity (False detection of obstacles)		- Enhancing Lightweight Authentication Protocol (LCAP) against the replay attack with a three-stage solution: o Refusing duplicate channel requests o Reconstruct the channel request message in such a way that represents both sender and receiver ECU IDs o Create a challenge-response procedure

In-Vehicle Networks: Attacks, Vulnerabilities, and Proposed Solutions, 2015 [25]	CAN	- Control override	- Integrity / availability (change priority of ECU messages, add false ECU)		- Use of Firmware update Over the Air (FOTA) to remove OBD-II port
An Overview of Vehicular Cybersecurity for Intelligent Connected Vehicles, 2022 [26]	CAN	- Frame sniffing - Message playback - Camouflage - DOS attack - Sniffing	- Integrity - Confidentiality	 **Figure 3.** Relationship between attack entry, security vulnerability, and exploitable features of intrusion detection in vehicle networks.	• Building a more secure vehicle CAN network intrusion detection system through an advanced machine learning algorithm to improve threat detection • Still some research in progress on this subject: ○ Data encryption (e.g., with lightweight AES) ○ Message authentication (e.g., TESLA, MAuth-CAN, one-way hash chain) ○ Intrusion detection (via deep neural network)

<table>
<tr>
<td>S2-CAN: Sufficiently Secure Controller Area Network, 2021 [27]</td>
<td>CAN</td>
<td></td>
<td>- Confidentiality and authenticity</td>
<td>
Table 1: Comparison with related approaches
<table>
<tr><th></th><th>Protection</th><th>Algorithm</th><th>HW/SW</th><th>Bus Load</th><th>Latency</th><th>MAC Length</th><th>Security Level</th></tr>
<tr><td>CaCAN [28]</td><td>Authenticity + Freshness</td><td>SHA256-HMAC</td><td>HW+SW</td><td>+100%</td><td>+2.2-3.2μs</td><td>1 Byte</td><td>2^7</td></tr>
<tr><td>IA-CAN [21]</td><td>Authenticity</td><td>Randomized CAN ID + CMAC</td><td>SW</td><td>+0%</td><td>8bit: +72ms
32bit: +150μs</td><td>1-4 Bytes</td><td>2^7-2^{31}</td></tr>
<tr><td>vatiCAN [33]</td><td>Authenticity + Freshness</td><td>SHA3-HMAC</td><td>SW</td><td>+16.2%</td><td>+3.3ms</td><td>8 Bytes</td><td>2^{63}</td></tr>
<tr><td>TESLA [34]</td><td>Authenticity + Freshness</td><td>PRF+HMAC</td><td>SW</td><td>+0%</td><td>+500ms</td><td>10 Bytes</td><td>2^{79}</td></tr>
<tr><td>LeiA [37]</td><td>Authenticity + Freshness</td><td>MAC</td><td>SW</td><td>+100%</td><td>N/A</td><td>8 Bytes</td><td>2^{63}</td></tr>
<tr><td>CANAuth [41]</td><td>Authenticity + Freshness</td><td>HMAC</td><td>HW+SW</td><td>+0%</td><td>N/A</td><td>10 Bytes</td><td>2^{79}</td></tr>
<tr><td>S2-CAN</td><td>Confidentiality + Authenticity + Freshness</td><td>Circular Shift + Internal ID Match</td><td>SW</td><td>+0%</td><td>+75μs</td><td>N/A</td><td>$\sim 2^{49}$</td></tr>
</table>
</td>
<td>- Enable a trade-off between performance and security</td>
</tr>
</table>

While several papers target CAN-related vulnerabilities through 2017, it is difficult to find more recent scientific evidence pointing in this direction. Lack of research on the subject or real improvements? In this sense, the article from the University of Michigan seems to point us towards the first reason, by designing an alternative CAN protocol to handle security issues.

Finally, the UNECE R155 regulation, as a final point, lists several vulnerabilities to be considered in its analysis of vehicle vulnerabilities:

- Sub-level threat n°11: messages received by the vehicle (for example X2V or diagnostic messages), or transmitted within it, contain malicious content
  - Vulnerability n°11.1: malicious internal (e.g., CAN) messages
  - Vulnerability n°11.3: malicious diagnostic messages
- Sub-level threat n°18: devices connected to external interfaces e.g., USB ports, OBD port, used to attack vehicle systems
  - Vulnerability n°18.1: external interfaces such as USB or other ports used as a point of attack, for example through code injection
  - Vulnerability n°18.2: media infected with a virus connected to a vehicle system
  - Vulnerability n°18.3: diagnostic access (e.g., dongles in OBD port) used to facilitate an attack, e.g., manipulate vehicle parameters (directly or indirectly)

In the same way, ENISA raises in its first attack scenario [1], as a high vulnerability, the possibility for an attacker to take control over the CAN bus by reprogramming ECUs.

3.1.2. External stacks – V2X (ITS-G5 and cellular)

Since 1999, US Federal Communications Commission (FCC) allocated 75 MHz of spectrum in the 5.9 GHz band to be used by automotive industry for C-ITS. In 2008, European Telecommunications Standards Institute (ETSI) allocated 30 MHz of spectrum in the 5.9 GHz band for ITS. ITS-G5 operates on the frequency band (5.9 GHz) from which its name G5 is derived.
On the other hand, mobile networks are evolving. LTE-V2X is expected to perform the transition path to 5G. But it still requires further examinations especially that in some vehicular use cases, it is needed to fulfil required latency and reliability to guarantee the efficiency of targeted C-ITS services.

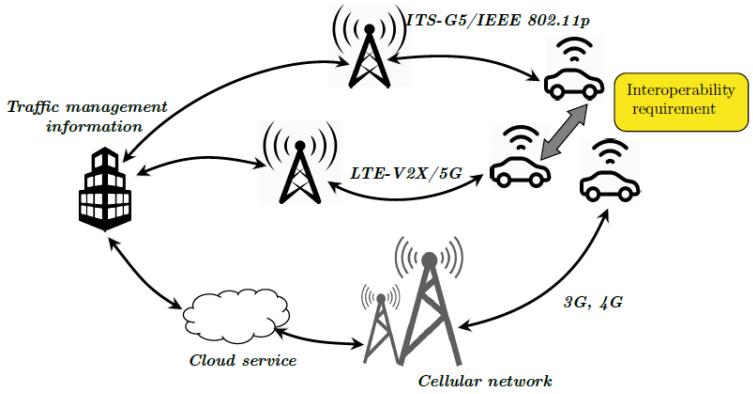


Figure 6 Heterogeneous architecture [28]

To tackle cybersecurity issues, ETSI and European projects have built specific methods and infrastructure to mitigate risks. A public key infrastructure has been built to secure V2X messages. The PKI consists of a root certificate authority (RCA), Enrolment authority (EA), and an authorization authority (AA), as shown in shown in the figure below. Each ITS station holds an asymmetric key pair where the public key is part of a digital certificate.

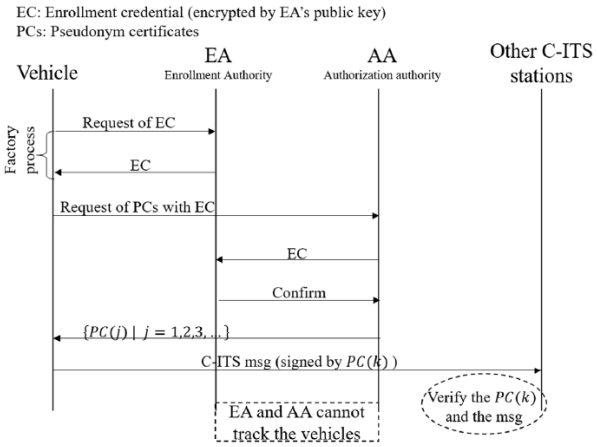


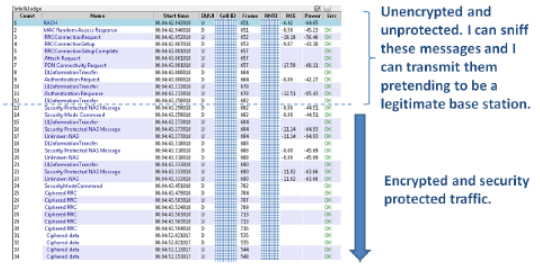
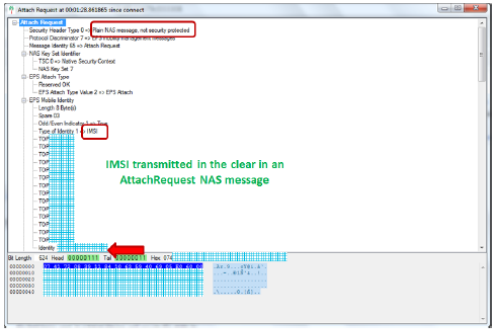
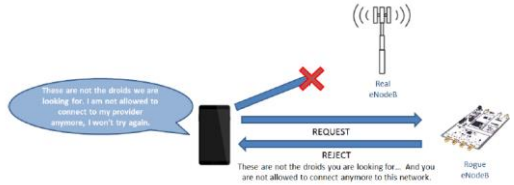
Figure 7 PKI assignment process

However, risk remains. Indeed, ITS projects do not have any strict security requirements regarding the operation of Certification Authorities and the C-ITS Stations. Several reports, articles and regulations point out the flaws in these systems.

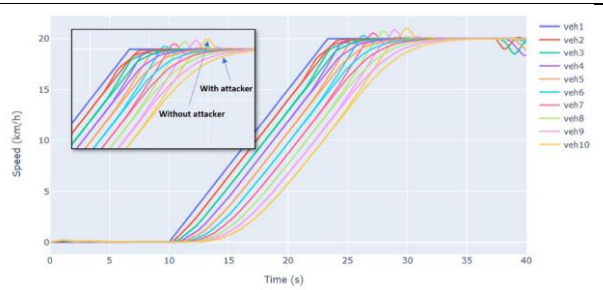
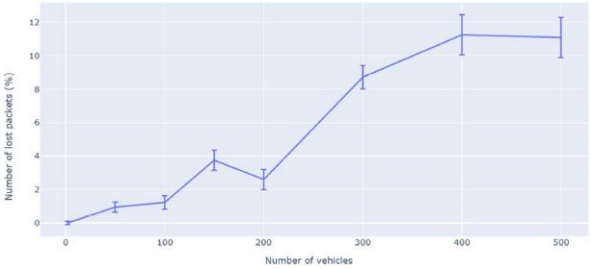
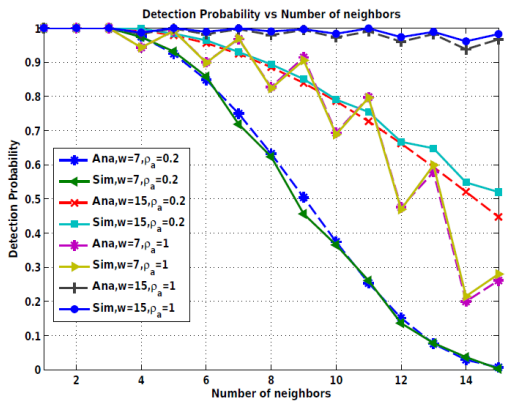
Reg/Article/Paper	Target	Type of threat	Impact on AV percetion	Illustration	Security countermeasures (proposed by the authors)[2]
UNECE - R155	Whole ITS system	Cf. section 3.4.1 and ANNEX - R 155 THREATS IDENTIFICATION TABLE			
ENISA – Good practices for security of smart cars (attack scenario n°2)	ITS-G5/C-V2X	- Illegitimate access to the car components	- Availability - Integrity - Confidentiality	/	1. Perform vulnerability surveys. 2. Third party testing of V2X applications. 3. Regularly assess the security controls and patch vulnerabilities. 4. Information sharing between different actors. 5. Adopt a holistic approach to security training and awareness among the employees. 6. Raise users' awareness. 7. Allow and encourage the use of strong authentication (e.g. multi-factor authentication). 8. Consider establishing a Computer Security Incident Response Teams (CSIRT). 9. Mitigate vulnerabilities or limitations of software libraries. 10. Protect mobile applications against reverse engineering and tampering of their binary code. 11. Securely store sensitive data on mobile devices.
Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA) (ETSI TR 102 893)	RSU and OBU	Cf. section 3.4.2 and Annex – ETSI ITS TVRA Tables 8 and 14			
C-V2X Security Requirements and Procedures: Survey and Research Directions [29]	C-V2X	- Fake nodes (that compromise privacy or provide false information or data/certificates) - RF congestion - Jamming (direct interference to transmitted signals)	- Availability - Integrity	/	- OTA updates - 3GPP states that "V2X network entities shall be able to authenticate the source of the received data communications, of data between V2X network entities shall be confidentiality and integrity protected and protected from replays" - Physical layer techniques: Use and leverage the diversity of the channel as well as interference mitigation techniques [16]. Embed signatures or other unique identifiers in messages to identify legitimate UEs and legitimate messages. - Cross-layer techniques: Revise scheduling, congestion control and other C-V2X procedures to make them more robust and aware of potential threats. For

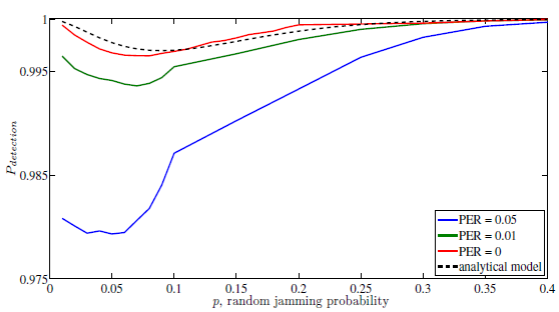
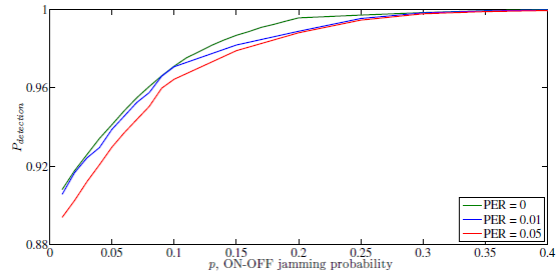
[2] Based on the sensors models under study (with possibly specific characteristics/performance compared to sensors with similar technology)

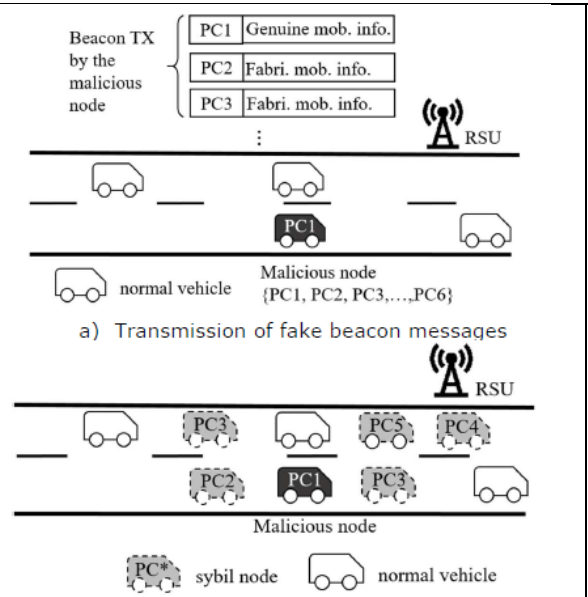
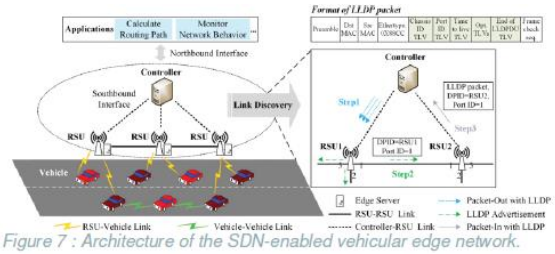
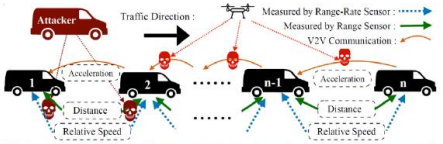
		- RF replay (confusion about inconsistent message content)			example, sched-ule messages for the purpose of enhancing distributed security awareness. - Edge and Cloud computing: Use the processing power of the Cloud to assist in determining fake transmitters and leverage edge computing resources to reduce latency and network congestion [17]. - Network-aided procedures: Have the network periodically notify UEs of legitimate UEs in the area using C-V2X or parallel communications protocols, such as regular LTE. - Radio environment map (REM): Have UEs built their REMs, where processed RF spectrum activity and the in-formation exchanged among UEs is stored and contrasted to identify anomalies. - Spectrum access system (SAS): The vehicular UEs and roadside units can regularly provide sensing information to a central SAS for RF anomaly detection. - Machine learning (ML): ML can be effectively employed to analyze huge amounts of data and classify it. When either the normal behavior is known or the attacks, or both, ML tools can be trained to distinguish between normal and abnormal activity. - Multiple sensor information processing: Use information from multiple sensors (cameras, radar, lidar, etc.) to validate C-V2X messages, weigh decisions and dynamically update node and information thrust metrics. Check message content for consistency using physical at-tributes, past messages, environmental information, etc.

Reg/Article/Paper	Target sensor	Type of threat	Impact on AV perception	Illustration	Security countermeasures (proposed by the authors)[3]
LTE security, protocol exploits and location tracking experimentation with low-cost software radio [30]	LTE	- Traffic capture	- Confidentiality - Availability (temporary block devices)	 Fig. 6. Initial connection to an LTE network sniffed from a real production network in the area of Honolulu, HI  Fig. 7. IMSI of the author's smartphone being transmitted in the clear triggered by a software-radio IMSI catcher  Fig. 8. Mobile device temporary block by a rogue LTE base station	/

[3] Based on the sensors models under study (with possibly specific characteristics/performance compared to sensors with similar technology)

Simulation of Cyberattacks in ITS-G5 Systems [31]	ITS-G5	- Packet congestion - Falsification attack	- Availability (undetected objects) - Integrity (false detection of obstacles)	 **Fig. 8.** Falsification attack - impact on speed  **Fig. 9.** Congestion packet loss	No countermeasure
Jamming Detection on 802.11p under Multi-channel Operation in Vehicular Networks [32]	ITS-G5	- Jamming	- Availability (undetected objects)	 Fig. 7. The influence of detection probability on number of vehicles	- Detection method for reactive jamming targeting

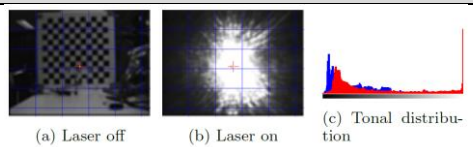
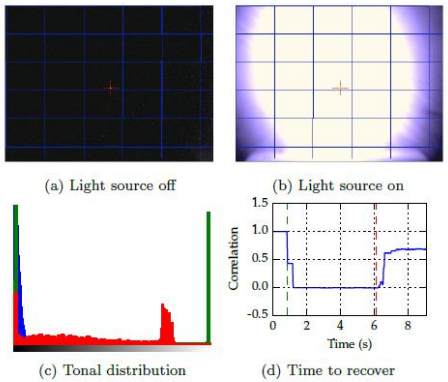
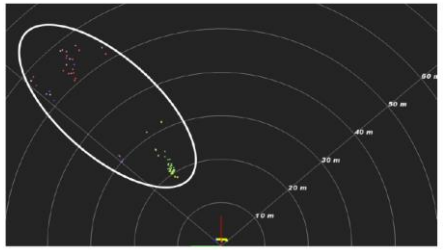
Real-Time Detection of Denial-of-Service Attacks in IEEE 802.11p Vehicular Networks [33]	ITS-G5	- Jamming	- Availability (undetected objects)	 Fig. 5. Attack detection probability for random jamming.  Fig. 6. Attack detection probability for ON-OFF jamming ($K = 2$).	- Jamming detector algorithm

State of the art on Cyberattacks in C-ITS, 2022 [34]	ITS-G5	- Sybil attack - DoS - Message injection - Poisoning attack - Platoon attack - Ransomware attack	- Availability - Integrity - Confidentiality	 Figure 2 : Sybil Attack in C-ITS  Figure 7 : Architecture of the SDN-enabled vehicular edge network.  Figure 9 : A vehicular platoon with potential threats against equipped sensors and V2V communication channels.	- Use expressive formal language for specifying protocols and their security properties - Use common security measures - Deploy an Intrusion Detection System at the high-level server - Sybil attack is a dangerous one, analyzing CAMs messages must be realized through statistical analysis for detecting the kind of attacks - Promote security assessment - BotVehicles (a set of malicious vehicles) are a real threat for the connected cars. A communication and CAM message analysis between vehicles are necessary.

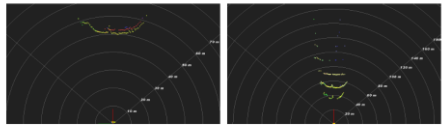
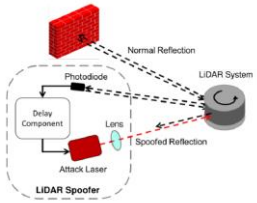
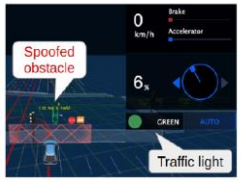
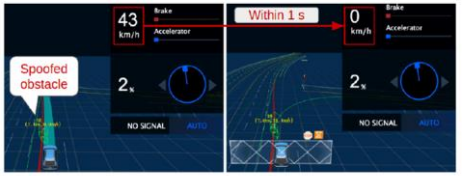
Through its recent arrival in the vehicular ecosystem, V2X components must be particularly scrutinized from a cybersecurity point of view. These devices will also be subject to numerous updates during their lifecycle, making analysis all the more necessary.

3.2 Threats on exteroceptive sensors

This section provides a (non-exhaustive) list of different articles/scientific papers dealing with threats/attacks targeting automated vehicles (AV), and especially their exteroceptive sensors (used by AV to construct a perception of their environment). Countermeasures proposed by authors of these papers are also summarized.

Article/Paper	Target sensor	Type of threat	Impact on AV percetion	Illustration	Security countermeasures (proposed by the authors)[4]
Remote Attacks on Automated Vehicles Sensors:Experiments on Camera and LiDAR [35]	Lidar Camera	Camera: - Denial of Service (Blinding attack/Confusing the auto-control) Lidar: - Spoofing (Replay/Relay attack)	- Integrity (false detection of obstacles) - Availability (undetected objects)	 (a) Laser off (b) Laser on (c) Tonal distribution Figure 4: Blinding MobilEye C2-270 with 650 nm laser  (a) Light source off (b) Light source on (c) Tonal distribution (d) Time to recover Figure 5: Auto-Exposure MobilEye C2-270 with 940 nm 5x5 LED matrix  Figure 7: Result of LiDAR relay attack. Light pulses are received from the left, and relayed from the right.	Camera: - Redundancy - Optics and materials (filter near-infrared light, filter out specific types of light) Lidar: - Redundancy - Random probing - Probe multiple times - Shorten the pulse period

[4] Based on the sensors models under study (with possibly specific characteristics/performance compared to sensors with similar technology)

				 (a) spoofing of one copy of the wall (b) spoofing of multiple copies of the wall Figure 11: Results of a LiDAR spoofing attack	
Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving [36]	Lidar	- Spoofing ("adversarial attack")	- Integrity (false detection of obstacles) - Availability (undetected objects)	 Figure 3: Illustration of LiDAR spoofing attack. The photodiode receives the laser pulses from the LiDAR and activate the delay component that triggers the attacker laser to simulate real echo pulses.  Figure 13: Demonstration of the AV freezing attack. The traffic light is turned green but the victim AV is not moving due to the spoofed front-near obstacles.  Figure 12: Demonstration of the emergency brake attack. Due to the spoofed obstacle, the victim AV makes a sudden stop decision to drop its speed from 43 km/h to 0 km/h within a second, which may cause injuries of passengers or rear-end collisions.	1) AV System-Level: - Filtering out the ground reflection in the pre-processing phase - Avoiding transforming 3D point cloud into input feature matrix or adding more features to reduce the information loss 2) Sensor-Level: - Detection techniques (Sensor Fusion) - Mitigation techniques (Reducing the receiving angle and filtering unwanted light spectra) - Randomization techniques (Firing laser pulses with unpredictable pattern, randomizing the laser pulses waveform, randomly turning off the transmitter to verify with the receiver if there are any unexpected incoming signals) 3) Machine Learning Model-Level: - Adversarial training and its variations

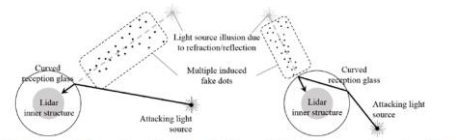
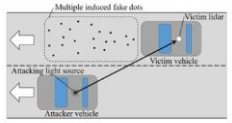
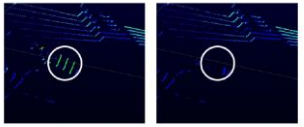
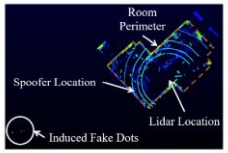
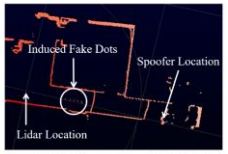
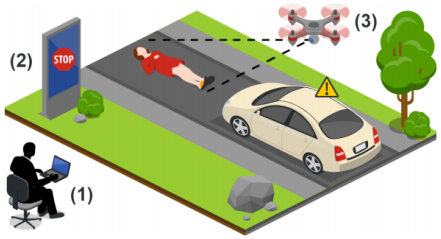
	Lidar	- Spoofing (replay attack) - Denial of Service/Jamming (Sensor saturation/blinding attack)	- Integrity (false detection of obstacles) - Availability (undetected objects)	 **Fig. 5.** Speculations of how the oblique incidence of light onto a curved reception glass induces fake dots in a direction different from that of the actual light source.  **Fig. 6.** Attack scenario exploiting a curved reception glass. The attacker and victim vehicles are heading the same direction, and the attacker obliquely illuminates the victim's lidar with a strong light source.  **Fig. 10.** VeloView output before (left) and after (right) exposure to a strong light source. We placed a metal plate ($41 \times 42cm^2$) in front of the lidar.  **Fig. 11.** VeloView output of the multiple induced fake dots.  **Fig. 12.** VeloView output of the fake dots closer than the spoofer. Note that the redder a dot, the closer it is to the lidar.	- Redundancy and Fusion - Saturation Detection - Reducing the Receiving Angle - Random-direction Pinging - Randomizing the Ping Waveform - Mitigating Curved Glass Effects
Phantom Attacks on Driver-Assistance Systems [37]	Camera	Spoofing ("phantom/illusion attack")	Integrity (false detection of objects/traffic signs/lines)	 Fig. 4: The Threat Model: An attacker (1) either remotely hacks a digital billboard (2) or flies a drone equipped with a portable projector (3) to create a phantom image. The image is perceived as a real object by a car using an ADAS/autopilot, and the car reacts unexpectedly.	- Detecting phantoms: validate the legitimacy of the object given its context and authenticity

Fig. 11: Embedded road sign in a Coca-Cola advertisement (a): full, (b) outline, and (c) embedding.

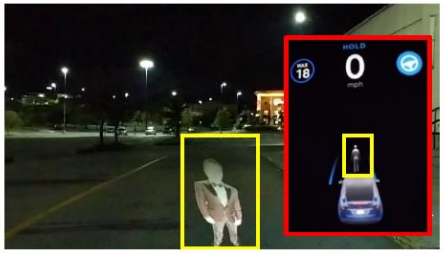


Fig. 12: Tesla's autopilot identifies the phantom as a person and does not start to drive. The red box contains a picture of the car's dashboard.

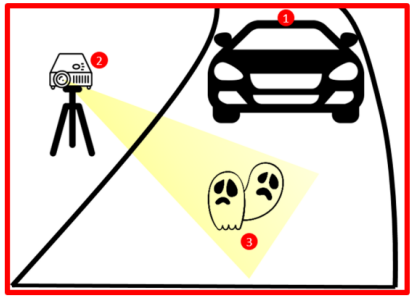
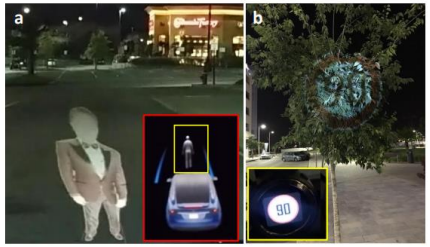


Fig. 21: Morphing Process Attacked Scene: Location 1 is where the front facing camera of the targeted car is located. Location 2 is where the attacker will perform the attack. Location 3 shows where the attacker wants to project the phantom.

				 Fig. 1: Perceptual Challenge: Would you consider the projection of the person (a) and road sign (b) real? Telsa considers (a) a real person and Mobileye 630 PRO considers (b) a real road sign. 	
Model Hacking ADAS to Pave Safer Roads for Autonomous Vehicles [38]	Camera	Spoofing ("adversarial attack")	Integrity (misclassification of traffic signs)		n/a

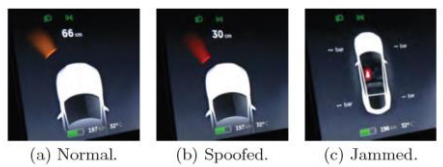
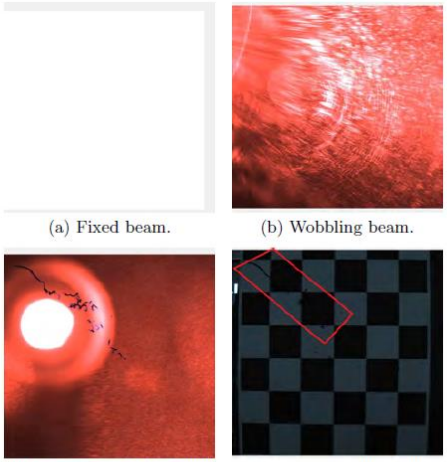
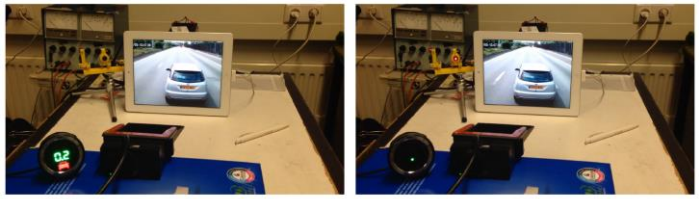
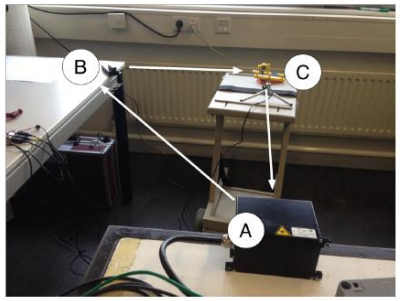
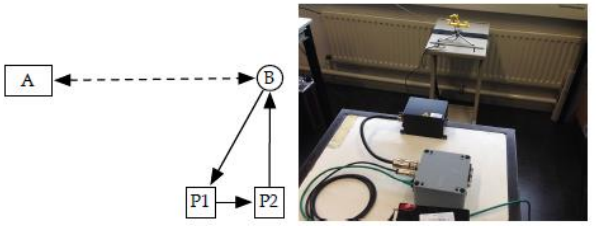
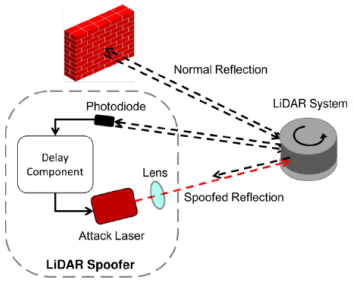
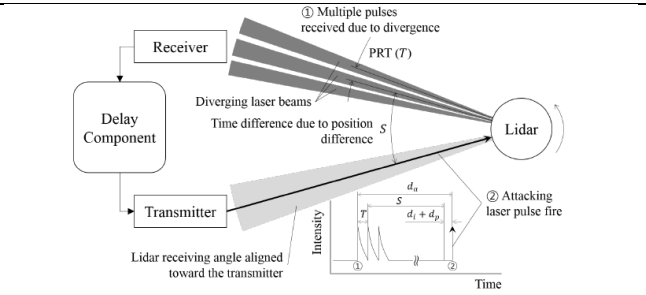
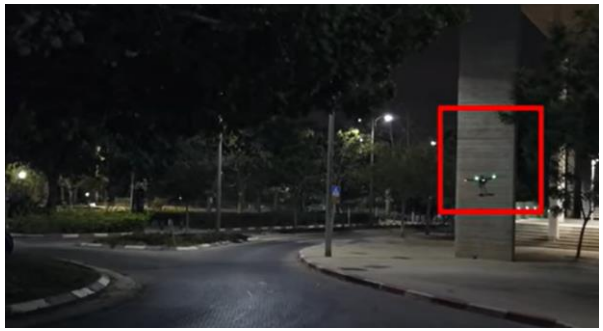
Can you trust autonomous vehicles: contactless attacks against sensors of self-driving vehicle [39]	Camera Radar Ultrasonic sensors	Ultrasonic sensors: - Spoofing - Denial of service (Jamming/Acoustic Quieting) Radar: - Spoofing (Relay) - Denial of service (Jamming) Camera: - Denial of service (Blinding attack)	- Integrity (false detection of obstacles) - Availability (undetected objects)	 (a) Normal. (b) Spoofed. (c) Jammed. Figure 4: Tesla parking distance display at normal, being spoofed, and being jammed[2].  (a) Drive gear. (b) Autopilot. (c) Jammed. Figure 9: Tesla dashboard display at drive gear, Autopilot, and Autopilot with radar jamming.  (a) Fixed beam. (b) Wobbling beam. (c) Damage caused by laser. (d) Damage is permanent. Figure 13: Blinding camera with confronted laser.	- Use multiple sensors for redundancy check

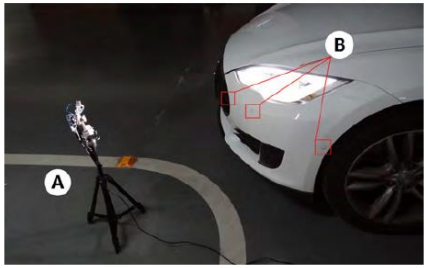
Table 5 Sensor threats state of the art

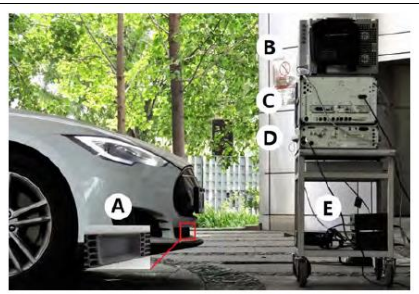
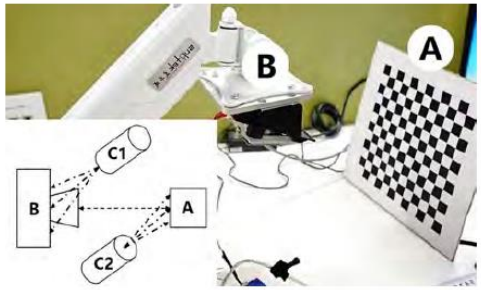
The Table 6describes the equipment that has been used in the previously mentioned papers/articles, or other equipment's that could possibly be used or abused by threat agents to generate attacks targeting exteroceptive sensors of automated vehicles.

Article/paper	Equipment (ab)used for attacks
Remote Attacks on Automated Vehicles Sensors:Experiments on Camera and Li-DAR **[1]**	• Attacks on CAMERA (*MobilEye C2-270*): - Laser 650 nm (*LEDSEE*) - LED spot 850nm (*Osram SFH4550*) - LED matrix 5x5 940nm (*LEDSEE*)  (a) Laser off, normal behavior of MobilEye C2-270 (b) Laser on, MobilEye C2-270 does not detect vehicle ahead Figure 13: MobilEye live blinding experiment. • Attacks on LIDAR (*ibeo LUX 3*): - Photodetector (*Osram SFH-213*) - Laser (*Osram SPL-PL90*) - 2 pulse generators (*HP 8011A and Philips PM 5715)*  Figure 6: Setup of a LiDAR relay attack  (a) Schematic (b) Actual setup Figure 9: Setup of a LiDAR spoofing attack

Adversarial Sensor Attack on LiDAR-based Perception in Autonomous Driving **[2]**	• Attacks on LIDAR (*Velodyne VLP-16*): - Photodiode *(OSRAM SFH 213 FA)* - Delay component (*Tektronix AFG3251 function generator*) - Laser driver module *(PCO-7114)* - Laser diode *(OSRAM SPL PL90)*  **Figure 3: Illustration of LiDAR spoofing attack. The photodiode receives the laser pulses from the LiDAR and activate the delay component that triggers the attacker laser to simulate real echo pulses.**
Illusion and Dazzle: Adversarial Optical Channel Exploits against Lidars for Automotive Applications **[3]**	• Attacks on LIDAR (*Velodyne VLP-16*): - Laser module (30mW, 905nm) - Power-adjustable laser module (800mW, 905nm) - Photodiode (*OSRAM SFH 213 FA*) - Pulsed laser diode (*OSRAM SPL PL90*) - Laser driver module *(PCO-7110-40-4)* - Delay component (*Agilent 33250A function generator*)

	 **Fig. 7.** Actual attack process: As the lidar rotates, multiple laser pulses, temporally separated by the PRT (T), are first captured by the attacker receiver (①). Then, after the actual required delay (d_a), the attacking laser pulse is fired (②). The graph below displays the temporal arrangement of events.
Phantom Attacks on Driver-Assistance Systems **[4]**	• Attacks on CAMERA (*MobilEye 630 PRO & Cameras of a TESLA Model X / HW 2.5*): 1) Projectors (mounted on a tripod or on a drone): - *Nebula Capsule* (portable projector with an intensity of 100 lumens and 854 x 480 resolution) - *AAXA P2-A LED projector* - *LG - CineBeam PH550 720p DLP projector* 2) Drones - *DJI Matrice 600* - *DJI Mavic* 3) Digital billboard (*device reference not provided*) 
Model Hacking ADAS to Pave Safer Roads for Autonomous Vehicles **[5]**	• Attacks on CAMERA (*MobilEye EyeQ3 on Tesla Model X & Model S with hardware pack 1*): - Adversarial stickers

	 - Black tape 
Can you trust autonomous vehicles: contactless attacks against sensors of self-driving vehicle **[6]**	• Attacks on ULTRASONIC sensors (*from Tesla Model S*) - Generator of 40 or 50 kHz square waves (*Arduino Uno board*)  **Figure 3: Setup of ultrasound experiment on Tesla Model S. A is the jammer, B is 3 ultrasonic sensors on the left side.** • Attack on RADAR (*from Tesla Model S*) - Signal Analyzer (*Keysight N9040B UXA + 89601B VSA Software*) - Oscilloscope (*SOS804A*) - Harmonic mixer (*VDI 100 GHz*) - Signal Generator (*Keysight N5193A UXG 10 MHz – 40 GHz*) - Frequency multiplier (*VDI WR10 75 – 110 GHz*)

	 Figure 8: Setup of Radar experiment on Tesla Model S. A is automotive Radar, B is oscilloscope, C is signal analyzer, D is signal generator, E is frequency multiplier, harmonic mixer, and their power supplies. • Attacks on CAMERA (*reference not provided*) - LED spot (*reference not provided*)  Figure 11: Setup of camera blinding experiment. A is a calibration board, B is a camera, C1 and C2 are laser emitters.
n/a	• Attacks on RADAR - Radar echo generator (*Rohde &Schwarz R&S AREG800A and AREG100A*) - Antenna array (*R&S QAT100*)

	 Such equipment is designed for test & validation of automotive radars (and ADAS/AD systems), but could possibly be abused to generate attacks on radar (e.g. spoofing attack: creating "phantom" objects with various characteristics, e.g. size, velocity, etc.).

Table 6 Equipment required for sensor attacks

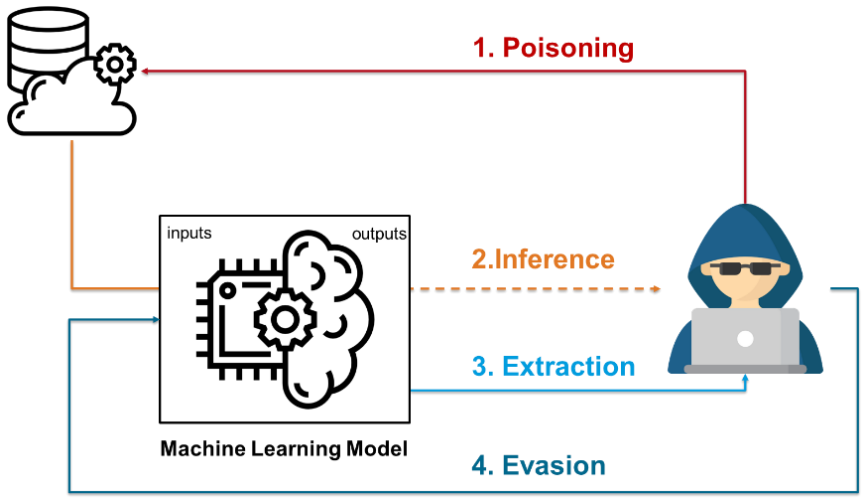
3.3 Threats to Artificial Intelligence and Machine Learning

In recent years, we have seen the integration of machine learning applications in multiple fields, even the most critical ones such as automotive and cybersecurity. Functions like autonomous driving and intrusion detection now rely on machine learning models to allow better adaptability to the environment and the detection of previously unknown threats [1]. However, while the development of these models mainly targeted performance, it is only recently that the security concerns raised by machine learning have been addressed. Indeed, recent works have shown the vulnerability of machine learning models to adversarial examples: data instances modified using a small and well-computed perturbation that compromises the prediction of the model [2]. Using adversarial methods, attackers could slightly modify the attributes of their attacks so that it is no longer classified as malicious by the intrusion detection model. This represents a serious threat to the security of critical systems relying on machine learning-based intrusion detection systems. Adversarial examples on neural networks have attracted a lot of attention in the research community. This has renewed interest in Adversarial Machine Learning, a research field that aims to evaluate and improve the robustness of machine learning models to malicious manipulations.

Interest in adversarial manipulations against machine learning models started in the early 2000s [3], but it is only after 10 years, when Szegedy et al. were able to fool neural networks with a small perturbation [4], that it became a factual preoccupation. At a moment when neural networks were gaining tremendous popularity for their performance, the phenomenon of adversarial examples represented a real threat to the numerous intended applications of these models. This threat motivated multiple research initiatives exploring different ways to generate adversarial examples and countermeasures to enhance the robustness of machine learning models. In this section, we present a brief overview of the most influential methods for the generation of adversarial examples (also designated as adversarial attacks).

3.3.1. Categories of Adversarial Machine Learning

Adversarial Machine Learning (AdvML) aims at evaluating and improving the robustness of machine learning models against malicious manipulations. The extensive literature in this field reports a wide variety of attacks that fall into four categories:



1. **Poisoning attacks** are achieved before the training phase by introducing perturbations among the training data to generate a corrupted model.
2. **Evasion attacks** happen after the model is trained. They are used to manipulate the input data of a model to trigger erroneous predictions.
3. **Extraction attacks** try to steal the parameters of a remote model in order to reproduce its behaviour or rob confidential information.
4. **Inference (Inversion) attacks** abuse a model to extort sensitive information learned from the training data.

3.3.2. Adversarial Examples Generation

Adversarial examples are data instances to which an imperceptible and well computed perturbation is applied. This perturbation aims to fool the machine learning model at test time into classifying the instance in the wrong class. Optimization methods are used to find a perturbation that: (i) minimizes the distance between the adversarial example and the original example; (ii) misclassifies the adversarial example; and (iii) complies with the data constraints. Given a data instance x to which the model assigns the label l, the corresponding adversarial examples x' is generated by adding a perturbation η to x. The generation can be described as the box constrained optimization problem in Equation (1):

$$\begin{aligned} &\min_{\eta} \|x' - x\| \\ &\text{s.t. } f(x') \neq l \\ &\quad x' \in I^m \end{aligned} \tag{1}$$

where I^m denotes the definition domain of the inputs, $\|\cdot\|$ denotes the distance norm, and the adversarial example $x' = x + \eta$.

According to Biggio and Roli [5], early work on adversarial attacks attempted to evade statistical spam detection by manipulating the content of the message [6]–[8]. But the research field attracted much more attention after the progress achieved in deep learning. When local generalization (in the neighbourhood of training examples) was assumed for neural networks, Szegedy et al. [9] discovered blind spots in the features space of image classifiers where the prediction of the model is arbitrary. These blind spots could be reached by applying an imperceptible non-random perturbation to training examples.
The same work revealed the property of transferability: adversarial examples generated on a model can mislead other models trained from scratch with different hyper-parameters, and even different examples of the training set. They also suggest that training the model on adversarial examples could improve its robustness, what would later be presented as adversarial training [10].

3.3.3. Taxonomy of adversarial examples

To analyse approaches for generating adversarial examples, Yuan et al. [11] categorise them along three dimensions, each of which is further decomposed into several aspects.

The first dimension is the threat model, it describes the different scenarios, assumptions and requirements that characterize an attacker. Its aspects include:

1. Adversarial falsification: whether the attacker aims to cause a false positive (e.g. facial recognition access control where the attacker wants to get an access) or a false negative (e.g. intrusion detection where the attacker aims to classify malicious traffic as benign).
2. Adversary's knowledge: the level of knowledge an adversary has about its target: (i) white-box attacks require a complete knowledge about the hyper-parameters, the parameters, and sometimes the loss function of the targeted model; while (ii) black-box attacks assumes the attacker can only query the model for an output (label or confidence score) without any prior knowledge about the structure or the parameters of the model.
3. Adversarial specificity: in a multi-class classification setting, (i) targeted attacks aim to misclassify an adversarial example with a specific label; while (ii) untargeted attacks aim to cause a misclassification regardless of the label. In a binary classification setting, the objective of targeted and untargeted attacks is equivalent.
4. Attack frequency: (i) one-time attacks only take a single optimization step, while (ii) iterative attacks take several iterations.

The second dimension is the perturbation, which should be imperceptible and compliant with the data constraints. Its aspects include:

1. Perturbation scope: (i) individual attacks generate a distinct perturbation for each example, and (ii) universal attacks generate a single perturbation for a set of examples.
2. Perturbation limitation: the perturbation can be (i) optimized if minimizing it is the goal of the optimization problem, or (ii) constrained if the optimization is constrained by a certain amount of perturbation.
3. Perturbation measurement: the most common metric to measure the perturbation is Lp distance norm described in Equation (2) for $p \geq 1$.

$$\|\eta\|_p = \left(\sum_{i=1}^{m} \|\eta\|^p \right)^{\frac{1}{p}} \qquad (2)$$

The most commonly used Lp norms are: the (i) L0 norm that computes the number of perturbed features; the (ii) L2 norm that computes the Euclidean distance; and the (iii) $L\infty$ norm that computes the highest perturbation applied to a feature.

Other metrics were proposed to measure the perturbation, for examples the Psychometric perceptual adversarial similarity score (PASS) [12], that measures the human perception of the perturbation.

The third dimension is the benchmark, it describes the framework on which the attacks were probed. This framework includes: (i) the datasets and (ii) the victim models on which the adversarial examples were generated.

Transferability In the initial work by Szegedy et al. [13] identified the property of cross model generalization and cross training-set generalization; a relatively large portion of adversarial examples are misclassified on models trained from scratch with different hyper-parameters and disjoint training sets. This observation supports the theory that adversarial examples are not just the results of overfitting to a certain model or training set, but they are rather universal. This property is more commonly known as the transferability of adversarial examples. It was explored in the literature [14], [15], especially for designing black-box adversarial attacks (addressed in details in subsequent sections).

3.3.4. White-box adversarial examples

In this section, we describe white-box methods for the generation of adversarial examples. In this setting, the attacker is assumed to have knowledge about the hyper-parameters of the model, including the architecture and the activation functions; the parameters of the model, which include the weights and the biases; and in some cases, the loss function used to train the model.

L-BFGS. When Szegedy et al. [5] first discovered adversarial examples in neural networks, they formulated the problem of finding adversarial examples with Equation (3), which is a targeted formulation of Equation (1).

$$\begin{aligned} \min \ & \|\eta\|_2 \\ \text{s.t.} \ & f(x+\eta) = l' \\ & x+\eta \in I^m \end{aligned} \qquad (3)$$

Where l′, the target class, is different from l (knowing that f (x) = l), and I^m is the domain definition of pixel values $[0, 1]^m$. In order to use the box constrained L-BFGS optimization method to solve the problem, they formulated it as the single minimization problem described in Equation (4).

$$\begin{aligned} \min_\eta \ & c\|\eta\|_2 + J_\theta(x+\eta, l') \\ \text{s.t.} \ & x+\eta \in I^m \end{aligned} \qquad (4)$$

They use line-search to find the minimum constant c > 0 (that weights the relative importance of the distance term and the loss function [16]) for which the perturbation η satisfies f (x + η) = l′.

Fast Gradient Sign Method (FGSM) Goodfellow et al. [17] suggest that adversarial examples are not due to the nonlinearity of deep neural networks, but rather a result of their linear behaviour in high-dimensional spaces. The same linearity that makes the models easier to train, exposes them to adversarial examples. According to their explanation, they propose a fast method to compute the adversarial perturbation: they linearize the loss function around the current value of θ, obtaining an optimal L∞-norm constrained perturbation, as shown in Equation (5).

$$\eta = \epsilon \ \text{sign}\left(\nabla x \ J\theta\ (x, l)\right) \qquad (5)$$

Where ϵ is the L∞-norm of the perturbation. This perturbation can be cheaply computed using backpropagation, which makes it practical for adversarial training. FGSM generates an untargeted perturbation: adversarial examples that are not classified in the correct class l. To formulate the problem in a targeted way, Kurakin et al. [18] proposed to optimize the adversarial examples with regard to a specific target class l′. In the paper, the authors use the least likely class as a target (the class with the smallest confidence score) and refer to this attack as the One-step least-likely class.

$$\eta = -\epsilon \ \text{sign}\left(\nabla_x J_\theta(x, l')\right) \qquad (6)$$

Rozsa et al. [19] proposed to use the raw value of the gradient instead of its sign, and referred to this method as the Fast Gradient Value Method (FGVM). The generated adversarial examples have no constraint on the amount of perturbation applied to each feature.

Basic Iterative Method (BIM) Instead of applying a single-step gradient update to generate the perturbation, Kurakin et al. [20] propose to apply multiple steps iteratively with a small magnitude α. After each step, the feature values of intermediate adversarial examples are clipped, in order to keep the perturbation in the L∞ ϵ-neighbourhood ($\|\eta\|_\infty < \epsilon$).

For an interval $I = [i_{min}, i_{max}]$ and a perturbation magnitude ϵ the clipping function is defined in Equation (8).

$$\text{Clip}_{x,\epsilon}\{x'\} = \min\{i_{max}, x + \epsilon, \max\{i_{min}, x - |\epsilon, x'\}\} \tag{8}$$

The iterative attack is then applied according to Equation (9):

$$\begin{aligned} x'_0 &= x, \\ x'_{n+1} &= \text{Clip}_{x,\epsilon}\{x'_n + \alpha \ \text{sign}\left(\nabla_x J_\theta\left(x'_n, l\right)\right)\} \end{aligned} \tag{9}$$

Kurakin et al. [20] later argued that untargeted attacks are sufficient for applications to datasets with a small number of highly distinct classes. However, with larger number of classes and varying degrees of significance in the difference between classes, untargeted attacks can result in uninteresting misclassifications. Thus, they introduce a targeted version of their iterative method that aims to misclassify the adversarial example in a specific class. As a target class, they choose the least-likely class according to the predication of the trained model.
This choice is motivated by the assumption that for well-trained classifier, the least likely method is usually highly dissimilar from the true class, which results in more interesting misclassifications. The attack, referred to as the iterative least-likely class method, is described in Equation (10).

$$\begin{aligned} l' &= \arg\min_y \{p\left(f\left(x\right) = y \mid x\right)\}, \\ x'_0 &= x, \\ x'_{n+1} &= \text{Clip}_{x,\epsilon}\{x'_n - \alpha \ \text{sign}\left(\nabla_x J_\theta\left(x'_n, l'\right)\right)\} \end{aligned} \tag{10}$$

The iterative attacks introduced in Kurakin et al. [20] were designed to target vision models operating in the real world and perceiving data through sensors. Their adversarial examples are more robust to transformation like printing or photography. Similar work has been proposed by Sharif et al. [21].

Jacobian-based Saliency Map Attack (JSMA) Papernot et al. propose JSMA, an attack that perturbs a (frequently small) fraction of the input features by focusing on the most influential ones, thus optimizing the L0 distance norm. While previous methods mainly exploit the gradient of the loss function, this method do not require knowledge about the training algorithm. Instead, JSMA uses the forward derivative (Jacobian of 'f', described in Equation (11)) to estimate the changes that a certain input has on the outputs of the model.

$$J_f(x) = \frac{\partial f(x)}{\partial x} = \left[\frac{\partial f_j(x)}{\partial x_i}\right]_{i \in 1..m,\ j \in 1..k} \tag{11}$$

The Jacobian is then used to construct a saliency map that indicates which features should be perturbed in order to influence efficiently the prediction of the model. The saliency map example described in Equation (12) values the input features whose increase either increases the probability of the target class, or decreases the probability of the other classes, or both.

$$S(x,l')[i] = \begin{cases} 0 & \text{if } J_{il'}(x) < 0 \text{ or } \sum_{j \neq l'} J_{il'}(x) > 0 \\ J_{il'}(x) \left| \sum_{j \neq l'} J_{il'}(x) \right| & \text{otherwise} \end{cases} \tag{12}$$

Equation (13) describes a similar example of the saliency map with regard to decreasing the value of the features.

$$S(x,l')[i] = \begin{cases} 0 & \text{if } J_{il'}(x) > 0 \text{ or } \sum_{j \neq l'} J_{il'}(x) < 0 \\ J_{il'}(x) \left| \sum_{j \neq l'} J_{il'}(x) \right| & \text{otherwise} \end{cases} \tag{13}$$

JSMA proceeds iteratively; at each iteration, the saliency map is computer and the feature with the highest saliency value for the targeted class is perturbed. The algorithm terminates when the targeted adversarial examples is effective (f (x′) = l′), or when the maximum distortion threshold Y is reached.

DeepFool was proposed by Moosavi-Dezfooli et al. [22] as a method to generate optimized adversarial examples. This attack tries to find the closes decision boundary from an original example, the minimal perturbation is then then the distance between the example and the closes boundary. This attack is untargeted, since it looks for the closest boundary regardless of the class it delimits. The authors first show how they can find the minimal perturbation on an affine classifier. However, neural network is non-linear, which makes it harder to find the closest classification boundary. The authors propose an iterative attack with a linear approximation to estimate this distance. **Erreur ! Source du renvoi introuvable.** graphically illustrates the intuition behind the linear approximation in DeepFool. They first define their method on a binary differentiable classifier, and generalize it to multiclass classifiers. Their method originally uses the L2 distance norm, but it can be extended to more general Lp norms with p ∈ [0,∞].

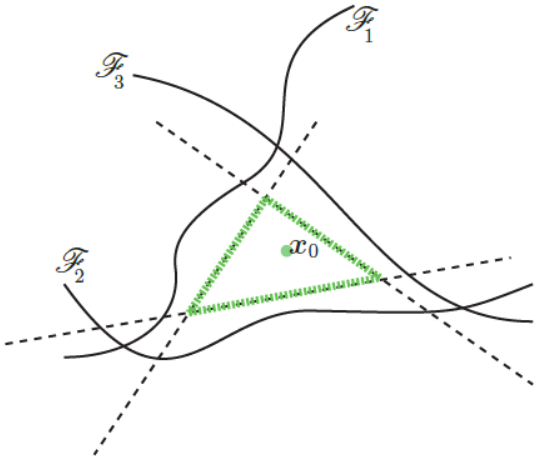


Figure 8 Illustration of the linearization in DeepFool (from Moosavi-Dezfooli [22])

The authors propose several candidates for the objective function. In their experiments, they use the objective function described in Equation (15):

$$g(x') = \max\left(\max_{i \neq l'} (z(x')_i) - z(x')_{l'}, -\kappa\right) \quad (15)$$

3.3.5. Black-box adversarial examples

In this section, we describe black-box methods for the generation of adversarial examples. This is a more realistic setting where the attacker is assumed to have limited knowledge about the model. Thus, it is impossible to exploit the gradients, the loss function or the parameters of the model. The attacker can either carry the model and access its output (Interactive black-box), or act without access to the model (Non-interactive black-box).

Interactive black-box

In this setting, the attackers can query the model with their own inputs and access the outputs. Two approaches are discussed here; the transfer attack and the zeroth-order optimization attack.

Transfer attacks. In order to attack models without any knowledge about their structure or parameters, Papernot et al. [35] train a substitute model fs on a synthetic dataset, which consists synthetic inputs generated by the adversary and labelled by the target model.

To generate synthetic inputs, the authors first used random inputs, like Gaussian noise, but the substitute models were unable to learn on such data. They argue that it is because the noise is not representative of the input distribution. Instead, they introduce a heuristic method to efficiently explore the input domain to find inputs that help the substitute model approximate the target decision boundaries, and limit the number of queries. The heuristic identifies the directions in which the model output is varying using the Jacobian-based Dataset Augmentation: at each iteration, it generates inputs by adding a value λ depending on the sign of the Jacobian matrix (Equation (11)) of the substitute model with regard to the label assigned by the target model. The term added to generate the new input is described in Equation (16).

$$x_{n+1}^{\text{synthetic}} = x_n^{\text{synthetic}} + \lambda \cdot \text{sign}\left(J_{f_s}\left[f\left(x_n^{\text{synthetic}}\right)\right]\right) \quad (16)$$

At each iteration, the target model is queried using the synthetic inputs, and the substitute model is trained on the labels it produces. Then, the Jacobian based dataset augmentation is used to generate new synthetic data based on the previous ones.

The training of a substitute model is the main part of transfer attacks. Once it is done, any white-box method could be used to generate adversarial examples. However, the effectiveness of transfer attacks is mainly dependent on the transferability of the attack from the substitute model to the target model [23].

Papernot et al. [14] generalize the phenomenon of transferability across the space of machine learning models. In addition to the intra-technique transferability, demonstrated across model trained with the same algorithm (mainly neural networks); the authors explore the cross-technique transferability, that considers models trained using different techniques. Throughout their

experiments, they show how black-box transfer attacks are possible against any unknown machine learning model. They also improve the substitute model training procedure introduced in Papernot et al. [14] with a periodical step size λ and reservoir sampling [24] to reduce the number of queries.

Zeroth-Order Optimization (ZOO) In order to avoid any potential loss in transferability from the substitute model to the target model, Chen et al. [23] propose to directly estimate the gradient of the target model using only queries. They use the problem formulation of Carlini and Wagner [25] and adapt it to the black-box setting: (i) the logits z (·) in objective function g (·) are replaced with log of the outputs of the model, as described in Equation (17);

$$g\left(x'\right) = \max\left(\max_{i \neq l'}\left(\log\left(f\left(x'\right)_i\right)\right) - \log\left(f\left(x'\right)_{l'}\right), -\kappa\right) \quad (17)$$

(ii) the gradients are approximated using a finite difference method, described in Equation (18) and the optimization problem is solved with ZOO.

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}_i} \approx \frac{f\left(\mathbf{x} + h\mathbf{e}_i\right) - f\left(\mathbf{x} - h\mathbf{e}_i\right)}{2h} \quad (18)$$

where ei is a standard basis vector with only the i-th component as 1.
The estimation of the gradient can be very expensive in high-dimensional feature spaces. To tackle this problem, these authors propose a stochastic coordinate descent where, at each iteration, one feature is chosen randomly chosen an updated. After estimating the gradient, first-order optimization methods like Adam [26] can be used to compute the update. Another idea consists of reducing the attack-space dimension using a transformation D (·), and gradually increase the dimension using a series of transformations.
GAN attacks Zhao et al. [27] introduce a framework to generate meaningfully similar adversarial examples in images and text, **Erreur ! Source du renvoi introuvable.** describes their framework. Instead of searching for adversarial examples in the input data space directly, they learn a projection from a dense and continuous space of representations to the data space using Generative Adversarial Network (GANs) [28].

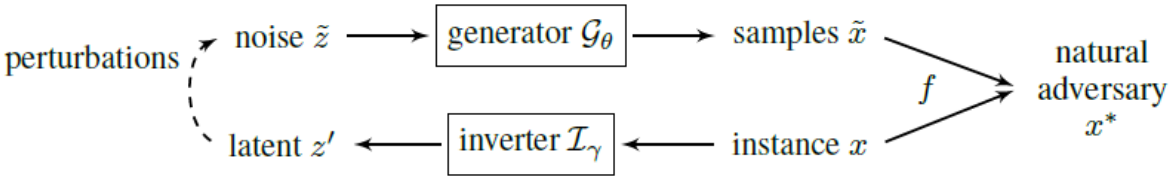


Figure 9 The generation framework of natural adversarial examples

A matching inverter Iγ is then trained to map original examples to corresponding representations in the latent space, adversarial examples are then found in the neighbourhood of the representation and generated with Gθ. The training objective of Iγ is shown in Equation (19), the loss of the inverter balandes the reconstruction error of x and the divergence between Gaussian distribution z Iγ (Gθ(z)) with a parameter λ.

$$\min_{\gamma} \mathbb{E}_{x \sim p_x(x)} \left\| \mathcal{G}_\theta \left(\mathcal{I}_\gamma(x) \right) - x \right\| + \lambda \cdot \mathbb{E}_{z \sim p_z(z)} \left[\mathcal{L} \left(z, \mathcal{I}_\gamma \left(\mathcal{G}_\theta(z) \right) \right) \right] \tag{19}$$

Non-interactive Black-box (No-Box)
This is a more constrained setting where the attacker has no direct access to the model and cannot execute queries.

Adversarial Examples Games (AEG). Bose et al. [3] viewed generation of adversarial examples as a form of adversarial game. The players are the generator g and the representative classifier fc, and both are jointly optimized in the maximin zero-sum game in Equation (20). The generator g learns to map a prior distribution pz into a distribution of adversarial examples, it is conditioned by original examples (x, y) that are drawn from a dataset D. The representative classifier fc learns to classify robustly the examples generated by g. The dynamics of the game progressively lead to a stronger generator and classifier. The Nash equilibrium of this game leads to a distribution of adversarial examples effective against any classifier from the hypothesis class F.

$$\max_{g \in \mathcal{G}_\epsilon} \min_{f_c \in \mathcal{F}} \mathbb{E}_{(x,y) \sim \mathcal{D}, z \sim p_z} \left[J_\theta \left(f_c(g(x, y, z)), y \right) \right] =: \varphi \left(f_c, g \right) \tag{20}$$

The attacker is assumed to have the ability to train a representative classifier fc from the same hypothesis class F as the target classifier ft, and with a reference dataset Dref sampled from the same distribution as the training dataset of ft. The size of the hypothesis class F can be reduced by assuming that ft performs well at the classification task. The authors enforced this assumption by adding a regularization term that penalizes bad performing representative classifiers, as described in Equation (21)

$$\max_{g \in \mathcal{G}_\epsilon} \min_{f_c \in \mathcal{F}} \mathbb{E}_{(x,y) \sim \mathcal{D}, z \sim p_z} \left[J_\theta \left(f_c(g(x, y, z)), y \right) \right] + \lambda \mathbb{E}_{(x,y) \sim \mathcal{D}_{\text{ref}}} \left[J_\theta \left(f_c(x), y \right) \right] =: \varphi \left(f_c, g \right) \tag{21}$$

Manipulating the game by introducing biases in the training of fc is a way of implicitly incorporating prior knowledge on the hypothesis class F of the target model ft.

3.3.6. Defences against adversarial examples

As soon adversarial examples were first discovered on neural networks [5], the research community started designing different attack approaches to generate adversarial examples. In parallel, the community started designing defence mechanisms to mitigate these attacks. In this section, we review the principal defence approaches against adversarial examples. According to Papernot et al. [15], we categorize them into two types: reactive defences and proactive defences. Other approaches are reviewed and summarized in Yuan et al. [11].

3.3.7. Reactive defences

Reactive defences are methods that mitigate the risk of adversarial examples on models that were already trained. Here, we review two main reactive defences: inputs reconstruction and adversarial examples detection.

3.3.8. Input reconstruction

Since adversarial examples are original data that are perturbed, a defence against them could be to remove the adversarial perturbation of the input before classifying it. This was proposed

by Gu and Rigazio [29] who train a denoising autoencoder to encode adversarial examples into original examples. Meng and Chen [30] add Gaussian noise before encoding the example with the autoencoder. Song et al. [48] introduce PixelDefend, a method to remove the adversarial perturbation by moving the example back to the training data distribution Pt.

3.3.9. Adversarial examples detection

In order to protect a model from adversarial examples, one solution would be to prevent them from reaching the model. To do so, researchers have proposed methods to detect adversarial examples before they are fed to the models. This can be done using a binary classifier that learns to distinguish adversarial inputs from original ones. Grosse et al. [31] propose to leverage the structure and the classification abilities of the original neural network to detect adversarial examples. They added and outlier class to the model and train it to classify adversarial examples in that class. They were able to distinguish the distribution of original examples Xorg and the distribution of adversarial examples Xadv using the Maximum Mean Discrepancy (MMD),

3.3.10. Proactive defences

Proactive defences are methods that aim to construct a robust model. Here, we review three proactive approaches: gradient masking, adversarial training, and network distillation.

3.3.11. Gradient masking

Since many white-box attacks use the gradients of the network to generate adversarial examples, one countermeasure would be hiding the gradients from the attacker. One way of doing so is to reduce the sensitivity of models to small changes [32] or to construct models that are not differentiable like learning trees. However, since the attacker can construct a surrogate model that is a smoother version of the target model, these models are often still vulnerable to adversarial examples through transferability [15]. Another way of masking the gradient would be to create a sharp curve in the loss function near the data points. By doing so, the gradient at that point do not represent the global curve of the loss function, thus giving the attacker a wrong direction. However, Tramèr et al. [33] were able to overcome this defence by taking a small random step before computing the gradient. The random step allows the attacker to escape the non-smooth vicinity of the data point.

3.3.12. Adversarial training

Since the very first work addressing adversarial examples on neural networks, Sezgedy et al. [5] suggested that back-feeding adversarial examples to train the model could improve its generalization. Their experiments showed that augmenting the training set with adversarial examples decreased their impact. This hypothesis was investigated by Goodfellow et al. [10] who proposed to incorporate adversarial examples in the training objective of the model. At every training step, they optimize the model on original training data and their adversarial examples. The authors used FGSM to generate adversarial examples, they added the generation term as a regularizer in the objective function, as described in Equation (22). The constant λ balances the importance of the original training objective and the adversarial objective.

$$\tilde{J}_\theta(x,l) = \lambda J_\theta(x,l) + (1-\lambda) J_\theta\left(x + \epsilon \operatorname{sign}\left(\nabla_{\boldsymbol{x}} J_\theta(x,l)\right)\right) \quad (22)$$

3.3.13. Defensive distillation

Hinton et al. [34] originally introduced network distillation as a technique to reduce the size of a large neural network or an ensemble of neural network by transferring their knowledge into a smaller one. Based on the assumption that the knowledge in neural networks is not only encoded in the parameters, but also in the probability vector. Thus, they transfer the knowledge by training a smaller neural network on the probability vectors produced by the original neural network (soft labels) instead of the data labels (hard labels).
The original network is trained with the softmax output layer s (·) that takes the logits z (·) and transforms them into a probability distribution, as described in Equation (23). The temperature parameter T controls the level of knowledge distillation; when larger than 1, it produces relatively nuanced probabilities; when smaller than 1, it amplifies the logits values and produces more discrete probabilities.

$$s(x) = \left[\frac{e^{z_i(x)/T}}{\sum_{j=0}^{n-1} e^{z_j(x)/T}} \right]_{i \in 0..n-1} \quad (23)$$

Papernot et al. [35] argue that adversarial attacks primarily exploit gradients of the model to compute the perturbation. This is made easier on a sensitive model with high gradients, since a small perturbation can induce high output variations. The authors show that network distillation can help to improve the model generalization capabilities and its robustness to adversarial perturbation. They introduce defensive distillation; a technique based on network distillation to train robust classifiers. An overview of the defensive distillation framework is shown in **Erreur ! Source du renvoi introuvable.** below.

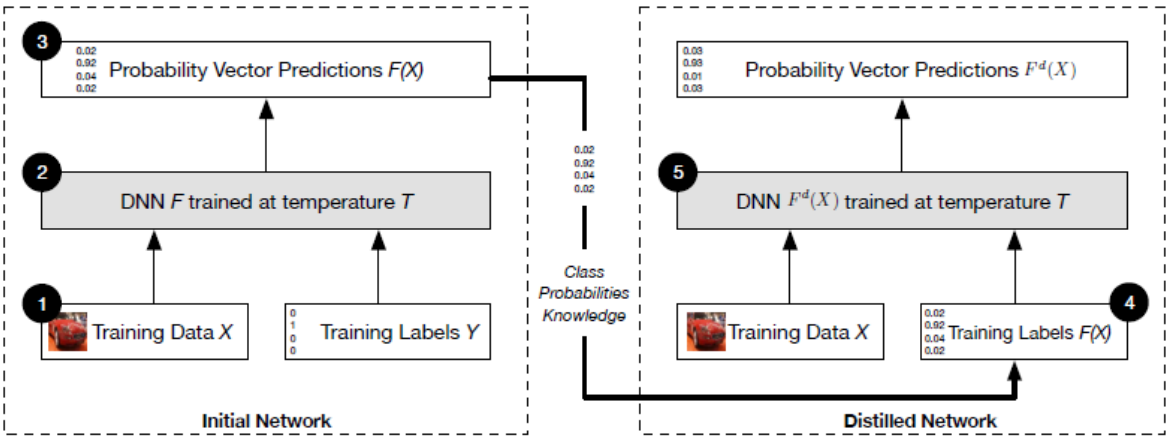


Figure 10 Defensive distillation framework (Perpenot et al. [35])

Since their objective is not to train a smaller model, they keep the same architecture between the original and distilled neural networks. The authors use a high temperature T to improve the smoothness of the distilled model and reduce its sensitivity to small perturbations. They showed that adversarial attacks struggled to find adversarial perturbations on distilled networks as their gradients were reduced by a factor of 10^{30}.

3.4 Documents with global risk identification for related systems

3.4.1. UNCE - R155 (Annex 5)

United Nations Economic Commission for Europe (UNECE) has defined the regulation R155 for cyber security introducing a Cybersecurity Management System (CSMS) [40] in automotive at organization level. This recent regulation addresses the risk associate with increasing connectivity and digitized vehicle environment. Its scope covers passenger vehicles, busses, light and heavy-duty trucks, quadricycles, and trailers and if not directly tackling AI based transports or ADS still it covers most of the challenges linked to cyber security of automotive systems. In the context of our study the most interesting part of this regulation for us is its Annex 5 which identifies several treats and their corresponding mitigations. The rest of the document which might be of interest latter on for us in the project covers certification and marking processes in the context of vehicle type homologation.

This annex is composed of three parts:

- Part A which describes the baseline for threats, vulnerabilities, and attack methods
- Part B which describes mitigations to the threats which are intended for vehicle types
- Part C which describes mitigations to the threats which are intended for areas outside of vehicles, e.g., on IT backends.

Again, this annexe is not dedicated to ADS, however it provides an important state of the art and overview of threats applicable to any kind of C-ITS and thus ADS which can be helpful for us. The main content of this table is presented in ANNEX – UNECE R 155 THREATS IDENTIFICATION TABLE.

3.4.2. ETSI ITS TVRA

The European Telecommunications Standards Institute (ETSI) is a standardization organization in the field of information and communications founded in 1988. Among many other topics, the ETSI Technical Committee (TC) ITS WG5 aims at providing security standards for ITS platforms and applications. One of their technical reports is of interest for us in our current study: the ITS TVRA [41]. This document presents a threat analysis for two main components of ITS systems, the RSU and the OBU.
This document starts by reminding the ITS architecture defined by the different ETSI standards, it presents the different components and their functionalities. Then, following the TVRA approach, it defines the security objectives for the system, the ITS functional security classes, the two target of evaluation (RSU and OBU) for which they describe in more details their functionalities, their interfaces, and their manipulated data in order to define the corresponding assets to be protected. Finally, it presents the threat analysis for these components together with the required countermeasures to protect the RSU and the OBU from those threats.
ITS OBUs and RSUs are important element in the scope of our study. They support most of the communication within global ITS system (cf. section 2.1). Thus, the threat faced by those elements are clearly in the scope of the project.
Even if the TVRA method can be criticized on such point as: the meaningfulness of. defining objectives before identifying risks, or identifying weaknesses of generic architecture and not real technical devices, the presented weaknesses; the threat analysis and vulnerability identification is quite extensive and is a good summary of the state of the art. This is a valuable input for our study.

For each threat to these devices, they identify the ITS-S problem area (the architectural or functional source of the weakness), the weakness it exploits, the threat agent able to exploit it, and the attacked interface, e.g.:

- Injection of false messages
    - ITS-S Problem Area
        - Absence of addressing in broadcast messages meaning source cannot be identified so malicious and irrelevant messages can only be rejected on the application layer, not at the network layer in the ITS stack
        - Uncertainty regarding how timestamps are created and how to use them to heck the validity of messages
    - Weakness
        - An RSU is unable to quickly determine whether a received message contains accurate information and is from a legitimate user and acts by relaying the message. An RSU can only check whether the message is valid and comes from a valid source. The time taken by an RSU to process a high volume of real or spurious messages could cause it to miss important incoming ITS messages. An RSU is unable to validate when a received message was originally generated.

The identified list of threat for the OBU

- Message saturation, Jamming of radio signals, Injection of false messages, Manipulation of ITS messages en route, Masquerade as ITS S (Vehicle or Roadside) or ITS network, Masquerade for fabrication of messages, Replay of "expired" (old) messages, Wormhole attacks, GNSS spoofing, Malicious isolation of one or more ITS S (Vehicle) (black hole), Eavesdropping, Traffic analysis, Location tracking

And for the RSU:

- Injection of a high volume of false emergency vehicle warning messages, Message saturation, Radio jamming, Injection of false messages, Replay of "expired" (old) messages, Wormhole attack, GNSS spoofing, Emergency vehicle masquerade, Eavesdropping, Traffic analysis, Location tracking, Transaction tampering, Denial of transmission

Most of these threats have already been presented in sections 3.1.1 and 3.1.2 but it provides a good taxonomy and some more specific descriptions details.
Table 8 “List of Vulnerabilities for ITS-S (Vehicle) ToE” and 14 “List of Vulnerabilities for ITS-S (Roadside)” of [41] are to be fully integrated in our state of the art and are copied in annex.
[42] is a public technical report owned and copyrighted by the ETSI, but for the sake of completeness of this deliverable we provide a copy of these tables extracted from this document. But the reader should refer to [42] for their full content and description.

4 THREATS

Based on the state of the art presented in the previous section and the reference architecture we have defined in section 2, we first start to identify threat agents, that are of interest in our study; A threat agent is an attacker profile defined based on both the system interfaces they has access to (e.g. vehicle radio interface, vehicle sensor attacking them for the roadside, physical access to the IVN, all interfaces or communications accessible form internet, etc.)

and their access rights (none or some role -legitimate or not- attributed by the system). This will help us to better identify the threats which are to be considered for each component of the system.
Based on this identification, we identify for each system asset (data or functional) all the know attacks taken from the previous state of the art accessible to each attacker profile and define if either we consider them to be:

1. Covered by technical means (specific implementations) to be tested within PRISSMA assurance framework
2. Covered by procedural and technical means to be audited within PRISSMA assurance framework
3. Out of scope of PRISSMA assurance framework.

4.1 Threat agents

Figure 11 present the different attacker positions we can identify form the state of the art. Each of these positions requires different equipment and knowledge to communication or interfere with specific systems interfaces and provides access to different attack type with potentially different consequences. That's why we separate them in the chosen following way.

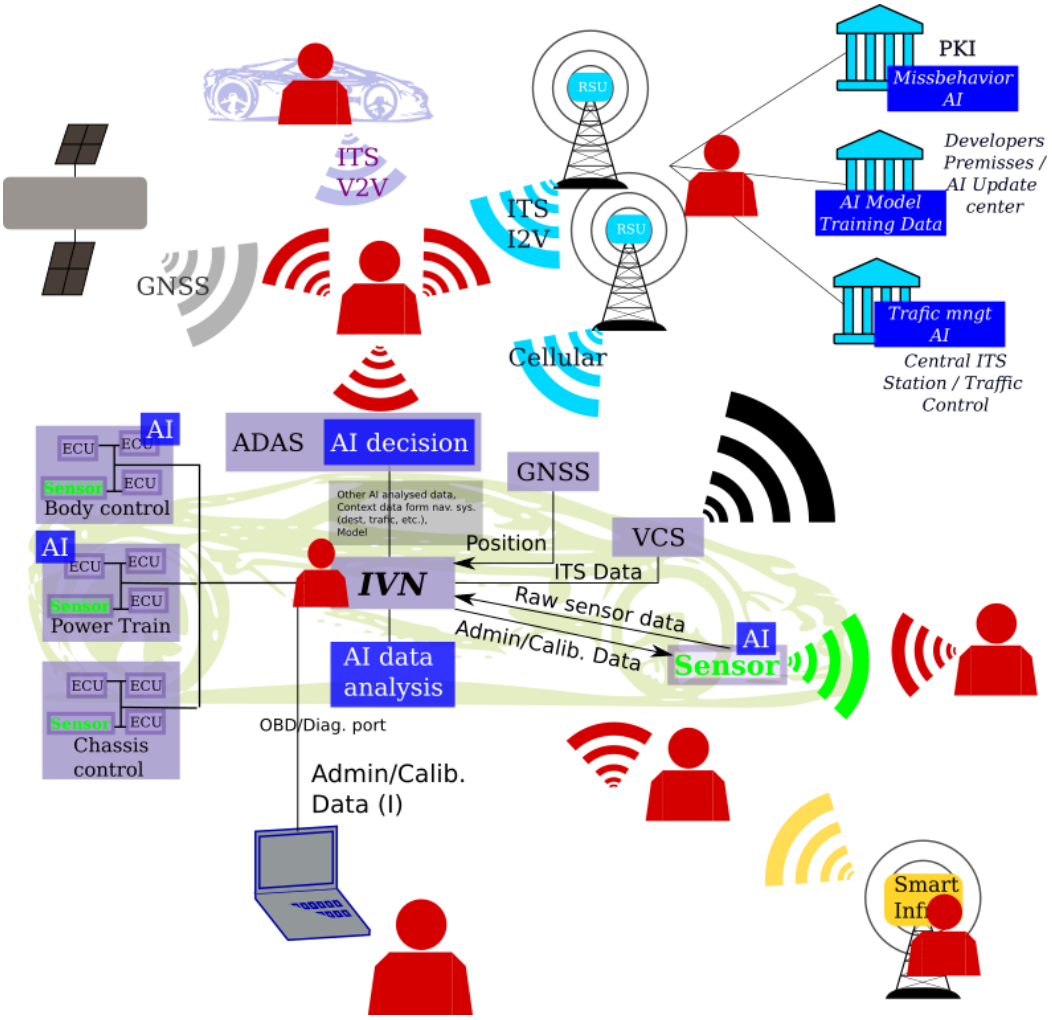


Figure 11 Threat agents

In Table 7 we provide a textual description of the threat agent positioned in the system in Figure 11.

Name	Description
Remote attackers	
Radio	An attacker able to emit or receive GNSS, G5, cellular radio signals to intercept, jam, replay, fake messages from or to the vehicle or the infrastructure.
Rogue ITS-S (vehicle or roadside unit)	An attacker using a rogue equipment sends rogue ITS messages to the autonomous system.
Internet	Remote attacker sending or intercepting messages between the infrastructure (PKI, central ITS, developer premises) and the vehicles or trying to get unauthorized access to the vehicle or the infrastructure components (data or functionalities).
Local attackers	
Rogue users	Users having a physical and user granted access to a C-ITS station (a VCS, an RSU or central station), provides, intercept or modify rogue information sent to or by other system components via the HMI or networks interface of the component.
Rogue administrator	Same as rogue users but with administrative privileges.
IVN	An attacker accessing the internal AV network.
Roadside	An attacker trying to modify AV surrounding to force AV wrong or potentially dangerous decisions by impacting/modifying AV sensor observations (light perturbation, objects modification or introduction e.g. painting signs, using sensor blinders, etc.).

Table 7 Threat agent description

4.2 PRISSMA threat scope analysis

As mentioned previously, in this section we identify threats for each threat agent and system asset association, and we choose which of the following three type we consider them:

1. (Blue) Threat to be covered by technical means (specific implementations) to be tested within PRISSMA assurance framework (PRISSMA innovative/dedicated assurance)
2. (Light grey) Covered by procedural and technical means to be audited within PRISSMA assurance framework (common/classical security assurance)
3. (Dark grey) Out of PRISSMA's assurance framework scope.

This categorization is rather empirical, but it's based on the following elements we have discussed and analysed in this study:

- **Impact of the threat:** the higher the impact, the higher the assurance should be and thus dedicated technical test should be recommended.
- **Threat feasibility:** the easier the threat is to be executed, the more important it is to assess that the system is protected against it.
- **Technical difficulty of the assessment**: assurance is expensive and cannot be guaranteed the same way for a whole supervision centre composed of tenth or hundreds of IT components and one communication unit of a few mega octet. In the first case only security validation through best practices recommendations and audit scale up, while in

the second more advanced and precise evaluation schemes can apply (e.g., conformity tests, vulnerability tests, full assurance certification like CC [42] certifications, etc.)

So, our approach has been to perform something similar to a risk analysis (impact and threat feasibility assessment) balanced with an assurance assessment difficulty. The elements to be added to the firs category (blue cells) are the one with the higher impact and which require the definition of an efficient and dedicated assurance framework. While the second are one that have high or critical impacts but for which classical assurance framework are adapted (light grey cells). Finally, the last category is for element with low risks or with security assurance challenges or costs not worth the associated risks.
We do not provide formulas to justify our empirical choices. We provide the outcome of an educated expert estimation, which is the result of the discussions and analysis of PRISSMA's partners.
Also, *since* the table cannot exhaustively contain all attacks mentioned in this document for each pair of assets and threat agent, we only provide references to section identifying attacks applicable for the specific pair even not all attacks in the section are applicable. Identifying those sections related sections is sufficient for us to allow our classification.

Asset	Threat agent						
	Radio	Rogue ITS-S	Internet	Rogue Users	Rogue Adm.	IVN	Roadside
Data							
Keys							
Canonical Public Key	-	-	Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
Data encryption key	-	-	Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
CA private keys	-	-	Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
Certificates							
CA Certificates	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
Enrolment Credential (EC)	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-

			access, arbitrary code execution, etc.				
Authorization Ticket (AT)	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
TLM certificate	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
Station registration data							
Canonical ID	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and cf. section 3.1.1	-
ITS-S Profile	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and	-

			code execution, etc.			cf. section 3.1.1	
Tag	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and cf. section 3.1.1	-
HMAC key	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and cf. section 3.1.1	-
CA Network addresses	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and cf. section 3.1.1	-
DC network address	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and cf. section 3.1.1	-
CPOC Network address	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle,	Privilege escalation,	Privilege escalation,	Gain of unauthorized	-

			Gain of unauthorized access, arbitrary code execution, etc.	arbitrary code execution, etc.	arbitrary code execution, etc.	access, arbitrary code execution, etc. and cf. section 3.1.1	
Policies							
Certificate Policy configuration data	-	-	Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-
Trust lists							
CRL	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and cf. section 3.1.1	-
CTL	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and cf. section 3.1.1	-
ECTL	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, Gain of unauthorized access, arbitrary	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Gain of unauthorized access, arbitrary code execution, etc. and	-

			code execution, etc.			cf. section 3.1.1	
PKI services							
Software/Execution of the software	Cf. section 3.1.2 and 3.4.2	-	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-
Misbehaviour detection							
Misbehaviour Report (MR)	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2
ITS data							
X2V Safety sensitive ITS application data	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2
X2V Sensitive ITS application data	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2
X2V Informative ITS application data	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2
X2I Safety Sensitive ITS	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, etc.	Privilege escalation,	Privilege escalation,	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2

application data				arbitrary code execution, etc.	arbitrary code execution, etc.		
X2I Sensitive but not safety critical ITS application data	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2
X2I Informative ITS application data	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2
LDM	Cf. section 3.1.2 and 3.4.2	Cf. section 3.1.2 and 3.4.2	Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2
Sensor Data	-	-	Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.1.2 and 3.4.2
ITS software	-	-	Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
GNSS							
Time and position	GNSS spoofing, radio jamming	-	-	-	-	Cf. section 3.1.1	GNSS spoofing, radio jamming
Travelers apps and ticketing system							
User traveling and ticketing information	Replay, Man in the middle, etc.	-	Gain of unauthorized access, Arbitrary	Privilege escalation,	Privilege escalation,	Cf. section 3.1.1	-

			code execution, etc.	arbitrary code execution, etc.	arbitrary code execution, etc.		
IT management							
Configuration and calibration data	Replay, Arbitrary code execution, Man in the middle, etc.	-	Gain of unauthorized access, Arbitrary code execution, Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
Functions							
PKI							
Certificate request management	Cf. section 3.1.2	Cf. section 3.1.2	Gain of unauthorized access, Arbitrary code execution, Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-
Trust list management	Cf. section 3.1.2	Cf. section 3.1.2	Gain of unauthorized access, Arbitrary code execution, Replay, Man in the middle, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-
Misbehaviour management	Cf. section 3.1.2	Cf. section 3.1.2	Gain of unauthorized access, Arbitrary code execution, Replay,	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-

			Man in the middle, etc.				
Developer servers							
AI software or model update	-	-	Cf section 3.3.4.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-
Field data collection	Cf. section 3.1.2	Cf. section 3.1.2	Gain of unauthorized access, Arbitrary code execution, Replay, Man in the middle, etc. Cf. section 3.3.2 and 3.3.5.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-
Central ITS							
Traffic management	Cf. section 3.1.2	Cf. section 3.1.2	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-
Vehicle remote control	-	-	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	-	-
GNSS							
GNSS	Jamming, spoofing.	-	-	-	-	-	Jamming, spoofing.

Roadside infrastructure							
V2X support	Cf. section 3.4.1 and 3.4.2	Cf. section 3.4.1 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, etc.	-	-	-	Cf. section 3.2
Road infrastructure monitoring and environment perception	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.1.2, 3.4.1 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, etc.	-	-	-	Cf. section 3.2
Vehicle							
Journey	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.1.2, 3.4.1 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.2
LDM	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.1.2, 3.4.1 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.2
V2X communication	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.1.2, 3.4.1 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
ADS	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.3.5.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.2

Environment perception	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.1.2, 3.4.1 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	Cf. section 3.2
Audit and diagnostic	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.1.2, 3.4.1 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
Remote control and management	Cf. section 3.1.2, 3.4.1 and 3.4.2	Cf. section 3.1.2, 3.4.1 and 3.4.2	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-
Travelers apps and ticketing system							
Ticketing and payment validation	Replay, Arbitrary code execution, Man in the middle, etc.	-	Gain of unauthorized access, Arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Privilege escalation, arbitrary code execution, etc.	Cf. section 3.1.1	-

References

[1] ENISA, “Good Practices For Security Of Smart Cars,” 2019.

[2] ENISA, “Cybersecurity challenges in the uptake if artificial intelligence in autonomous driving,” doi:10.2760/551271, 2021.

[3] ETSI, “TS 102 940 - Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management,” 2016.

[4] ETSI, “302 665 - Intelligent Transport Systems (ITS); Communications Architecture”.

[5] A. R. J. R. S. E. G. B. F. D. M. N. M. M. Simon Ulbrich, “Towards a Functional System Architecture for Automated Vehicles,” *CoRR,* 2017.

[6] ETSI, “TS 103 097 Intelligent Transport Systems (ITS); Security; Security header and certificate formats”.

[7] ETSI, “TS 102 941 Intelligent Transport Systems (ITS); Security; Trust and Privacy Management”.

[8] ETSI, *TR 102 460 - Intelligent Transport Systems (ITS); Security; Pre-standardisation study on Misbehavior Detection; Release 2.*

[9] ETSI, *TS 102 731 Intelligent Transport Systems (ITS); Security; Security,* 1.1.1, 2010.

[10] ETSI, “302 637-2: Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service”.

[11] ETSI, “302 637-3: Specifications of Decentralized Environmental Notification Basic Service”.

[12] Z. K. S. D. F. S. S. J. P. P. R. El-Rewini, “Cybersecurity Challenges in Vehicular Communications,” *Vehicular Communications 23,* June 2020.

[13] H. T. N.W. Lo, “Illusion attack on VANET applications - a message plausibility problem,” *IEEE Global Telecommunications Conference,* 2007.

[14] A. C. V.H. La, “Security attacks and solutions in vehicular ad hoc networks: a survey,” *Int. J. AdHoc Netw. Syst. 4,* 2014.

[15] J. Douceur, “The sybil attack,” *International Workshop on Peer-to-Peer Systems,* p. 251–260, 2002.

[16] J.-L. M. H. H. I.A. Sumra, “Timing attack in vehicular network,” *Proceedings of the 15th World Scientific and Engineering Academy and Society (WSEAS) International Conference on Computers,* p. 151–155, 2011.

[17] S. R. J. Kasra Amirtahmasebi, “Vehicular Networks-Security, Vulnerabilities and Countermeasures,” *Ph.D. thesis, University of Gothenburg, Goteborg, Sweden,* 2010.

[18] A. P. B. Parno, “Challenges in securing vehicular networks,” *Proceedings of the Workshop on Hot Topics in Networks (HotNets-IV),* 2005.

[19] W. G. Q. Z. G. Y. Y. Liu, “Lvap: lightweight v2i authentication protocol using group communication in vanets,” *Int. J. Commun. Syst. 30,* 2017.

[20] M. C. H. L. H. H. M. Islam, “Cybersecurity attacks in vehicle-toinfrastructure (V2I) applications and their prevention,” *Computing Research Repository ,* 2017.

[21] H. C. J. C. J.Y. Kim, “An efficient authentication scheme for security and privacy preservation in V2I communications,” *IEEE 72nd Vehicular Technology Conference,* 2010.

[22] C. Smith, “The Car Hacker's Handbook - A Guide for Penetration Tester,” No Starch Press, 2016.

[23] S. Z. W. S. a. Y. S. Jiajia Liu, “In-Vehicle Network Attacks and Countermeasures: Challenges and Future Directions,” *IEEE Network,* pp. 50-58, September/October 2017.

[24] P. Noureldeen, M. A. Azer, A. Refaat and M. Alam, “Replay attack on lightweight CAN authentication protocol,” *12th International Conference on Computer Engineering and Systems (ICCES),* 2017.

[25] T. R. A. M. Y. J. T. M. Paul Carsten, “In-Vehicle Networks: Attacks, Vulnerabilities, and Proposed Solutions,” *Proceedings of the 10th Annual Cyber and Information Security Research Conference,* 2015.

[26] Y. H. N. K. N. T. X. C. a. S. W. Tian Guan, “An Overview of Vehicular Cybersecurity for Intelligent,” *Sustainability 2022,* 2022.

[27] J. W. S. J. L. a. K. G. S. Mert D. Pesé, “S2-CAN: Sufficiently Secure Controller Area Network,” *The University of Michigan, ACSAC ’21, December 6–10, 2021, Virtual Event, USA,* 2021.

[28] A. F. M. K. Gerard Chalhoub, “C-Roads - Overview about heterogeneous vehicular communications,” 2020.

[29] V. Marojevic, “C-V2X Security Requirements and Procedures: Survey and Research Directions,” 2018.

[30] R. P. Jover, “LTE security, protocol exploits and location tracking,” 2016.

[31] H. L. R. K. Jean Cassou-Mounat, “Simulation of Cyberattacks in ITS-G5 Systems,” *15th International Workshop Nets4Cars/Nets4Trains/Nets4Aircraft 2020,* vol. Springer LNCS 12574, pp. 3-13.

[32] A. B. A. R. Huong Nguyen-Minh, “Jamming Detection on 802.11p under Multi-channel,” *IEEE 11th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob),* pp. 764-770, 2015.

[33] A. V. M. J. a. J. L. Nikita Lyamin, “Real-Time Detection of Denial-of-Service Attacks in,” *IEEE Communications Letters,* vol. 18, pp. 110-113, 2014.

[34] R. KHATOUN, “State of the art on Cyberattacks in C-ITS, Attacks, taxonomy and Countermeasures,” *InDiD 2.7.3.2.1,* 2022.

[35] B. S. M. F. F. K. Jonathan Petit, “LiDAR, Remote Attacks on Auto-mated Vehicles Sen-sors:Experiments on Camera and LiDAR,” *Security Innovation, University of Twente, University of Ulm,* 2015.

[36] C. X. B. C. Y. Z. W. P. S. R. Q. A. C. K. F. a. Z. M. Yulong Cao, “Adversarial Sensor At-tack on LiDAR-based Perception in Autonomous Driving,” *University of Michigan & University of California,* 2019.

[37] D. N. R. B.-N. Y. M. O. D. Y. E. Ben Nassi, “Phantom Attacks on Driver-Assistance Systems,” *Ben-Gurion University of the Negev, Georgia Tech, Independent Tesla Researcher,* 2020.

[38] S. Povolny, “Model Hacking ADAS to Pave Safer Roads for Autonomous Vehicles,” *McAfee,* 2020.

[39] W. X. J. L. Chen Yan, “Can you trust autonomous vehicles: contact-less attacks against sen-sors of self-driving vehicle,” *Zhejiang University, Qihoo 360,* 2016.

[40] U. N. E. C. f. E. (UNECE), “R155 for cyber security introducing a Cybersecurity Management System (CSMS),” *https://unece.org/transport/documents/2021/03/standards/un-regulation-no-155-cyber-security-and-cyber-security.*

[41] ETSI, “TR 102 893 V1.2.1 - Intelligent Transport Systems (ITS); Security; Threat, Vulnerability and Risk Analysis (TVRA),” 2017.

[42] c. a. p. p. ISO/IEC JTC 1/SC 27 Information security, “ISO/IEC 15408 Information technology — Security techniques — Evaluation criteria for IT,” 2009.

[43] K. S. S. J. P. R. Zeinab El-Rewinia, Cybersecurity challenges in vehicular communications, Elsevier, 2020.

[44] D. K. Y. K. a. Y. K. Hocheol Shin, “Illusion and Dazzle: Adversarial Optical Channel Exploits against Lidars for Automotive Applications,” *Korea Advanced Institute of Science and Technology,* 2017.

ANNEX – UNECE R 155 THREATS IDENTIFICATION TABLE

From **Part A. Vulnerability or attack method related to the threats**

4.3.2 Threats to vehicles regarding their communication channels	4	Spoofing of messages or data received by the vehicle	4.1 **Spoofing of messages** by impersonation (e.g. 802.11p V2X during platooning, GNSS messages, etc.)
			4.2 **Sybil attack** (in order to spoof other vehicles as if there are many vehicles on the road)
	5	Communication channels used to conduct unauthorized manipulation, deletion or other amendments to vehicle held code/data	5.1 Communications channels permit **code injection**, for example tampered software binary might be injected into the communication stream
			5.2 Communications channels permit **manipulate** of vehicle held data/code
			5.3 Communications channels permit **overwrite** of vehicle held data/code
			5.4 Communications channels permit **erasure** of vehicle held data/code
			5.5 Communications channels permit introductionof data/code to the vehicle (write data code)
	6	Communication channels permit untrusted/unreliable messages to be accepted or are vulnerable to session hijacking/replay attacks	6.1 Accepting information from an **unreliable or untrusted source**
			6.2 **Man in the middle** attack/ session hijacking
			6.3 **Replay attack**, for example an attack against a communication gateway allows the attacker to downgrade software of an ECU or firmware of the gateway
	7	Information can be readily disclosed. For example, through eavesdropping on communications or through allowing unauthorized access to sensitive files or folders	7.1 **Interception of information** / interfering radiations / monitoring communications
			7.2 Gaining **unauthorized access** to files or data
	8	Denial of service attacks via communication channels to disrupt vehicle functions	8.1 **Sending** a large number of garbage **data** to vehicle information system, **so that it is unable to**

			provide services in the normal manner
			8.2 **Black hole attack**, in order to disrupt communication between vehicles the attacker is able to block messages between the vehicles
	9 An unprivileged user is able to gain **gain privileged access**, for privileged access to vehicle systems example root access		9.1 An unprivileged user is able to
	10 Viruses embedded in communication systems		10.1 **Virus** embedded in communication media infects vehicle media are able to infect vehicle systems
	11	Messages received by the vehicle (for example X2V or diagnostic messages), or transmitted within it, contain malicious content	11.1 Malicious **internal** (e.g. CAN) **messages**
			11.2 Malicious **V2X messages,** e.g. infrastructure to vehicle or vehicle-vehicle messages (e.g. CAM, DENM)
			11.3 Malicious diagnostic messages
			11.4 Malicious **proprietary messages** (e.g. those normally sent from OEM or component/system/function supplier)
4.3.3. Threats to vehicles regarding their update procedures	12	Misuse or compromise of update procedures	12.1 Compromise of **over the air software update procedures**. This includes fabricating the system update program or firmware
			12.2 Compromise of **local/physical software update procedures**. This includes fabricating the system update program or firmware
			12.3 The **software** is **manipulated before the update process** (and is therefore corrupted), although the update process is intact
			12.4 **Compromise** of cryptographic keys of the software provider **to allow invalid update**

	13 It is possible to deny legitimate update server or network to updates and/or unlock of customer specific features	13.1 Denial of Service attack against **prevent rollout of critical software updates**

4.3.4 Threats to vehicles regarding unintended human actions facilitating a cyber attack	15	Legitimate actors are able to take actions that would unwittingly facilitate a cyber-attack	15.1 Innocent victim (e.g. owner, operator or maintenance engineer) being **tricked into taking an action** tounintentionally load malware or enable an attack
			15.2 **Defined security procedures** are not followed
4.3.5 Threats to vehicles regarding their external connectivity and connections	16	Manipulation of the connectivity of vehicle functions enables a cyber-attack, this can include telematics; systems that permit remote operations; and systems using short range wireless communications	16.1 Manipulation of **functions designed to remotely operate systems**, such as remote key, immobilizer, and charging pile
			16.2 **Manipulation of vehicle telematics** (e.g. manipulate temperature measurement of sensitive goods, remotely unlock cargo doors)
			16.3 Interference with **short range wireless systems** or sensors
	17 Hosted 3rd party software, e.g. 17.1 **Corrupted applications**, or those with poor software entertainment applications, used as a security, used as a method to attack vehicle systems means to attack vehicle systems		
	18	Devices connected to external interfaces e.g. USB ports, OBD port, used as a means to attack vehicle systems	18.1 **External interfaces** such as USB or other ports used as a point of attack, for example through code injection
			18.2 Media infected with a **virus** connected to a vehicle system
			18.3 **Diagnostic access (e.g. dongles in OBD port)** used to facilitate an attack, e.g. manipulate vehicle parameters (directly or indirectly)

4.3.7 Potential vulnerabilities that could be exploited if not sufficiently protected or hardened	26	Cryptographic technologies can be compromised or are insufficiently applied	26.1 Combination of short **encryption keys** and long period of validity enables attacker to break encryption
			26.2 Insufficient use of cryptographic algorithms to protect sensitive systems

			26.3 Using already or soon to be deprecated **cryptographic algorithms**
	27 Parts or supplies could be 27.1 **Hardware or software, engineered to enable an attack** or compromised to permit vehicles to be fails to meet design criteria to stop an attack attacked		

28	Software or hardware development permits vulnerabilities	28.1	**Software bugs**. The presence of software bugs can be a basis for potential exploitable vulnerabilities. This is particularly true if software has not been tested to verify that known bad code/bugs is not present and reduce the risk of unknown bad code/bugs being present
		28.2	**Using remainders** from development (e.g. debug ports, JTAG ports, microprocessors, development certificates, developer passwords, ...) can permit access to ECUs or permit attackers to gain higher privileges
29	Network design introduces vulnerabilities	29.1 **Superfluous internet ports left open**, providing access to network systems	
		29.2	Circumvent **network separation** to gain control. Specific example is the use of unprotected gateways, or access points (such as truck-trailer gateways), to circumvent protections and gain access to other network segments to perform malicious acts, such as sending arbitrary CAN bus messages
31 Unintended transfer of data can 31.1 Information breach. Personal data may be leaked when the occur **car changes user** (e.g. is sold or is used as hire vehicle with new hirers)			
32	Physical manipulation of systems can enable an attack	32.1	**Manipulation of electronic hardware**, e.g. unauthorized electronic hardware added to a vehicle to enable "man-in-the-middle" attack **Replacement of authorized electronic hardware** (e.g., sensors) with unauthorized electronic hardware **Manipulation of the information** collected by a sensor (for example, using a magnet to tamper with the Hall effect sensor connected to the gearbox)

From Part B with some mitigations

Part B. Mitigations to the threats intended for vehicles 1. Mitigations for "Vehicle communication channels"
Mitigations to the threats which are related to "Vehicle communication channels" are listed in Table B1.
Table B1
Mitigation to the threats which are related to "Vehicle communication channels"
Table A1 Threats to "Vehicle communication channels" Ref Mitigation reference
4.1 Spoofing of messages (e.g. 802.11p V2X during M10 The vehicle shall verify the authenticity and integrity of platooning, GNSS messages, etc.) by impersonation messages it receives
4.2 Sybil attack (in order to spoof other vehicles as if M11 Security controls shall be implemented for storing cryptographic there are many vehicles on the road) keys (e.g., use of Hardware Security Modules)

5.1 Communication channels permit code injection into M10 vehicle held data/code, for example tampered M6 software binary might be injected into the communication stream The vehicle shall verify the authenticity and integrity of messages it receives Systems shall implement security by design to minimize risks		
5.2 Communication channels permit manipulation of vehicle held data/code	M7	Access control techniques and designs shall be applied to protect system data/code
5.3 Communication channels permit overwrite of vehicle held data/code		
5.4 Communication channels permit erasure of vehicle 21.1 held data/code		
5.5 Communication channels permit introduction of data/code to vehicle systems (write data code)		
6.1 Accepting information from an unreliable or M10 The vehicle shall verify the authenticity and integrity of untrusted source messages it receives		
6.2 Man in the middle attack / session hijacking	M10	The vehicle shall verify the authenticity and integrity of
6.3 Replay attack, for example an attack against a communication gateway allows the attacker to downgrade software of an ECU or firmware of the gateway		messages it receives
7.1 Interception of information / interfering radiations / M12 Confidential data transmitted to or from the vehicle shall be monitoring communications protected		
7.2 Gaining unauthorized access to files or data M8 Through system design and access control it should not be possible for unauthorized personnel to access personal or system critical data. Example of Security Controls can be found in OWASP		
8.1 Sending a large number of garbage data to vehicle M13 Measures to detect and recover from a denial of service attack information system, so that it is unable to provide shall be employed services in the normal manner		
8.2 Black hole attack, disruption of communication M13 Measures to detect and recover from a denial of service attack between vehicles by blocking the transfer of shall be employed messages to other vehicles		
9.1 An unprivileged user is able to gain privileged M9 Measures to prevent and detect unauthorized access shall be access, for example root access employed		
10.1 Virus embedded in communication media infects M14 Measures to protect systems against embedded viruses/malware vehicle systems should be considered		
11.1 Malicious internal (e.g. CAN) messages M15 Measures to detect malicious internal messages or activity should be considered		

11.2 Malicious V2X messages, e.g. infrastructure to vehicle or vehicle-vehicle messages (e.g. CAM, DENM)	M10	The vehicle shall verify the authenticity and integrity of messages it receives
11.3 Malicious diagnostic messages		
11.4 Malicious proprietary messages (e.g. those normally sent from OEM or component/system/function supplier)		

Part 2 concerns the update process over the air or not. Is it in the scope ?

Part B **Mitigations for "External connectivity and connections" Table B4**

Table A1 Threats to "External connectivity and connections" Ref		Mitigation reference	
16.1 Manipulation of functions designed to remotely operate vehicle systems, such as remote key, immobiliser, and charging pile		M20	Security controls shall be applied to systems that have remote access
16.2 Manipulation of vehicle telematics (e.g. manipulate temperature measurement of sensitive goods, remotely unlock cargo doors)			
16.3 Interference with short range wireless systems or sensors			
17.1	Corrupted applications, or those with poor software security, used as a method to attack vehicle systems	M21	Software shall be security assessed, authenticated and integrity protected. Security controls shall be applied to minimise the risk from third party software that is intended or foreseeable to be hosted on the vehicle
1. 18.1 External interfaces such as USB or other ports used as a 18.2 Media infected with viruses connected to the vehicle		M22	Security controls shall be applied to external interfaces point of attack, for example through code injection
18.3 Diagnostic access (e.g. dongles in OBD port) used to facilitate an attack, e.g. manipulate vehicle parameters (directly or indirectly)		M22	Security controls shall be applied to external interfaces

ANNEX – ETSI TR 102 863 TABLES 8 AND 14

ID	Threat	ITS-S Problem Area	Weaknesses	Threat Agent	Attack interface
V-V1	- Message saturation	Intrinsic high density of ITS message traffic due to broadcasting and beaconing in V2V systems	The time taken by an ITS-S (Vehicle) to process a high volume of real or spurious messages or fabricated queue entries could: (1) cause it to miss important incoming ITS messages (2) cause it to delay or miss the sending of outgoing ITS messages or relaying of incoming ITS messages (3) leave it with no resources free for other essential tasks such as monitoring sensors and updating driver-displays (4) leave it with no resources free for other essential tasks such as monitoring sensors and updating driver-displays	Malware installed on target ITS-S (Vehicle) filling the incoming message queue with spurious but valid messages Malicious ITS-S broadcasting a high level of ITS message traffic	A, B (also on behalf of K)
		Lack of flow control in V2V broadcast messaging			
		Absence of addressing in broadcast messages meaning source cannot be identified so malicious and irrelevant messages can only be rejected by the application, not at the network layer in the ITS stack			
		The random re-attempt period in the "Listen before send" message transmission method does not make optimum use of the available bandwidth			

ID	Threat	ITS-S Problem Area	Weaknesses	Threat Agent	Attack interface
V-V2	- Jamming of radio signals	Inability of the ITS-S (Vehicle) to quickly detect and isolate interference on radio channels	Transmissions to and from an ITS-S (Vehicle) can be lost while interference is detected and mitigated	External jammer equipment	A, B
V-V3	- Injection of false messages - Manipulation of ITS messages en route	Absence of addressing in broadcast messages meaning source cannot be identified so malicious and irrelevant messages can only be rejected at the application layer, not at the network layer in the ITS stack	An ITS-S (Vehicle) is unable to determine quickly whether a received message is valid and from a legitimate user and then acts on information received in the message Relayed messages are open to manipulation in an ITS-S en route. Received messages that are intended for relaying can be withheld	Malware on ITS-S (Vehicle or Roadside) within range Equipment posing as a genuine ITS-S (Vehicle) or as an RSU sending valid but irrelevant ITS messages	A, B
V-V4	- Masquerade as ITS-S (Vehicle or Roadside) or ITS network	CAM and DNM messages do not include any form of identification information	An ITS-S (Vehicle) is unable to determine quickly whether a received message is valid and from a legitimate user and then acts on information received in the message The contents of the LDM can be incorrectly modified by received messages containing false time, position or status information or by maliciously planted software	Equipment posing as a genuine ITS-S (Vehicle) or as an RSU sending false information in ITS messages that are otherwise valid	A, B (also on behalf of K)
		Vehicle-to-Vehicle messages include no validation or legitimacy checks			
V-V5	- Masquerade for fabrication of messages	CAM and DNM messages do not include any form of identification information	An ITS-S (Vehicle) is able to perform only basic checks on the validity of a received message and its contents The contents of the LDM can be incorrectly modified by received messages containing false time, position or status information or by maliciously planted software	Equipment posing as a genuine ITS-S (Vehicle) or as an RSU sending false information in ITS messages that are otherwise valid Malicious application in the ITS network sending false information in ITS messages that are otherwise valid	A, B
		Vehicle-to-Vehicle messages include no validation or legitimacy checks			

ID	Threat	ITS-S Problem Area	Weaknesses	Threat Agent	Attack interface
V-V6	- Replay of "expired" (old) messages - Wormhole attacks - GNSS spoofing	Uncertainty regarding how timestamps are created and how to use them to check the validity of messages	An ITS-S (Vehicle) is unable to validate when or where a received message was originally generated	Equipment posing as a genuine ITS-S (Vehicle) or as an RSU sending "expired" information in ITS messages that are otherwise valid Equipment posing as a genuine ITS-S (Vehicle) or as an RSU sending information in ITS messages that are valid except for the source location	A, B
V-V7	- Malicious isolation of one or more ITS-S (Vehicle) (black hole)	ITS-S (Vehicle) memory is can be modified by information received over the air interface	Malware can be initiated, accessed or installed over the air	Malware on ITS stations that disables some or all functionality of one or more of the ability to create, process, receive and send ITS messages	A, B
V-V8	- Eavesdropping - Traffic analysis - Location tracking	Broadcast messages are in general intended for all ITS-S within range.	All ITS messages (even those associated with subscription services) are broadcast in the 5,9 GHz band and can, therefore, be intercepted by any capable receiver Some ITS BSA messages reveal the geographic location of the sending ITS-S	Equipment posing as a genuine ITS-S (Vehicle) or as an RSU receiving information for malicious analysis of content and recording on message patterns, etc.	A, B
		Absence of addressing in broadcast messages meaning that non-ITS-S equipment can also receive ITS messages			

ID	Threat	ITS-S Problem Area	Weaknesses	Threat Agent	Attack interface
V-V9	- Denial of transmission	CAM and DNM messages do not include any form of identification information	There is no requirement for an ITS-S (Vehicle) to maintain an auditable log of all messages sent and received by it. Such a log would quickly become very large due to the high density of ITS messages.	Equipment posing as a genuine ITS-S (Vehicle) or as an RSU sending false information in ITS messages that are otherwise valid Malware installed on target ITS-S (Vehicle) creating and sending false information in ITS messages that are otherwise valid	A, B (also on behalf of K)
		Vehicle-to-Vehicle messages include no validation or legitimacy checks			
		ITS-S (Vehicle) cannot positively identify relevant information to maintain record of the originator of ITS messages causing harm to the ITS-S (Vehicle)			

Table 8: List of Vulnerabilities for ITS-S (Vehicle) ToE

ID	Threat	ITS-S Problem Area	Weakness	Threat Agent	Attack interface
V-R1	- Injection of a high volume of false emergency vehicle warning messages	CAM and DNM messages do not include any form of identification information	An RSU is unable to quickly determine whether a received message contains accurate information and is from a legitimate emergency services vehicle and acts by relaying the message. An RSU can only check whether the message is valid and comes from a valid source The time taken by an RSU to process a high volume of real or spurious messages could cause it to miss important incoming ITS messages	Equipment posing as a genuine Emergency Vehicle sending false information in ITS messages that are otherwise valid Equipment replaying "expired" emergency vehicle warnings	B
		Absence of addressing in broadcast messages meaning source cannot be identified so malicious and irrelevant messages can only be rejected on the application layer, not at the network layer in the ITS stack			
V-R2	- Message saturation	CAM and DNM messages do not include any form of identification information			B

ID	Threat	ITS-S Problem Area	Weakness	Threat Agent	Attack interface
		Absence of addressing in broadcast messages meaning source cannot be identified so malicious and irrelevant messages can only be rejected on the application layer, not at the network layer in the ITS stack	The time taken by an ITS-S (Vehicle) to process a high volume of real or spurious messages or fabricated queue entries could cause it to miss important incoming ITS messages The time taken by an RSU to process a high volume of real or spurious messages or fabricated queue entries could leave it with no resources free for other essential tasks such as relaying and acting upon emergency vehicle warnings or other safety-related messages	Malware installed on target RSU filling the incoming message queue with spurious but valid messages Malicious ITS-S broadcasting a high level of ITS message traffic	
		Uncertainty regarding identification, authentication and authorization of ITS application and information on an RSU			
V-R3	- Radio jamming	Inability of an RSU to quickly detect and isolate interference on radio channels	Transmissions to and from an RSU can be lost while interference is detected and mitigated	External jammer equipment	B
V-R4	- Injection of false messages	Absence of addressing in broadcast messages meaning source cannot be identified so malicious and irrelevant messages can only be rejected on the application layer, not at the network layer in the ITS stack	An RSU is unable to quickly determine whether a received message contains accurate information and is from a legitimate user and acts by relaying the message. An RSU can only check whether the message is valid and comes from a valid source The time taken by an RSU to process a high volume of real or spurious messages could cause it to miss important incoming ITS messages An RSU is unable to validate when a received message was originally generated	Equipment posing as a genuine ITS-S (Vehicle) sending false information in ITS messages that are otherwise valid	B
		Uncertainty regarding how timestamps are created and how to use them to heck the validity of messages			
V-R5	- Replay of "expired" (old) messages - Wormhole attack - GNSS spoofing	Uncertainty regarding how timestamps are created and how to use them to heck the validity of messages	An RSU is unable to validate when a received message was originally generated	Equipment posing as a genuine ITS-S sending "expired" information in ITS messages that are otherwise valid GNSS spoofing equipment	B

ID	Threat	ITS-S Problem Area	Weakness	Threat Agent	Attack interface
V-R6	- Emergency vehicle masquerade	CAM and DNM messages do not include any form of identification information	An RSU is unable to quickly determine whether a received message contains accurate information and is from a legitimate emergency services vehicle and acts by relaying the message. An RSU can only check whether the message is valid and comes from a valid source	ITS-S masquerading as Emergency Vehicle Equipment posing as a genuine Emergency Vehicle	B
		Absence of addressing in broadcast messages meaning source cannot be identified so malicious and irrelevant messages can only be rejected on the application layer, not at the network layer in the ITS stack			
V-R7	- Eavesdropping - Traffic analysis - Location tracking	Broadcast messages are in general intended for all ITS-S within range	All ITS messages (even those associated with subscription services) are broadcast in the 5,9 GHz band and can, therefore, be intercepted by any capable receiver Some ITS BSA messages reveal the geographic location of the sending ITS-S	Equipment posing as a genuine ITS-S (Vehicle) or RSU recording information in ITS messages for malicious analysis of content, behavioral patterns, etc.	B, J
		Absence of addressing in broadcast messages meaning that non-ITS-S equipment can also receive ITS messages			
V-R8	- Transaction tampering	Broadcast messages are in general intended for all ITS-S within range	All ITS messages (even those associated with subscription services) are broadcast in the 5,9 GHz band and can, therefore, be intercepted by any capable receiver An RSU is unable to validate either when a received message was originally generated or whether any subscription information in the messages is valid	Equipment posing as a valid ITS-S (Vehicle)	B
		Absence of addressing in broadcast messages meaning that non-ITS-S equipment can also receive ITS messages			
		Uncertainty regarding how timestamps are created and how to use them to heck the validity of messages			
V-R9	- Denial of transmission	CAM and DNM messages do not include any form of identification information			B, J

ID	Threat	ITS-S Problem Area	Weakness	Threat Agent	Attack interface
		RSU cannot positively identify relevant information to maintain record of the originator of ITS messages causing harm to the RSU	There is no requirement for an RSU to maintain an auditable log of all and specific types of messages sent and received by it. Such a log should be maintainable for an RSU	Equipment posing as a genuine RSU or ITS-S (Vehicle) sending false information in ITS messages that are otherwise valid Malware installed on target RSU creating and sending false information in ITS messages that are otherwise valid Valid ITS-S (Vehicle) with motivation to deny sending or receiving ITS messages, such as ITS messages from authorities	

Table 9: List of Vulnerabilities for ITS-S (Roadside)