Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

Future generation Security for smart and connected Cars - FuSeCar

Deliverable D4.1: Local misbehavior detection algorithm

WP4: Misbehavior detection for vehicular communication protocols and architectures

Authors:
Giovanni Gambigliani Zoccoli[1], Mattia Trabucco[2], Mauro Andreolini[2], Luca Ferretti[2], and Mirco Marchetti[1]
{name.surname}@unimore.it

[1]Department of Engineering "Enzo Ferrari"
[2]Department of Physics, Informatics and Mathematics
University of Modena and Reggio Emilia

Current revision: R1.1
Delivery date: April 28th, 2025

Revision history

Authors	Changes	Date	Revision
Giovanni Gambigliani Zoccoli Mattia Trabucco Mirco Marchetti	Creation of the document, tentative structure	November 3rd, 2024	R0.1
Mattia Trabucco Mauro Andreolini Luca Ferretti Mirco Marchetti	First draft of Section 2	November 5th, 2024	R0.2
Giovanni Gambigliani Zoccoli Luca Ferretti Mauro Andreolini Mirco Marchetti	First draft of Section 3	December 6th, 2024	R0.3
Luca Ferretti Mauro Andreolini Mirco Marchetti	Refinement of the draft of Section 3	December 17th, 2025	R0.4
Giovanni Gambigliani Zoccoli Mattia Trabucco Mirco Marchetti	First draft of Section 4	January 22th, 2025	R0.5
Mattia Trabucco Luca Ferretti Mauro Andreolini Mirco Marchetti	Draft of Section 5	February 19th, 2025	R0.6
Luca Ferretti Mauro Andreolini Mirco Marchetti	First draft of complete document	March 25th, 2025	R1.0
Giovanni Gambigliani Zoccoli Mattia Trabucco Luca Ferretti Mauro Andreolini Mirco Marchetti	Revision of document and minor fixes	April 28th, 2025	R1.1

Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA

Contents

1 Introduction

The automotive industry has undergone significant transformation through vehicle domain innovations, evolving from isolated, mechanical vehicles to interconnected systems within Cooperative Intelligent Transportation Systems (C-ITS) [7]. Initially, the focus was on onboard electronics and telematics, but the shift to C-ITS integrates connectivity, enabling vehicles to interact with each other, surrounding infrastructure, and other transportation users. This connectivity provides drivers with real-time information for enhanced safety, efficiency, and sustainability [18]. The integration of technologies such as Vehicle-to-Everything (V2X) communication marks a paradigm shift from standalone vehicles to cooperative ecosystems. This transformation addresses challenges such as traffic congestion, road accidents, and environmental impacts. Key innovations, such as Cellular-V2X (C-V2X), improve communication latency and reliability compared to traditional Dedicated Short Range Communications (DSRC), thereby paving the way for automated and connected mobility [22]. Furthermore, the rise of software-defined vehicles (SDVs) emphasizes software over hardware, enabling rapid innovation through over-the-air updates and modular architectures. The incorporation of artificial intelligence for predictive maintenance, personalized user experiences, and advanced driver assistance systems (ADAS) contributes to smarter, more responsive vehicles. As vehicles become increasingly connected, they integrate into larger intelligent transportation networks, facilitating seamless communication with smart city infrastructures and enabling features such as dynamic route optimization based on real-time traffic data.

1.1 C-ITS and VANETs

A crucial aspect of C-ITS is its reliance on Vehicular Ad-hoc Networks (VANETs) as the foundational communication framework. VANETs enable dynamic, ad-hoc networking among vehicles, supporting real-time data exchange, and serve as the cornerstone for C-ITS applications [1]. These networks are vital for facilitating Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) interactions, enabling communication between vehicles and roadside units (RSUs), traffic signals, and other infrastructure. This constant exchange of information is key to safety-critical applications, such as collision avoidance, cooperative driving, and situational awareness, allowing vehicles to "see" beyond their immediate surroundings and make more informed decisions [6].

Through VANETs, real-time cooperative awareness is established, with vehicles sharing critical information such as position, speed, and heading. This enables proactive actions like avoiding collisions and optimizing traffic flow. VANETs also support advanced applications like platoon driving, where vehicles travel closely together to reduce air resistance and improve fuel efficiency, and intersection management, where vehicles can communicate with traffic signals to reduce congestion and waiting times.

In addition to safety, VANETs also support Vehicle-to-Cloud (V2C) communication, allowing vehicles to interact with cloud-based services for enhanced traffic management, fleet monitoring, and predictive maintenance. This allows vehicles to share data on road conditions, weather, and traffic, which can then inform decisions such as real-time traffic rerouting or environmental monitoring.

The inherent mobility and dynamic nature of VANETs address challenges like high mobility, variable topology, and intermittent connectivity. Through multi-hop communication, where vehicles relay messages to one another, VANETs maintain connectivity in highly dynamic environments, such as highways or urban areas with dense traffic. Furthermore, the adherence to standards like IEEE 802.11p and ETSI ITS-G5 ensures interoperability across different manufacturers and infrastructure providers, promoting global scalability for C-ITS systems.

Beyond basic safety, VANETs play a crucial role in integrating smart city infrastructures, enabling communication with smart traffic systems, parking management, and environmental monitoring. This helps improve both urban mobility and energy efficiency. As C-ITS evolves, VANETs will be central to supporting autonomous driving and smart mobility services, ensuring reliable and low-latency communication for autonomous vehicles

operating alongside human-driven vehicles.

1.2 V2X Communication

At the core of VANETs and C-ITS are V2X messages, which encompass various communication types, including Vehicle-to-Vehicle (V2V) for collision avoidance, Vehicle-to-Infrastructure (V2I) for traffic signal coordination, Vehicle-to-Pedestrian (V2P) for pedestrian safety alerts, and Vehicle-to-Network (V2N) for broader connectivity [22]. These messages transmit critical data, such as position, speed, heading, acceleration, and event notifications, in standardized formats like Cooperative Awareness Messages (CAMs) and Decentralized Environmental Notification Messages (DENMs) for ETSI-based systems, or Basic Safety Messages (BSMs) in IEEE-based systems [13]. The periodic and event-driven nature of these messages enables proactive decision-making in connected and automated driving scenarios.

V2X communication is supported by a layered protocol stack, defined by ETSI ITS-G5 in Europe and IEEE WAVE (Wireless Access in Vehicular Environments) in the United States. At the physical and medium access control layers, both stacks rely on IEEE 802.11p, a variant of IEEE 802.11 designed for low-latency and high-mobility vehicular environments. Above the access layer, the networking and transport layers support both IPv6-based and non-IP communication, allowing flexibility to meet diverse application needs. The facilities layer defines standardized message formats, such as CAMs, DENMs, and map-related messages, enabling interoperability across different vehicle manufacturers and infrastructure providers. A dedicated security layer, based on Public Key Infrastructure (PKI), ensures message authentication, integrity protection, and privacy-preserving mechanisms through pseudonym certificates. This layered architecture enables the efficient dissemination of safety-critical information while maintaining interoperability and security across diverse vehicular environments.

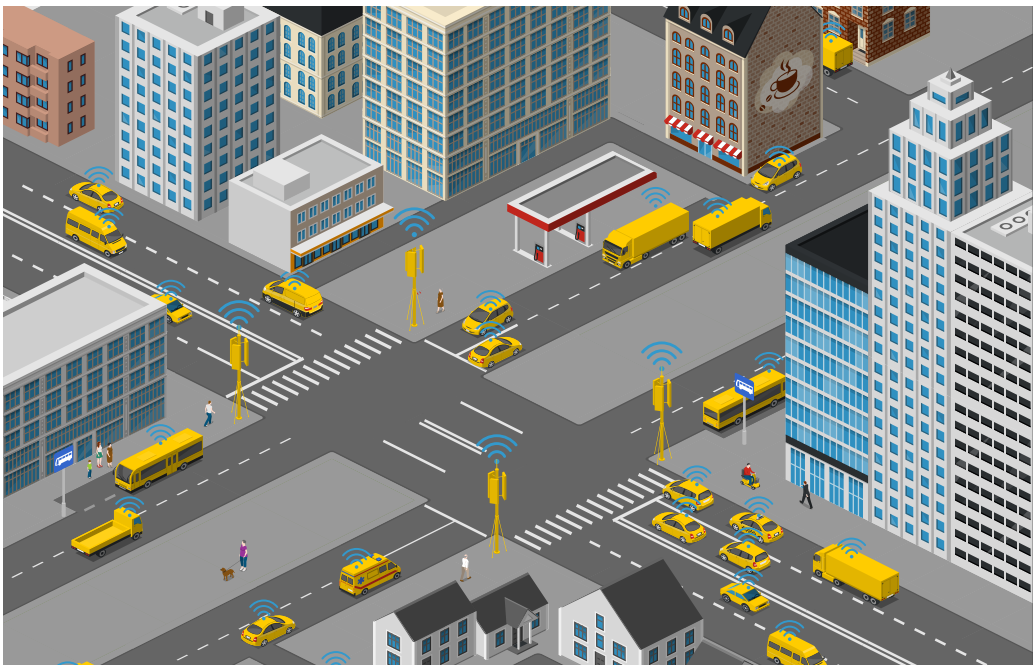

Figure 1: Example of C-ITS scenario

V2X communication significantly enhances transportation efficiency, road safety, and the overall driving experience by facilitating direct and cooperative interactions between vehicles and infrastructure. Use cases include platoon driving to reduce fuel consumption, cooperative perception to extend vehicle sensing ranges, and priority management for emergency vehicles at intersections. As V2X technology evolves with the advent of 5G, it is expected to support ultra-reliable low-latency communications and massive connectivity, further

enabling advanced automated driving functions and dense urban deployments. An example of a VANET-enabled C-ITS scenario is shown in Figure 1, where vehicles and infrastructure entities cooperate through V2X communications in proximity to road intersections.

The remainder of this document is structured as follows. Section 2 provides a detailed background on C-ITS security mechanisms, attacker models, and existing misbehavior detection approaches. Section 3 presents the proposed local misbehavior detection system, which leverages vehicle onboard sensors and fuses them with V2X information to improve detection accuracy and resilience against stealthy attacks. Finally, Section 6 present the conclusion of this work.

2 Background

In this section we will provide a detailed description of the required knowledge for the understanding of this document. In particular, Section 2.1 describes the Public key Infrastructure as a pillar of the VANETs network, while Section 2.2 analyze the two types of attackers considered as threat model of a general VANETs. Lastly, Section 2.3 introduce the SixPack attack, a stealthy attack which mimic the behavior of real vehicle using Vehicle-to-Vehicle communication.

2.1 PKI in VANETs

Public Key Infrastructure (PKI) has long been the backbone of security in vehicular ad-hoc networks (VANETs). PKI ensures that only legitimate vehicles and roadside units (RSUs) are allowed to communicate by using *digital certificates* and *public/private key pairs* [26]. The public key is used to verify the authenticity of messages, while the private key is used to sign messages, ensuring their integrity and preventing tampering. This setup effectively defends against impersonation and man-in-the-middle attacks.

Within the European context, the ETSI (European Telecommunications Standards Institute) standards define how PKI is structured and used for vehicular communication. The technical specification `ETSI TS 103 097` defines the security header and certificate formats for Cooperative ITS (C-ITS) applications, specifying how certificates should be formatted and included in ITS messages to support secure data structures [8]. This standard profiles and reuses elements from the North American IEEE 1609.2 standard (used in IEEE WAVE security), defining appropriate extensions and constraints for European deployments. In particular, `ETSI TS 103 097` specifies secure data structures (including security headers and signed data), certificate formats, and security profiles that dictate how certificates are attached to periodically transmitted messages like CAMs and DENMs [17].

In practice, each vehicle or RSU is provisioned with a public–private key pair and a set of certificates issued by a trusted *Certification Authority (CA)*. Certificates in the ETSI framework include *Root CA certificates*, *subordinate CA certificates*, *Authorization Tickets*, and *Enrolment Credentials*, each serving roles in establishing identity and trust chains within the PKI system. To reduce communication overhead, `ETSI TS 103 097` also allows the use of *certificate digests* in safety messages, where a short digest of a certificate is included in most messages, with full certificates sent less frequently; vehicles can request the full certificate when needed to verify a digest [24]. This mechanism balances security with bandwidth constraints inherent in high-frequency V2X communication.

The ETSI ITS PKI framework does not operate alone. It is supported by complementary specifications such as `ETSI TS 102 941` [9], which defines trust and privacy management including certificate revocation and trust list management, and `ETSI TS 103 601` [10], which addresses security management messages and distribution protocols for PKI deployment. These standards work together to implement a coherent security credential management system (SCMS) that ensures authorized communication across vehicles and infrastructure.

Other international standards also influence vehicular PKI architectures. For example, the `IEEE 1609.2` family (used in the United States) defines message domain security services, including certificate usage and cryptographic message formats, which `ETSI TS 103 097` aligns with for cross-compatibility [21]. Additionally, broader automotive cybersecurity standards such as `ISO/SAE 21434` emphasize secure key management and secure software practices within vehicle systems, reinforcing PKI's role in the overall security lifecycle. These standards together ensure that the PKI not only authenticates message sources but also integrates with vehicle-wide cybersecurity measures.

Despite its robustness in authenticating messages, PKI does not inherently validate the semantic correctness of the data carried in V2X messages. Vehicles may still broadcast incorrect positions, inaccurate speeds, or false hazard warnings while possessing valid certificates, thereby exploiting the trust granted by PKI. This

highlights the need for additional mechanisms such as *Misbehavior Detection Systems (MDS)* to detect and mitigate internal attacks that leverage legitimate credentials.

2.2 VANETs attacker and Misbehavior Detection System

In a VANET, we can define two main types of attackers: *external attackers* and *internal attackers*.

External attackers are typically unauthorized entities trying to disrupt communication, often using techniques like *jamming* or *spoofing*. **Jamming** interferes with communication channels by flooding the frequency with noise, preventing legitimate vehicles and infrastructure from transmitting critical safety messages. **Spoofing** involves impersonating legitimate vehicles or roadside units (RSUs), injecting fake messages into the network to mislead other nodes. These attacks can have serious consequences, such as preventing emergency alerts from reaching drivers or causing vehicles to misinterpret their surroundings [28]. These attacks are detectable by PKI, which ensures that only authenticated entities can broadcast messages. PKI-based systems can verify the identity of the message sender, preventing unauthorized entities from disrupting communications or by revoking authorization to malicious users.

On the other hand, *internal attackers* are authorized entities within the network that have valid credentials, making them much more difficult to detect. Since they are part of the trusted system, internal attackers can broadcast messages that appear legitimate, making traditional security measures such as PKI insufficient. PKI only ensures the authenticity of the message source, not the content of the message itself. An internal attacker might falsify position, speed, or hazard information, leading to unsafe driving decisions or traffic disruptions. For example, an attacker could send false position data that causes surrounding vehicles to take incorrect actions, such as braking or changing lanes unnecessarily, which could result in accidents. This ability to manipulate the system without raising alarms makes internal attackers particularly dangerous, as they exploit the inherent trust in the system.

To address this limitation, recent academic efforts have focused on developing *Misbehavior Detection Systems (MDS)* that analyze the content and context of the messages being exchanged. These systems go beyond PKI and attempt to detect anomalous behavior that might not be immediately apparent from identity verification alone. One such solution is the *GOLIATH* framework, which integrates blockchain technology for decentralized verification of messages. By storing messages in an immutable ledger, GOLIATH can detect tampered data and provide an additional layer of trust and transparency, making it harder for internal attackers to manipulate data without being detected [19]. Other solutions incorporate machine learning models, which are trained on large datasets of driving behavior to identify anomalies or deviations from expected patterns. These models can help detect internal attacks by flagging data that does not conform to typical vehicular behavior.

Another promising approach uses *plausibility and consistency checks*, as seen in the F^2MD [15] framework, which performs these checks to validate the content of V2X messages for inconsistencies. This system cross-references the reported data with expected road conditions, traffic patterns, and vehicle trajectories to determine whether the reported message is plausible. For example, if a vehicle reports a sudden change in position that is inconsistent with surrounding traffic, the system would flag it as potentially malicious.

While these solutions are effective at detecting simple attacks, such as incorrect positioning or basic message inconsistencies, they struggle to identify more complex and stealthy attacks, such as the *SixPack attack* [29]. In a SixPack attack, the malicious vehicle mimics the behavior of real vehicles, sending data that fits typical driving patterns and traffic conditions, making it difficult to detect through traditional content analysis. Since the attack focuses on mimicking realistic behavior rather than broadcasting false data, it can evade detection systems that rely solely on plausibility checks or statistical models. SixPack attacks highlight the challenge of detecting sophisticated and stealthy attacks that do not exhibit obvious anomalies in the content of the V2X messages but instead carefully simulate normal driving behavior to blend in with the network.

2.3 SixPack Attack

The *SixPack attack*, introduced as a dynamic internal attack on Vehicular Ad-hoc Networks (VANETs), was originally designed to manipulate the behavior of C-ITS (Cooperative Intelligent Transport Systems) entities by simulating the activation of the braking system, leading to unsafe driving decisions and traffic disruptions. In its original version, SixPack included six phases: Start, Init, FakeBrake, Recovery, Rejoin, and Stop, each phase contributing to the attack's success in evading detection by misbehaving vehicles while maintaining the appearance of normal driving behavior.

SixPack $v2$ builds upon this foundation by enhancing the core phases of the attack, namely *FakeBrake*, *Recovery*, and *Rejoin*, to improve its evasion capabilities. The attack's primary goal is to manipulate the digital representation of the vehicle (its position, speed, and braking status) in the V2X messages while maintaining consistency with the actual vehicle's trajectory, thereby avoiding detection by traditional misbehavior detection systems like F^2MD.

In the *FakeBrake* phase, the attacker simulates a sudden activation of the Anti-lock Braking System (ABS), sending a message indicating a sharp deceleration, even though the vehicle itself is not braking. This causes surrounding vehicles and infrastructure to react as if the vehicle is in an emergency braking situation. The attacker then enters the *Recovery* phase, where the vehicle's digital representation is gradually adjusted to match the real position of the vehicle, following a predefined interpolation algorithm. This phase ensures that the digital and physical positions of the vehicle gradually rejoin, making it more challenging for detection systems to notice any discrepancies.

The *Rejoin* phase further refines this process, with an enhanced interpolation mechanism that takes into account the vehicle's actual movement and surrounding traffic, creating a more realistic path that closely matches the vehicle's real trajectory. This avoids triggering the position and speed plausibility checks commonly used in misbehavior detection systems. The major enhancement in *SixPack* $v2$ is the use of a novel path-reconstruction algorithm, which generates a more realistic representation of the vehicle's trajectory. This improvement significantly increases the attack's ability to evade detection, especially in systems like F^2MD that rely on plausibility checks based on the vehicle's reported data. The attack's dynamic nature, combined with realistic path simulation, allows it to remain undetected even by advanced detection mechanisms that analyze the content of V2X messages. The improvements implemented in SixPack $v2$ are visualized in Figure 2, where both versions of SixPack are tested in an equal scenario. The trajectory depicted in Figure 2 illustrates the *recovery* phase of the algorithms, demonstrating how both versions of SixPack produce forged GPS positions to emulate a genuine vehicle.

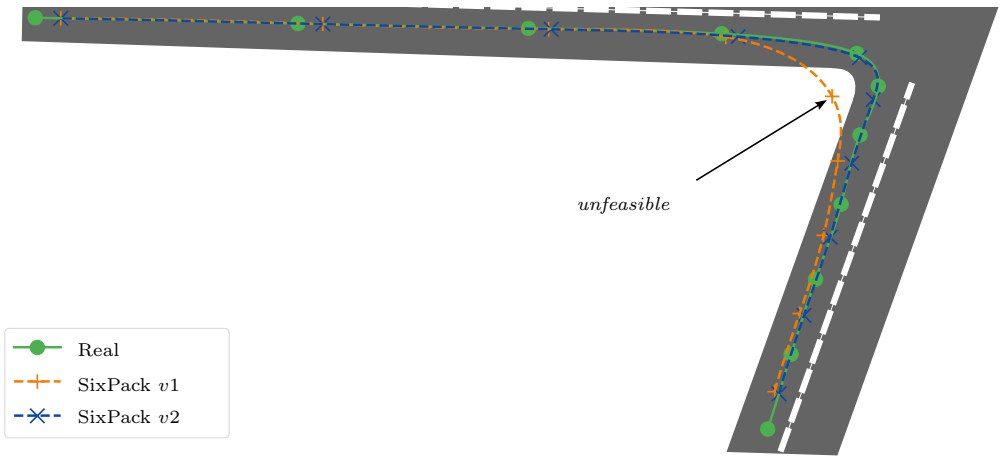

Figure 2: Real trajectory vs. SixPack $v1$ vs. SixPack $v2$

Attacks like *SixPack*, which simulate normal vehicle movements, require more sophisticated detection methods that not only analyze the content of the messages but also incorporate broader contextual information, including the vehicle's interactions with other vehicles and infrastructure. Thus, while systems like F^2MD can effectively detect simpler attacks, $SixPack\ v2$ showcases the limitations of current detection frameworks when faced with stealthier, behavior-mimicking attacks.

The performance of F^2MD in detecting both versions of the SixPack attack are presented in Figure 3. The first version of the SixPack attack is depicted in blue, while the the $SixPack\ v2$ is presented using the orange color. On the $x-axis$ of Figure 3 are reported the three metrics used to evaluate the detection performance of F^2MD (*Precision*, *Recall*, *F1*); while on the $y-axis$ is reported the value of this indexes.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$\mathcal{F}_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

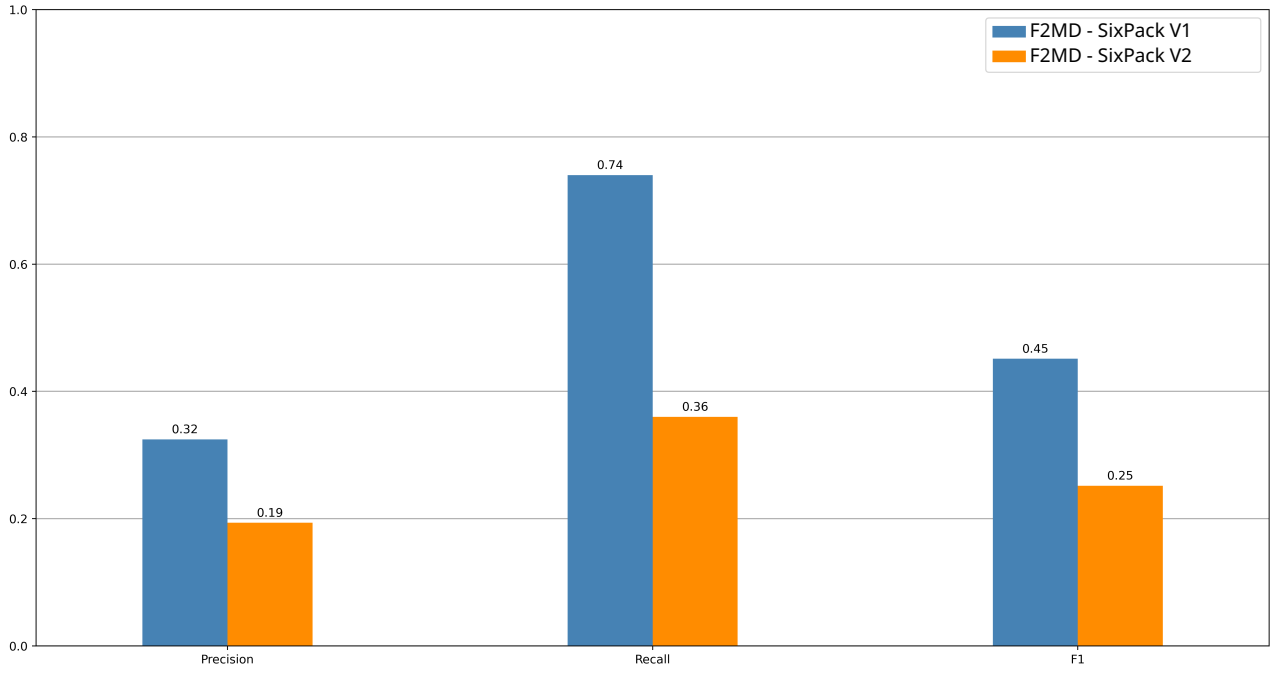

Figure 3: Performance evaluation of F^2MD detection method against SixPack and SixPack$v2$.

The *precision* metric (Formula 1) measures the proportion of correctly identified anomalies among all detected anomalies, while the *recall* metric (Formula 2) reflects the proportion of true anomalies successfully identified out of the total number of actual anomalies. In this context, TP, FP, and FN represent the counts of true positives (anomalous messages correctly detected), false positives (legitimate messages wrongly flagged as anomalies), and false negatives (anomalous messages incorrectly classified as normal), respectively. The *F1* (Formula 3) is calculated as the harmonic mean of precision and recall, providing a balanced summary where both metrics are equally weighted. Each of these performance measures ranges from 0 to 1, with values near 0

indicating poor detection capability (correspondingly, the ability of the attack to evade detection) and values close to 1 indicating almost perfect detection performance (i.e., the inability of the attack to evade detection).

As it possible to observe from Figure 3 $SixPack\ v2$ is able to achieve higher evasion performance, almost halving the detection capabilities of F^2MD framework. In particular, focusing on the *precision* metric the original version of SixPack reach a low value of 0.32 which is even reduced by the new version of the $SixPack\ v2$ reaching 0.19 value. A stronger trend can be observed considering the *recall* index, where the original version of the SixPack attack reach value of about 0.8, while the F^2MD framework against the $SixPack\ v2$ struggles to reach 0.4. As a result the *F1* metric of F^2MD against the $SixPack\ v2$ reach value of 0.25 meaning that the F^2MD framework is not able to detect stealthy attacks like the SixPack which mimic the behavior of legitimate vehicle. Moreover, the performance increase reached by the $SixPack\ v2$ highlight that the improvements of trajectory reconstruction algorithm decrease the detection capabilities of misbehavior detector based on the only analysis of the content of the V2X messages.

3 Design of the Local Misbehavior Detection System

This section presents a novel detection methodology designed to identify stealthy attacks in C-ITS communication using information extracted from received V2X message and external perceptual systems. Our detection mechanism is based on the assumption that each node of the C-ITS participating in the detection task has access to trusted information about the position of nearby vehicles generated by some form of physical system.

Examples of these perceptual systems include LiDARs and radars commonly deployed on vehicles to support *Advanced Driver Assistance Systems* (ADAS), as well as surveillance cameras and other perceptual devices that can be deployed on roadside units. These data sources can be used by a detector to create a simplified digital representation of its surroundings, employing various approaches proposed in the literature [5, 27, 12]. Our detection model does not require a complete and detailed digital model of all nearby objects, since it focuses only on the positions of vehicles in close proximity of the detector. The *Local* detection is presented in Section 3.1 where the two variants are described in two different subsections.

3.1 Vehicle detection

The *Local* detection method is designed for deployment on vehicles participating in C-ITS, and it necessitates them to be equipped with on-board perceptual systems, such as radars, LiDARs and cameras to measure the relative positions of entities in the proximity of the vehicle. We designed and tested two different versions of the *Local* detection model to represent different perceptual systems, providing common coverage areas [20].

Vehicle - Cone

The first version of the *Local* detector (*cone*) represents a vehicle equipped with forward-facing sensors, typical in modern vehicles as a basic form of ADAS. This configuration results in a conical-shaped coverage area (which can be defined as C_a) positioned in front of the vehicle, as illustrated in Figure 4. The conical shape is effective for detecting objects in front of the vehicle, offering a balance between range and coverage width [14].

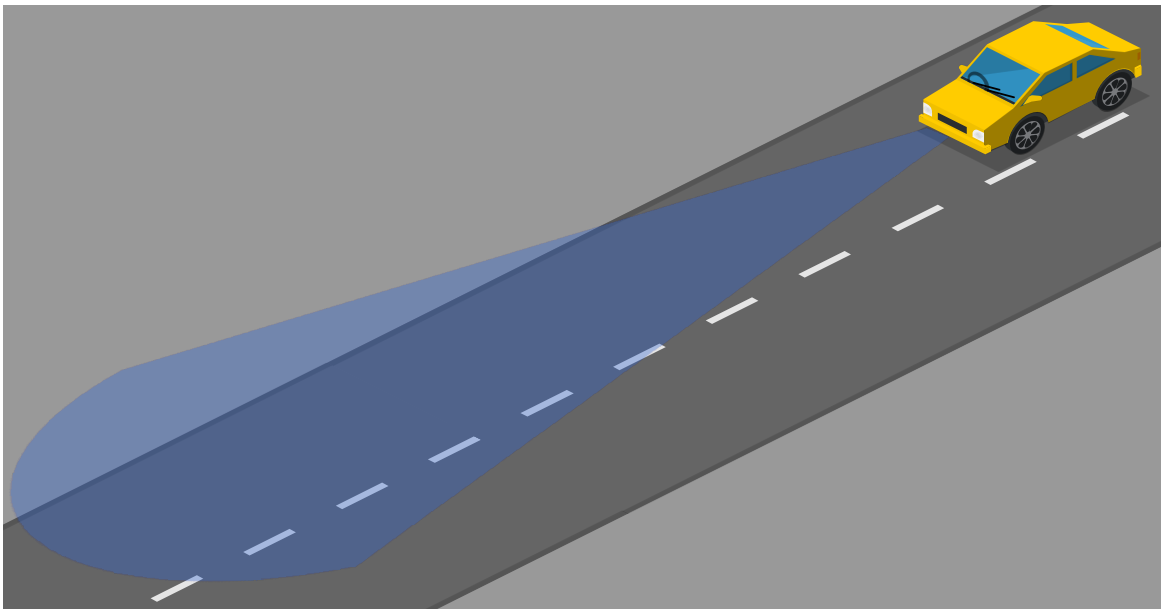

Figure 4: Representation of the conic-shaped coverage area in front of the vehicle

Since the conical shape can be represented as a section of a circular area, we used formula 4 to define the min (α_{min}) and the max (α_{max}) of the circular section. Additionally, GPS coordinates are converted into *radial*

and *polar* coordinates using formula 5 and 6, respectively.

$$[\alpha_{min}, \alpha_{max}] \Rightarrow \alpha \pm \frac{\beta}{2} \tag{4}$$

$$\rho = \sqrt[2]{(x_p - x_c)^2 + (y_p - y_c)^2} \tag{5}$$

$$\theta = \arctan\left[\frac{(y_p - y_c)}{(x_p - x_c)}\right] \tag{6}$$

Here, β represents the width of the cone centered on the same axis with respect to the heading of the vehicle. The GPS coordinates (x_c, y_c) are used to compute the polar coordinates (x_p, y_p) of nearby vehicles.

The detection algorithm uses the conditions expressed in Formula 7 to check whether the GPS coordinates of surrounding vehicles fall within the coverage area C_a of the detector vehicle. In this formula, θ (angle) and ρ (distance) are the polar conversion of the positions of the nearby vehicles, while α_{min} and α_{max} represents the boundary angles of the detector vehicle and R is the maximum distance of the perceptual system.

$$\alpha_{min} \leqslant \theta \leqslant \alpha_{max}, \rho \leqslant R \tag{7}$$

Vehicle - Ellipse

The second version of the *Local* detector (*ellipse*) represents a vehicle equipped with multiple perceptual systems, such as radars and LiDARs, capable of identifying objects all around the vehicle. This results in an elliptical-shaped coverage area (which we defined again as C_a) with the center of the ellipse slightly shifted towards the front of the vehicle [14], as depicted in Figure 5.

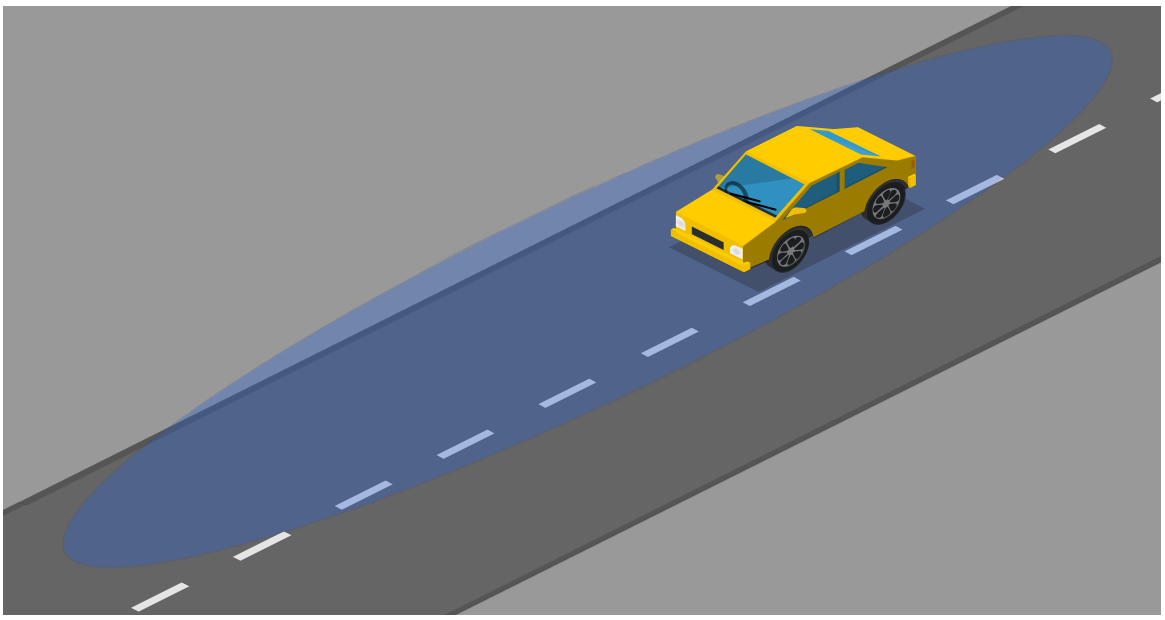

Figure 5: Representation of the elliptic-shaped coverage area surrounding the vehicle

The detection algorithm utilizes formula 8 to check whether the GPS coordinates of surrounding vehicles fall within the coverage area C_a of the detector vehicle. In this formula, (x_c, y_c) represents the center of the ellipse, (x_p, y_p) represents the coordinates of nearby vehicles, and α is the heading of the detector vehicle.

$$\frac{[\cos(\alpha)(x_p - x_c) + \sin(\alpha)(y_p - y_c)]^2}{a^2} + \frac{[\sin(\alpha)(x_p - x_c) - \cos(\alpha)(y_p - y_c)]^2}{b^2} \leq 1 \tag{8}$$

The size of the ellipse is determined by the parameters a and b, representing the *length* (i.e., the maximum distance from the center of the ellipse on the horizontal axis) and the *height* (i.e., the maximum distance from the center of the ellipse on the vertical axis) of the ellipse, respectively.

Occlusion

A critical limitation for both versions of the *Local* detection (*cone* and *ellipse*) arises when a vehicle or an obstacle obstructs the line of sight of the detector's perceptual system. This obstruction creates a blind zone where the detector is unable to perceive any other vehicle, as illustrated in Figure 6. This limitation decreases the perceptual capabilities of both versions of the *Local* detector, especially in scenarios involving sudden braking like the one proposed by the SixPack $v2$ attack. The blind zone has significant implications for V2X communication: if a vehicle located in the blind zone broadcasts its position via BSM or CAM message, the detector vehicle cannot verify that information with its own data obtained by the perceptual system. Consequently, the integrity and reliability of the shared position cannot be confirmed, raising concerns about trustworthiness and potential security vulnerabilities.

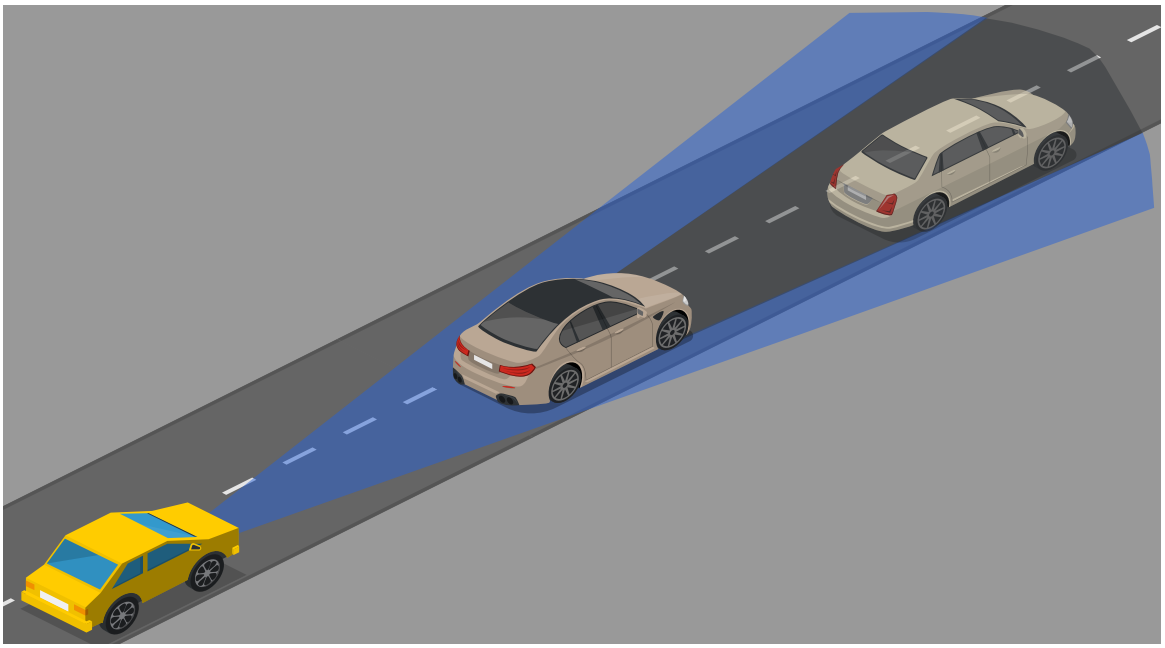

Figure 6: Representation of the shadow area

To better understand this limitation, we introduce a mathematical formulation of the shadow area. Using a polar coordinate system centered on the detector vehicle, the shadow area can be described based on angular and radial boundaries determined by the distance and width of the occluding vehicle. The shadow area is depicted in Figure 7, where the blue color represents the coverage area of the perceptual system. This area is primarily focused on the forward direction but can be applied to either the *ellipse* or *cone* versions of the *Local*

detector. The light blue color indicates the blind zone generated by the occluding entity, which is represented using the black vertical line.

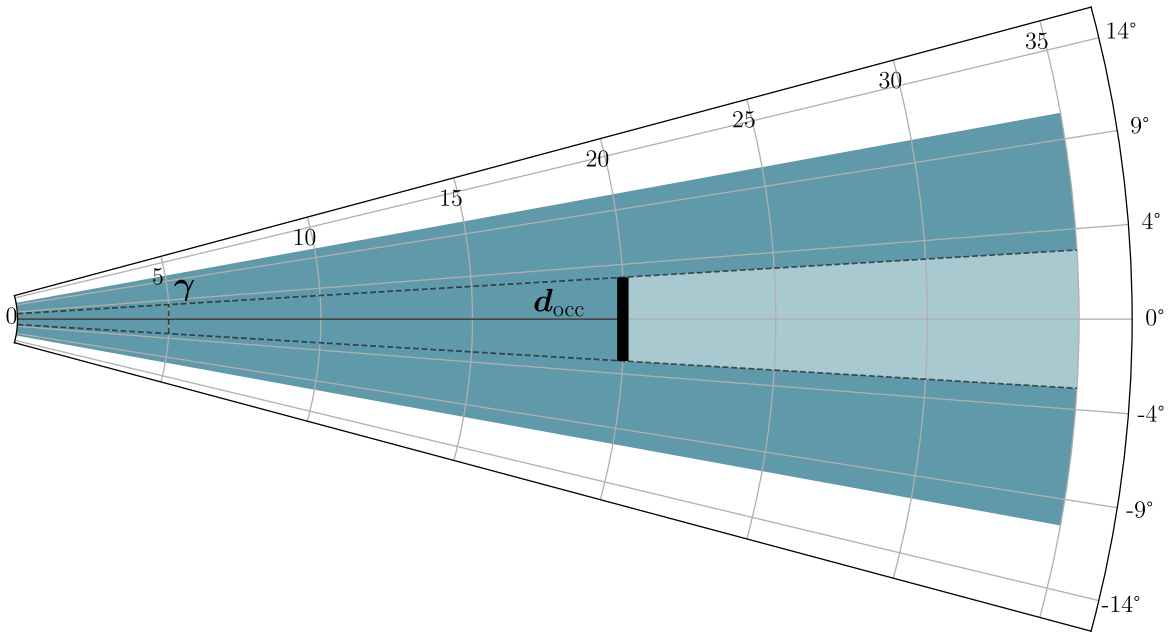

Figure 7: Polar representation of the shadow area

This formalization enables a near approximation of the zone that is inaccessible by the perceptual systems of the vehicles, and of V2X messages whose broadcasted positions cannot be confirmed through the on-board perceptual systems.

Let the detector vehicle be located at the origin $(0, 0)$ of a polar coordinate system (ρ, θ), where ρ is the distance and θ is the angle measured from the polar axis. Based on the geometry depicted in Figure 7, the *shadow area* can be defined with $\gamma \in [\pm\gamma_{shadow}]$ and $\rho \in [d_{\text{occ}}, R]$, where:

- d_{occ} is the radial distance from the detector vehicle to the rear of the occluding entity;
- R is the maximum perceptual range of the detection system in the forward direction;
- $\gamma_{\min}$ and $\gamma_{\max}$ correspond to the angular span of the occlusion based on the width of the obstructing entity.

The shadow area S can therefore be formally described as:

$$S = \{(\rho, \theta) \,|\, d_{\text{occ}} \leq \rho \leq R,\ \gamma_{\min} \leq \theta \leq \gamma_{\max}\} \tag{9}$$

Within this region, any position x_p, y_p received via standard V2X communication (BSM or CAM message) from a surrounding vehicle cannot be verified by the detector vehicle's perceptual system. We define the verification function $V(x_p, y_p)$ as:

$$V(x_p, y_p) = \begin{cases} 0, & \text{if } (x_p, y_p) \in S \\ 1, & \text{if } (x_p, y_p) \in C_a - S \end{cases} \tag{10}$$

Hence, for all $(x_p, y_p) \in S$, the detector vehicle is effectively blind and unable to confirm the consistency of the received V2X information through its own perception systems. The check of the shadow area of Formula 10 needs to be included in the previous validation outlined above for both versions of the vehicle detector. By

incorporating the shadow area verification, the system can more effectively identify whether a V2V message falls within the occluded region, ensuring that only reliable and verifiable positions are used in the detection process.

4 Detection procedure

The detection procedure, which requires data from both C-ITS and perceptual system to identify any potential attack, begins after reception of a V2X message, which can be either BSMs in the DSRC/WAVE standard or CAMs in the ETSI ITS-G5 standard. Both BSMs and CAMs messages include the following three mandatory parameters:

- `position`: the vehicle GPS coordinates;
- `timestamp`: the timestamp associated with the last BSM or CAM message containing position information;
- `pseudonym`: a value used to identify the vehicle.

We emphasize that the pseudonym is included in the mandatory part of V2X communication messages, known as `TemporaryID` in [16] and `authorization tickets` in [11], and they are used to enhance the privacy of communications within C-ITS networks [3]. The detection algorithm discards all messages whose coordinates are outside the coverage range of the perceptual system used by the detector. If the GPS coordinates extracted from the message are within the vehicle's range, the algorithm analyzes the output of the sensor fusion system to identify the vehicle corresponding to the extracted GPS coordinates. The detection algorithm is designed to consider an approximation error due to the sensor fusion procedure. If the location extracted from the message does not match any entity location identified with the perceptual system of the vehicle, then an anomaly is raised. We defined three main phases of the detection algorithm, being the *V2X message evaluation*, *perceptual system representation*, and *anomaly detection* phases.

The flowchart in Figure 8 illustrates the three phases of the detection procedure, outlining the sequence of steps used to validate V2X messages with the data obtained by the detector's perceptual system, following a left-to-right flow. We also want to remark that the process begins after the reception of every V2X message.

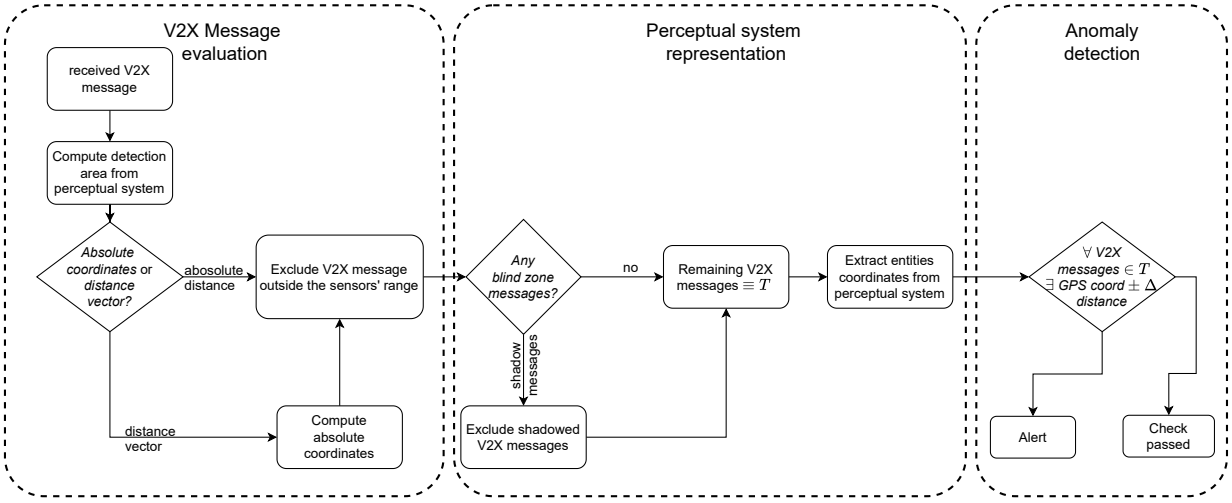

Figure 8: Flowchart of the detection procedure

4.1 V2X message evaluation

As a first step of the detection procedure, the system computes the detection area (C_a) based on the current position of the detector entity and its perceptual system. Then, for each incoming V2X message, the algorithm verifies whether the GPS coordinates (x_p, y_p) contained in the message fall within the entity's coverage area, according to the coverage area *cone* and *ellipse* for *Local* entities.

The *cone* variant converts the GPS position received in the V2X message into *radial* and *polar* (Formula 4 and 6). Then, the algorithm employs the constraints of Formula 7 to evaluate whether the polar value is within the circular sector, and the radial distance is within the maximum range, respectively. The *ellipse* variant method use Formula 8. All the detection areas are calculated using the algorithm 1 based on the type of the detector.

When the position is validated, i.e., it resides within the perception area of the perceptual system used in the different variants, the algorithm proceeds to the next phase. Otherwise, if the coordinates of the message results outside the detection area C_a, the detection procedure ends.

Algorithm 1: Check Detection Area

```
1  Function CHECK_DETECTION_AREA(x_p, y_p)
2      (x_c, y_c) ← DETECTOR.GET_POSITION();
3      d_type ← DETECTOR.GET_TYPE();
4      if d_type is elliptical then
5          α ← GET_VEHICLE_DIRECTION();
6          C_a ← CALCULATE_ELLIPSE_AREA(x_c, y_c, α);
7      else if d_type is conical then
8          α ← GET_VEHICLE_DIRECTION();
9          C_a ← CALCULATE_CONE_AREA(x_c, y_c, α);
10     return (x_p, y_p) is in C_a;
```

4.2 Perceptual system representation

In this phase, the algorithm identifies entities that fall within the entity's coverage area using the formulas previously described. In particular, the detector uses its current position along with the output from its own perceptual system to evaluate the coordinates of surrounding entities. If the positions are expressed using distance vectors, an additional step is performed to calculate the absolute coordinates. The first phases of the detection procedure are stated in the Algorithm 1 where a general position x_p, y_p (either a V2X message or surrounding entity) is evaluated to verify the coordinates are within the coverage area depending on the detection node considered. As an example, a vehicle equipped with a LiDAR can compute (x_p, y_p) of a given object within the LiDAR range by combining its own position (x_c, y_c) with the distance and angles provided by the LiDAR, while an ITL equipped with calibrated cameras can extract positions from the video stream using computer vision approaches [23].

The next step of the detection process (which is mostly related to the *Local* detection) evaluates the position of the surrounding entities to determinate the presence of occlusions and calculate the corresponding blind zone (S).

4.3 Anomaly detection

In the final phase, the algorithm compares the digital representation of entities extracted from the received V2X message (which fall inside the updated coverage are $C_a - S$) with their real representation extracted from the perceptual system of the detector. An anomaly is raised if the digital representation of an entity does not align with any entity seen from the perceptual system with a tolerance of ± 2 meters to account for any sensor's approximation errors. We set this value considering a tolerance higher than the ones presented in [4], where the authors experimentally demonstrate that different automotive perception sensors have an approximation error up to 1 meter. In particular, we increase the approximation error by 1 meter to consider any additional

approximation error of the GPS antenna in the presence of obstruction, which is not evaluated in [4]. We remark that this additional approximation error on the GPS antenna is extremely conservative, and that in the presence of high buildings (i.e., the GPS satellites are not visible at road level) the GPS system could provide a position that falls between 1 and 5 meters [25].

Additionally, we implemented a time-based filter to overcome the limitation of the simulation environment, where the data (i.e., V2X messages and perception systems output) are only available after the simulation is completed. In particular, each detection entity compares the position of each entity extracted by the *perceptual system representation* with the last V2X message received for each entity up to a maximum of T_b, which is the beacon interval of the simulation. In our experiments, T_b is set equal to 1 second. We remark that this threshold value in the simulation environment allows the detection entity to use only the last message sent by the visible C-ITS entities, and that in a real-world application C-ITS entities could have different beacon period, thus requiring an additional filtering step.

The entire detection procedure phases are reported in Algorithm 2 starting from the reception of a V2X message to the last signaling phase.

Algorithm 2: Detection Procedure

Function *Comparison_Checks(msg)*
  $(x_c, y_c) \leftarrow$ EXTRACT_POSITION(msg);
  if *CHECK_DETECTION_AREA*(x_c, y_c) **then**
    $find \leftarrow$ False;
    $entities \leftarrow$ GET_SURROUNDING_ENTITIES();
    for e *in* $entities$ **do**
      $e_{pos} \leftarrow$ GET_POSITION(e);
      if e_{pos} *is a distance vector* **then**
        $e_{pos} \leftarrow$ CALCULATE_COORD(e_{pos});
      $S \leftarrow$ CALCULATE_SHADOW_AREA(v_{pos}, e_{pos});
      if m_{pos} *in* $(C_a - S)$ *and* $m_{pos} \approx e_{pos}$ **then**
        $find \leftarrow$ True;
        BREAK;
    if *not* $find$ **then**
      $sender \leftarrow$ EXTRACT_SENDER(msg);
      RAISE_ANOMALY();
  else
    return // detection procedure ends

5 Performance of the local detection

In this section, we compare the detection performance of both variants of the *Local* detection method (i.e., the *cone* and *ellipse* variants) with the F^2MD framework against the SixPack $v2$ attack. We clarify that F^2MD is used in its intended form: all C-ITS entities (vehicles) run their local misbehavior checks and forward the resulting reports to the Management Authority (MA), which performs the global detection stage. For this reason, although the *Local* detector operates at vehicle level and F^2MD is designed to operates a higher-level (infrastructure-oriented) framework, both detection methods rely on the same C-ITS data produced by participating entities.

In the different test scenarios, we configure the *cone* variant of the detection algorithm with a value of β equal to 20 *degrees* and ρ equal to 35 *meters*, while the *ellipse* variant is configured with a value of a equal to 35 and b equal to 3 meters. We remark that all the values used to model the perceptual systems in both *Local* and detection methods are compliant with the current specifications of sensors commercially available on the market [14, 4].

In every simulation, each vehicle is configured to be either *attacker*, *detector*, or *neutral*. The percentage of vehicles belonging to the *attacker* category is fixed throughout all the simulation scenarios (5% of the total number of vehicles).

This attacker probability is set to replicate a realistic scenario where malicious vehicles are the minority, ensuring that the detection system is challenged without overwhelming the normal traffic flow. We remark also that such low attacker percentage represents a worst-case scenario for the detection system, since a higher number of attackers would improve the *recall* of the detectors, thus leading to higher detection performance. The main goal of these experiments is to provide a meaningful understanding of the capabilities of the detection system against a stealthy attack that is conducted by a minority of the vehicles. Hence, we focus the results on the comparison between different percentages of *detector* nodes with a fixed number of *attacker* vehicles.

The number of vehicles in the *detector* category is a test parameter on which the performance evaluation is based. All vehicles in the *detector* category are configured to evaluate both our *Local* implementations and F^2MD checks simultaneously. Due to the nature of the SixPack $v2$ attack, which targets vehicles trailing the one used to launch the attack, vehicles belonging to the *attacker* category are configured to only launch an attack when at least one vehicle is trailing. This configuration choice is required to ensure that all instances of SixPack $v2$ have a valid target and to avoid activation of the attack without consequences for other vehicles. We also want to remark that, to create a more challenging and realistic testing scenario, the target vehicles are chosen independently of all the vehicle categories. This means that, if the target vehicle belongs to the *neutral* or *attacker* category the attack could not be detected.

Performance of the F^2MD and the *Local* detectors is compared by considering a variable percentage of vehicles in the *detector* category, from 10% to 100% of the vehicles not already selected as *attackers*. For each detection configuration, we replicate the same simulation scenario 100 times and used a boxplot representation to highlight variance in the detection results.

The detection results of the F^2MD framework and the *cone* and *ellipse* variants of the *Local* detection method are summarized in three plots, one dedicated to the *precision* metric (Figure 9), one for the *recall* metric (Figure 10), and one for the $\mathcal{F}_1$ metric (Figure 11). The x-axis of each figure contains the percentage of vehicles in the *detection* category (with the corresponding number of vehicles in round brackets) used by the detectors, while the y-axis shows the scale of the performance index. In each of these plots, results of F^2MD are red, while results of the *Local* methods are green (*cone*) and blue (*ellipse*).

Focusing on the precision metric (Figure 9) it is possible to notice that our approach outperforms the F^2MD detection framework in both variants. In particular, the *precision* metric of the *Local* detection methodology reaches the maximum value of 1.0 in both detection configurations with a standard deviation close to 0, implying that both approaches of the *Local* detection method are extremely resilient to false positives. On the other

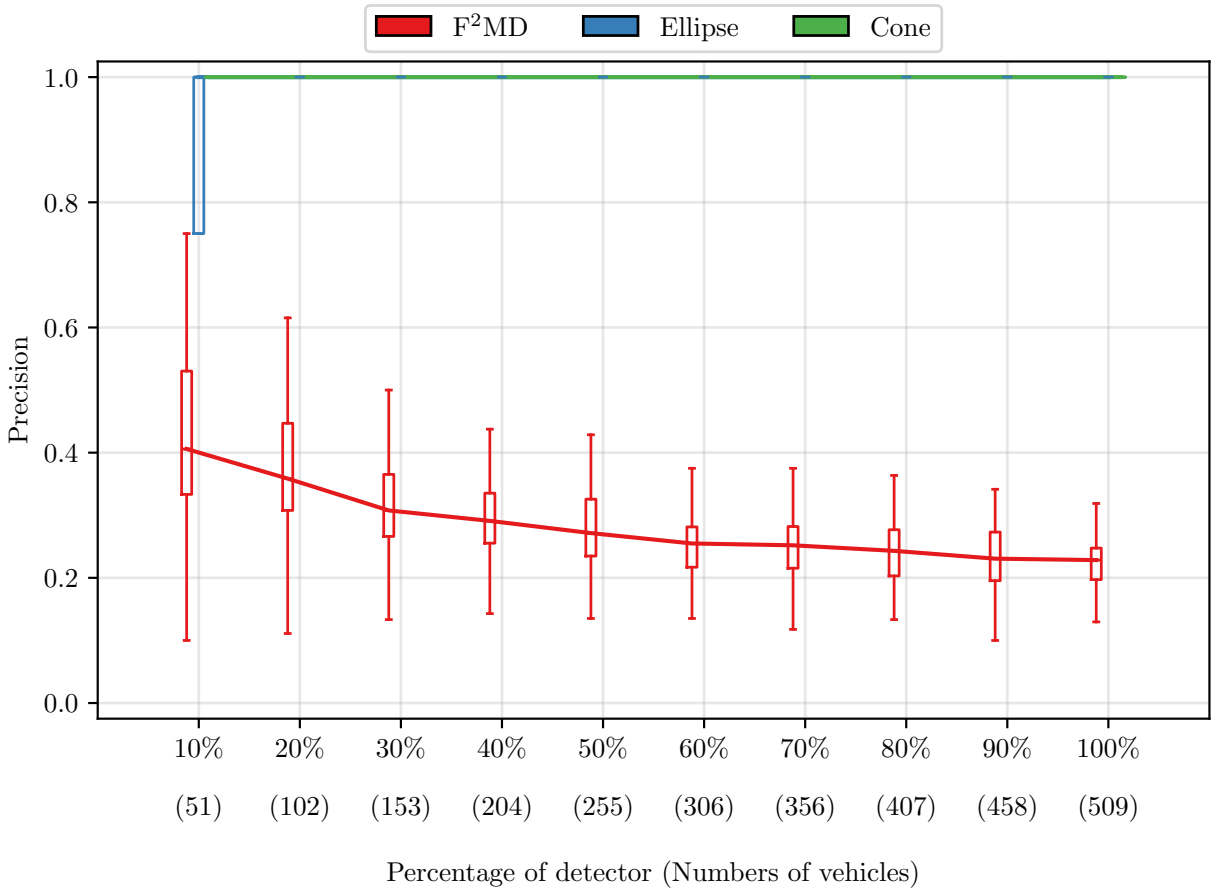

Figure 9: Comparison of the *precision* index of F^2MD and the *Local* detection methods against the SixPack $v2$ attack.

hand, the median *precision* value of the F^2MD framework has a decreasing trend inversely proportional to the percentage of vehicles contributing to the detection process, starting with a value slightly higher than 0.4 with 10% of detectors down to a value close to 0.25 with 100% of vehicles participating in the detection process. The decrease in both the median and variance of the precision for F^2MD occurs because, as more vehicles contribute reports, the MA aggregates a larger amount of partially redundant information, increasing the likelihood of false positives. This results in more consistent (lower variance) but slightly less precise (lower median) outcomes. When only a few vehicles participate, the system relies on a smaller and more heterogeneous set of reports, leading to greater variability and occasional high-precision results when the contributing vehicles have favorable coverage.

This high false positive rate is a well known issue for all intrusion detection algorithms, often preventing their applicability in real contexts [2].

Results focused on the *recall* of the *Local* detectors and F^2MD are presented in Figure 10, where it is possible to notice that all three detectors are able to achieve high *recall* values with a higher number of vehicles participating in the detection task. The F^2MD framework achieves higher *recall* values than both variations of the *Local* detection method, with the *cone* implementation being more effective in the identification of an ongoing attack than the *ellipse* variation. However, we remark that all detection methods exhibit a similar trend and very similar *recall* performance, reaching values higher than 0.6 with 100% of detection vehicles.

The F^2MD achieves a slightly higher *recall* performance compared to both versions of *Local* detection since

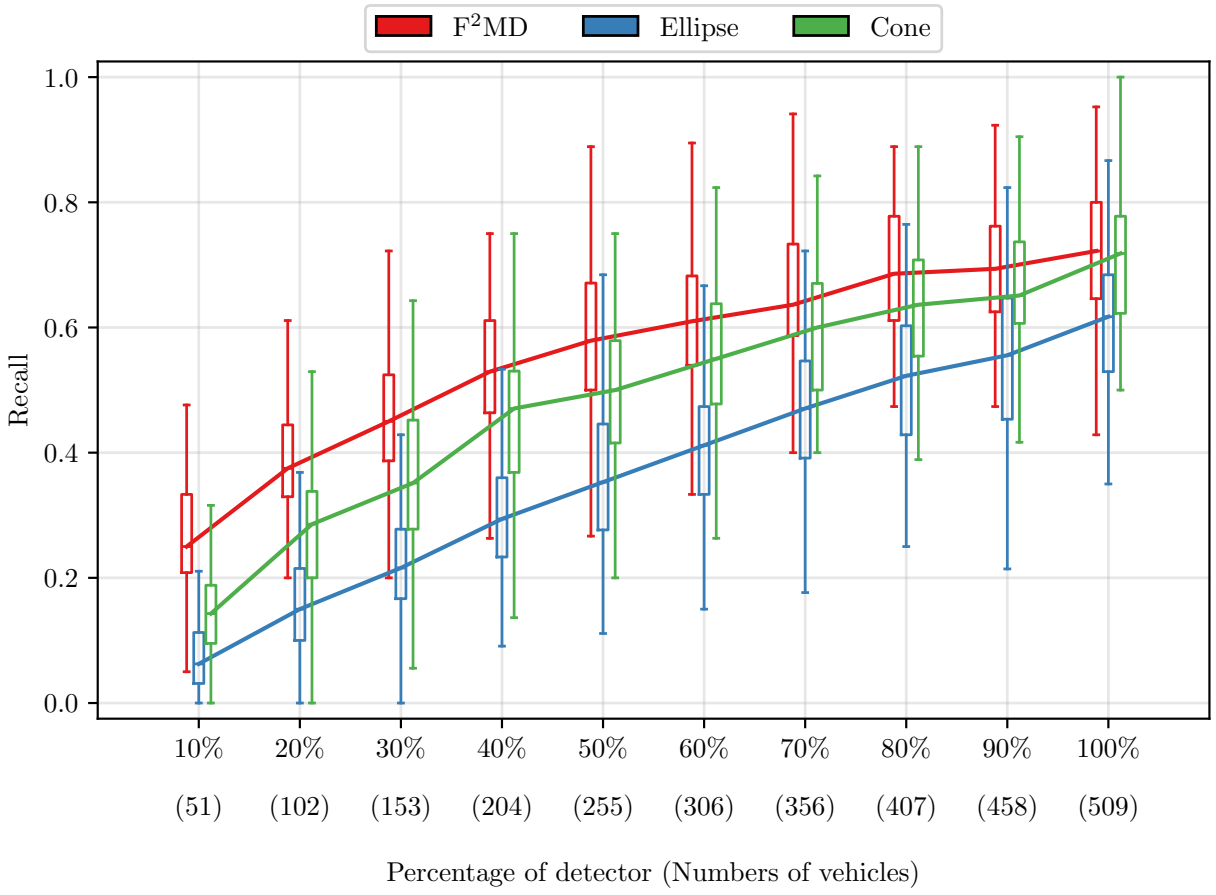

Figure 10: Comparison of the *recall* index of F^2MD and the *Local* detection methods against the SixPack $v2$ attack.

it relies only on the analysis of the content of the messages, allowing all vehicles receiving malicious messages to contribute in the detection process, ensuring a wider range. In contrast, *Local* detection only analyzes messages which are within its own sensing area be it *cone* or *ellipse* shape, as defined in Section 3.1.

A limitation of the *Local* detection arises from the presence of shadowed areas in the *perceptual system representation* phase. When an attacking vehicle is occluded by another *neutral* vehicle positioned between the attacker and the detector, the detector's sensing model prevents it from observing the attacker's reported position. As a consequence, the corresponding V2X messages cannot be evaluated in the detection process, leading to an increased number of false negatives. This effect directly contributes to the lower *recall* observed in Figure 10 and reflects the inherent constraints of on-board sensing systems when line-of-sight is obstructed.

Finally, the $\mathcal{F}_1$ of the *Local* detectors and F^2MD are summarized in Figure 11, showcasing that both variations of our novel detection method outperforms F^2MD, which has a median $\mathcal{F}_1$ value always lower than 0.4 without regards to the number of detectors. On the other hand, the $\mathcal{F}_1$ achieved with both *Local* detection variations is higher than the one achieved by F^2MD when at least 20% of the vehicles are participating in the detection task, with an overall detection maximum of 0.9142 with the *cone* variant (with 100% of the vehicles participating in the detection task). We remark that the poor detection performance of F^2MD (and the good detection performance of the *Local* variations) are mainly related to the poor (high for *Local*) *precision* presented in Figure 9 of the detector, with the *Local* variations being able to achieve high detection results without raising a single false positive.

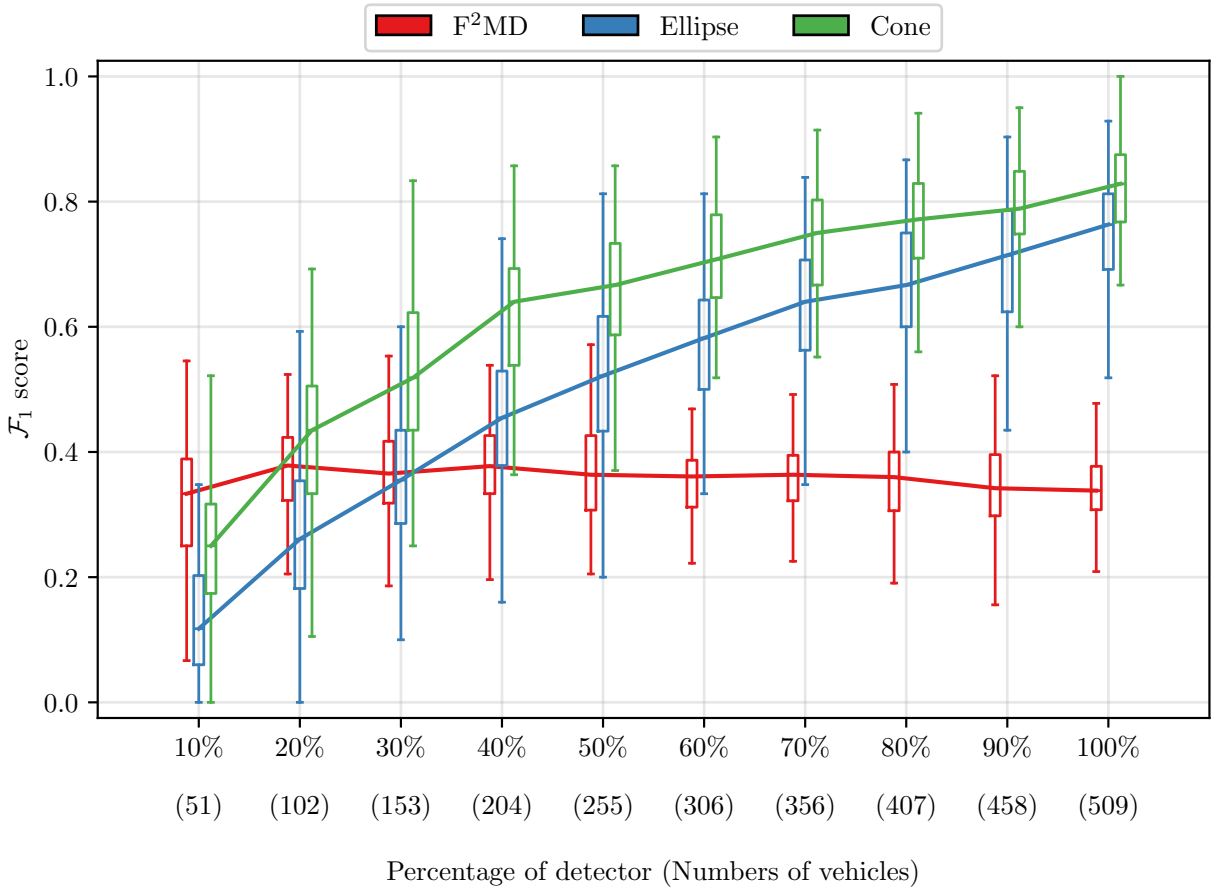

Figure 11: Comparison of the $\mathcal{F}_1$ index of F^2MD and the *Local* detection methods against the SixPack $v2$ attack.

6 Conclusions

In conclusion, this deliverable introduced a local misbehavior detection approach that leverages on the comparison between the data gathered from the perception system of the vehicle to assess the consistency of received V2X messages with the physical environment. The proposed method formalizes the notion of verifiable space through two detector configurations: conical and elliptical model thereby adapting the verification process to heterogeneous sensing capabilities. In addition, the algorithm explicitly accounts for occlusions by defining a shadow region that is removed from the verification set, which limits false accusations in situations where perception cannot provide reliable confirmation. Building on these definitions, the detection procedure combines (i) message parsing and localization of claimed positions, (ii) extraction and normalization of perceived entities (including transformation from relative to absolute coordinates when required), and (iii) a spatial association step that flags a message as suspicious when its claimed position cannot be matched to any perceived vehicle within a chosen tolerance. Overall, this work have two main contributions: demonstrates how misbehavior detection based on the analysis of the V2X messages are not sufficient for detecting stealthy attacks such as the SixPack attack. The second part perception-grounded checks can complement communication-only plausibility mechanisms and improve robustness against stealthy, behavior-mimicking attacks. Future work should focus on quantitative evaluation under realistic sensor noise and localization uncertainty, principled selection of matching thresholds, and extensions that handle dense traffic and multi-object association more systematically, as well as cooperative fusion across multiple detectors to increase coverage and reduce blind spots.

References

[1] Ahmad Abuashour and Michel Kadoch. Vehicular ad-hoc networks: Architecture, applications and challenges. *International Journal of Computer Science & Network Security*, 20(2):26–36, 2020.

[2] Stefan Axelsson. The base-rate fallacy and the difficulty of intrusion detection. *ACM Trans. Inf. Syst. Secur.*, 3(3):186–205, aug 2000. doi:10.1145/357830.357849.

[3] Messaoud Babaghayou, Nabila Labraoui, Ado Adamou Abba Ari, Nasreddine Lagraa, and Mohamed Amine Ferrag. Pseudonym change-based privacy-preserving schemes in vehicular ad-hoc networks: A survey. *Journal of Information Security and Applications*, 2020.

[4] Minjin Baek, Donggi Jeong, Dongho Choi, and Sangsun Lee. Vehicle trajectory prediction and collision warning via fusion of multisensors and wireless vehicular communications. *Sensors*, 20(1):288, 2020.

[5] Faran Awais Butt, Jawwad Nasar Chattha, Jameel Ahmad, Muhammad Umer Zia, Muhammad Rizwan, and Ijaz Haider Naqvi. On the integration of enabling wireless technologies and sensor fusion for next-generation connected and autonomous vehicles. *IEEE Access*, 10:14643–14668, 2022.

[6] Claudia Campolo, Antonella Molinaro, and Riccardo Scopigno. *Vehicular ad hoc networks: standards, solutions, and research*. Springer, 2015.

[7] Anirut Choosakun, Yaowapa Chaiittipornwong, and Chaiyapat Yeom. Development of the cooperative intelligent transport system in thailand: A prospective approach. *Infrastructures*, 6(3):36, 2021.

[8] ETSI. Intelligent Transport Systems (ITS); Security; Security header and certificate formats; Release 2. Technical Specification (TS) ETSI TS 103 097 V2.1.1, European Telecommunications Standards Institute, Sophia Antipolis Cedex, France, October 2021. URL: https://www.etsi.org/deliver/etsi_ts/103000_103099/103097/02.01.01_60/ts_103097v020101p.pdf.

[9] ETSI. Intelligent Transport Systems (ITS); Security; Trust and Privacy Management; Release 2. Technical Specification (TS) ETSI TS 102 941 V2.2.1, European Telecommunications Standards Institute, November 2022. URL: https://www.etsi.org/deliver/etsi_ts/102900_102999/102941/02.02.01_60/ts_102941v020201p.pdf.

[10] ETSI. Intelligent Transport Systems (ITS); Security; Security management messages communication requirements and distribution protocols; Release 2. Technical Specification (TS) ETSI TS 103 601 V2.1.1, European Telecommunications Standards Institute, Sophia Antipolis Cedex, France, March 2024. URL: https://www.etsi.org/deliver/etsi_ts/103600_103699/103601/02.01.01_60/ts_103601v020101p.pdf.

[11] European Telecommunications Standards Institute (ETSI). Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management. Technical Report TS 102 940 V2.1.1, ETSI, 2021. Accessed: July 2025. URL: https://www.etsi.org/deliver/etsi_ts/102900_102999/102940/02.01.01_60/ts_102940v020101p.pdf.

[12] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

[13] Sohan Gyawali, Shengjie Xu, Yi Qian, and Rose Qingyang Hu. Challenges and solutions for cellular based v2x communications. *IEEE Communications Surveys & Tutorials*, 23(1):222–255, 2021.

[14] Henry Alexander Ignatious, Manzoor Khan, et al. An overview of sensors in autonomous vehicles. *Procedia Computer Science*, 198:736–741, 2022.

[15] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. Ben-Jemaa, and P. Urien. Simulation framework for misbehavior detection in vehicular networks. *IEEE Transactions on Vehicular Technology*, 2020.

[16] John B. Kenney. Dedicated short-range communications (dsrc) standards in the united states. *Proceedings of the IEEE*, 99(7):1162–1182, 2011. doi:10.1109/JPROC.2011.2132790.

[17] Salabat Khan, Fei Luo, Zijian Zhang, Mussadiq Abdul Rahim, Mubashir Ahmad, and Kaishun Wu. Survey on issues and recent advances in vehicular public-key infrastructure (vpki). *IEEE Communications Surveys & Tutorials*, 24(3):1574–1601, 2022.

[18] Mengyue Li and Hanying Guo. Overview of c-its deployment projects in europe and usa. *arXiv preprint arXiv:2010.07299*, 2020.

[19] C. Longari et al. Goliath: A scalable framework for misbehavior detection in vehicular networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(6):2670–2681, 2020. doi:10.1109/TITS.2020.3023063.

[20] Enrique Marti, Miguel Angel De Miguel, Fernando Garcia, and Joshue Perez. A review of sensor technologies for perception in automated driving. *IEEE Intelligent Transportation Systems Magazine*, 11(4):94–108, 2019.

[21] Jonathan Petit et al. Cooperative its security standards: Implementation, assessment and next challenges. *IEEE Communications Surveys & Tutorials*, 2020. Discusses the alignment of ETSI TS 103 097 with IEEE 1609.2 to achieve global cross-compatibility.

[22] Miguel Sepulcre, Javier Gozalvez, and Gokulnath Thandavarayan. On the potential of v2x message compression for vehicular networks. *IEEE Access*, 8:20826–20842, 2020.

[23] Haryong Song, Wonsub Choi, and Haedong Kim. Robust vision-based relative-localization approach using an rgb-depth camera and lidar sensor fusion. *IEEE Transactions on Industrial Electronics*, 63(6):3725–3736, 2016.

[24] Andre Weimerskirch. V2x security & privacy: the current state and its future. In *ITS World Congress, Orlando, FL*, page 21, 2011.

[25] Duojie Weng, Baoguo Yu, Jingbo Zhao, Shuo Li, Hangyu Zhou, and Ying Xu. Augmenting vehicle gnss positioning through cross-street measurements in urban canyons. *Measurement*, 257:118603, 2026. URL: https://www.sciencedirect.com/science/article/pii/S0263224125019621, doi:10.1016/j.measurement.2025.118603.

[26] X. Xu, Y. Wang, and P. Wang. Comprehensive review on misbehavior detection for vehicular ad hoc networks. *Journal of Advanced Transportation*, 2022:1–27, 2022. doi:10.1155/2022/4725805.

[27] Pengtao Yang, Dongliang Duan, Chen Chen, Xiang Cheng, and Liuqing Yang. Multi-sensor multi-vehicle (msmv) localization and mobility tracking for autonomous driving. *IEEE Transactions on Vehicular Technology*, 69(12):14355–14364, 2020.

[28] Y. Zhang et al. Security in vehicular ad hoc networks: Challenges and solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(9):2549–2561, 2017. doi:10.1109/TITS.2017.2685299.

[29] Giovanni Gambigliani Zoccoli, Francesco Pollicino, Dario Stabili, and Mirco Marchetti. Sixpack v2: enhancing sixpack to avoid last generation misbehavior detectors in vanets. In *2022 IEEE 21st International Symposium on Network Computing and Applications (NCA)*, volume 21, pages 243–249. IEEE, 2022.