Trust Management Framework for Misbehavior Detection in Collective Perception Services

Jiahao Zhang, Ines Ben-Jemaa, Fawzi Nashashibi

Trust Management Framework for Misbehavior Detection in Collective Perception Services

Jiahao Zhang*, Ines Ben Jemaa*, and Fawzi Nashashibi†
*IRT SystemX, †INRIA – Team ASTRA, Paris, France
Email: {jiahao.zhang, ines.ben-jemaa}@irt-systemx.fr, fawzi.nashashibi@inria.fr

***Abstract*— Collective Perception Messages (CPM) enable vehicles to share their perceived objects with their neighbors in V2X network. These perception data extend local vehicles' perception and consequently improve road safety awareness. However, attacks on perception data are challenging and require advanced and efficient misbehavior detection mechanism especially in specific road scenarios where contradictory information need to be analysed. In this work, we introduce a trust management framework to detect misbehaving nodes through transmitted CPM messages. Our framework is based on trust assessment built through several processing steps. It addresses conflict situation when contradictory data are received using the Subjective Logic mechanism. The results show that our solution is effective in detecting misbehaving nodes based on their attributed trust scores. In addition, we show the impact of our solution and some CPM configuration parameters on safety services and especially on risk anticipation in intersection scenarios.**

I. Introduction

Autonomous vehicles are equipped with several sensors that enable the detection of dynamic objects on the road. However, sensor perception of the surrounding objects is limited to their Field Of View (FOV) and blocked by the occluding obstacles. Collective Perception (CP) services are designed to enable vehicles to extend their local perception of the environment with additional perception data. These data are sent by their neighbors through Collective Perception Messages (CPM) [1]. In this service, neighboring vehicles or Road-Side-Units (RSU) broadcast the list of their local sensors' perceived objects in the V2X (Vehicle-to-Everything) network. The receiving vehicles merge these data with their local perceived objects to elaborate an extended view of the environment. The generalized architecture proposed in [2] is one of the first proposed architectures where V2X exchanged messages are considered as an additional virtual sensor along with the local embedded sensors. The global perception information is then used by high level ITS services such as collision warning, obstacle avoidance and others, to take appropriate driving decisions.

However, such cooperative system leaves the opportunity for an attacker to broadcast erroneous perception information in his neighborhood intentionally. It is also possible that a perception device module is faulty and thus provides false perception information. In general, to enable ITS services to function correctly, it is mandatory to validate and to verify the consistency and the trustworthiness of each received CPM data when it is fused with the local perception and with other received V2X messages data. These operations are part of the misbehavior detection process [3].

So far, several works exist on Misbehavior detection in the literature [4] [5]. However, most of them are mainly addressing manipulation attacks of the kinematic senders' data. Such attacks consist on, for a given ego vehicle, modifying its own kinematic information transmitted in the beacon information (*e.g.* through the Cooperative Awareness Messages, CAM). Misbehavior attacks on CPM data, on the other hand, consist essentially on modifying the perceived scene data by manipulating the perceived objects kinematic characteristics. Smart attackers are even able to report continuously consistent and correct perceived scene but which may lead safety application to take inappropriate decisions and then create severe safety damages. These new attacks impose additional detection challenges. For instance, a typical complex situation occurs when an ego vehicle receives two contradictory statements on one or several perceived objects from its neighbors. To handle this conflict situation, the ego needs to detect accurately the misbehaving sender and to react appropriately to that. In this paper, we propose and implement a misbehavior assessment framework based on an enhanced trust management model at several levels of the misbehavior detection process. Our framework covers a large set of collective perception attacks ranging from basic attacks to advanced attacks leading to conflict situations. Our trust management model takes into consideration specific communication parameters and realistic perception scenarios. It integrates at a certain level the subjective logic mechanism which allows the fusion of several incoming data from the neighborhood when a conflict situation is detected.

The paper is organized as follows: section II presents the related work to the addressed topic. Section III details our proposed framework which is experimented and evaluated in section V. Finally, section VI concludes the paper and gives some perspectives.

II. Related Work

Misbehavior detection in cooperative vehicular systems captures a lot of attention in the past few years. Giving the ephemeral aspects of the communication links in such systems, existing solutions are mainly based on data-centric techniques [3]. The main idea is to analyse the transmitter's kinematic data to verify their plausibility and consistency [5]. Some methods use either direct verification or probabilistic techniques while other methods deploy machine learning

techniques [6] [7]. Even if they show encouraging detection results, the focus of these work was limited to data manipulation attacks on beacon messages (such as CAM messages).

Recently, some works addressed the misbehavior attacks on collective perception data. [8] proposes a generic framework for misbehaviour detection using collective perception. Their framework is based on several levels of detection. The work elaborates a useful generic view giving details on how to proceed with data verification for collective perception but does not give a concrete solution nor validation results of the proposed framework.

There is another category of works which is based on assessing trust based on the reported neighbors messages. [9] is one of the first works that propose to attribute trust to data rather than nodes in the context of ephemeral networks such as the vehicular network. They propose a generic framework template where they attribute trust for each individual piece of information called *evidence* reporting a specific event at different phases. The ultimate trust establishment phase is based on different data fusion techniques. The authors show that using the Dempster Shafer theory in the fusion process works well when there is a high uncertainty about the event while the Bayesian inference performs best when the a prior knowledge are provided. The author of [10] propose a misbehavior detection approach based on Bayesian filter to estimate the trust of the received collective perception data. They mainly address the manipulation of perceived object attacks and show that their solution is effective for these attacks. However their approach does not take into consideration the particularities of the perception environment which is quite dynamic.

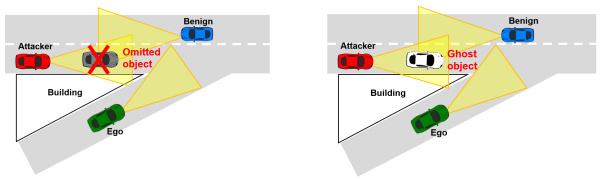
The work in [11] from the multi-sensor fusion field addresses a similar issue which is the detection of faulty perception sensor in a distributed sensor network to support automated driving. The proposed model takes into consideration realistic perception situations and the fusion process is based on the Dempster Shafer (DS) belief theory. Even if the DS shows promising results in fingerprinting the faulty behavior, it have been proven that it may produce counter-intuitive results in conflict situations.

Subjective logic (SL) [12] [13] is another data fusion approach which provides a formalism to represent opinions as belief, disbelief, and uncertainty. Some previous works use SL for Misbehavior detection in V2X systems [14] [15] and show promising results. Our proposed solution is inspired by the generic framework of [9] and is implemented in the context of collective perception. From the different trust assessment phases which consider realistic perception situations, our approach integrates the SL in the final step when a certain level of conflict is reached.

III. Misbehavior Detection for Collective Perception Services

A. Attacker Model

We consider an attacker which is part of the cooperative vehicular system. The attacker is able to send and to receive the V2X messages. This is because he is authenticated and the integrity of his V2X messages is validated by the receivers. The attacker is able to change intentionally the data contained in the CPM message. Specifically, he may change the values of the attributes of the perceived objects continuously. For instance, as shown in Fig.1(a), the attacker



(a) Omission attacks (b) Ghost attacks

Fig. 1: Examples of advanced attacks on cooperative perception

may omit the transmission of a perceived vehicle in an intersection. The perception of the ego vehicle, that is about to perform an insertion maneuver, is limited due to the existence of obstacles such as buildings. This scenario is particularly dangerous especially if the omitted vehicle is not equipped with communication capabilities and may generate a high risk of accident. Fig.1(a) illustrates an other attack scenario. The attacker may create a ghost object on the intersection and sends it in the CPM message. Consequently, the attacker will gain the priority even if we suppose that the ego vehicle has the highest priority to perform the insertion maneuver.

B. Problem Statement and contributions

In the previous section, we illustrate two examples of *advanced* attacks on collective perception services. Compared to attacks where data are not plausible or are not consistent in several consecutive CPMs, these attacks are more challenging to detect. First, the attacker sends periodically consistent information in the consecutive CPMs. Second, the ego vehicle has a limited FoV and thus is not able to verify the correctness of the CPM sent information (*e.g.* either the omitted object or the ghost object in Fig.1. Third, the scenario may generate a word against word situation where the attacker sends a false confirmation about an omitted (or a ghost) object, whereas a benign vehicle (the blue vehicle in Fig.1) sends a totally opposite statement. In addition to all these challenges, it is also known that road perception environments are highly changing and that sensors information are potentially uncertain.

Our contribution will address all the already presented challenges. We propose a misbehavior detection framework based on trust assessment that

1) detects basic as well as advanced collective perception attacks.
2) considers communication and perception environment particularities to assign trust scores.
3) addresses the conflict situation where an ago receives contradictory CPM data.

IV. Trust Assessment Approach

A. Preliminaries

We denote each perceived environmental entity with an identity (*e.g.* vehicles, RSUs, pedestrians, and etc). All the existing nodes can be sources or objects. The source nodes can communicate with each others (*i.e.*, send and receive V2X data). The object nodes which are the environment entities perceived and reported by a source are not equipped with V2X capabilities. In the following, we present the classes of nodes that we assume in our approach. The classes depend on two criteria, the perception of a node and its validity.

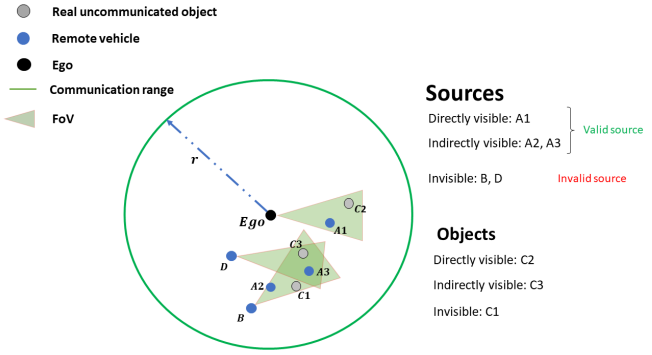


Fig. 2: Node types

1) Source Node Perception:

- **Directly visible**, it means that ego can perceive the node directly thanks to its onboard sensors. It is the case of the source node A1 or the object node C2 in Fig.2.
- **Indirectly visible**, the node cannot be perceived directly by ego, but its existence can be proved by more than one source (*i.e.* at least two neighbors reporting in their CPM that they perceive the node or at least the node itself sends a CAM and another neighbor reports its existence through a CPM). In Fig.2, this is the case of node A2 reported by node B through a CPM and by itself through a CAM and A3 reported by both node B and D through CPM.
- **Invisible**, the node cannot be perceived directly by ego and no one can prove its existence. This is the case of node C1 which is only reported by a CPM sent by node B as illustrated in Fig.2.

2) Source Node Validity:

- **Valid source**, we assume that the directly visible source and the indirectly visible source are valid sources that pass through the process of misbehavior detection with success and there is no detected conflict in its perception data with both the ego's data and the neighbors data.
- **Non valid source** are sources that are either invisible (both directly and indirectly) to the ego or that are detected as misbehaving or there is a possible conflict in their data.

B. Trust assessment Framework

We designed a trust assessment framework based on four phases as illustrated in Fig.3. This trust assessment framework combining with the misbehavior detection part and the decision logic part can help us detect majority of naive attacks (*e.g.* kinematic data modification) and determine the misbehaving node from a controversial event.

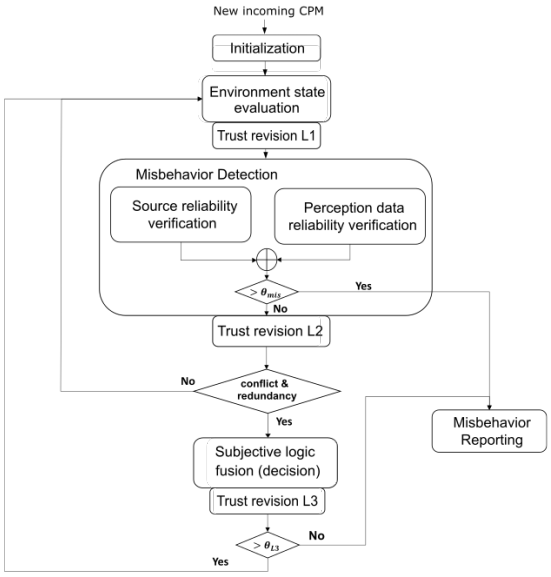


Fig. 3: General Framework

1) Initialization: The initialization phase assigns an initial trust value to all the sources. For instance, we may consider the source type (i,e. vehicle, RSU) or the source permissions (i,e. ordinary vehicle, police vehicle, etc). We assign the default trustworthiness for different source types at different levels. When ego receives the first V2X message from a new source, ego assigns a trustworthiness value to this new source depend on the source type. For instance, ego assigns a higher value for RSUs than to ordinary vehicles. This is because we assume that RSUs are more protected than usual OBUs (On Bord Units) embedded in cars.

2) Trust revision L1: Status Evaluation: In a trust system, the establishment of trustworthiness should consider the environmental factors. The communication lifetime with the ego is an attribute of particular importance. It is reasonable to consider that the node which has a longer connection period with the ego (*i.e.* neighborhood period) is assigned a higher trust value. In addition, when the ego perceives a source directly with its own sensors, it would assigns it a higher trust value. As shown in Fig.4, First, when ego receives a new V2X message from a neighboring source, ego should check if it has received a V2X message from this source in the past ($N > 1$, N is the number of received messages from the same source) and if this source is a valid source (as defined in Section IV-A.2). Then, ego should check if this source is within its local perception range. If ego perceives directly this source, ego should attribute an offset of $+\delta$ to the trust

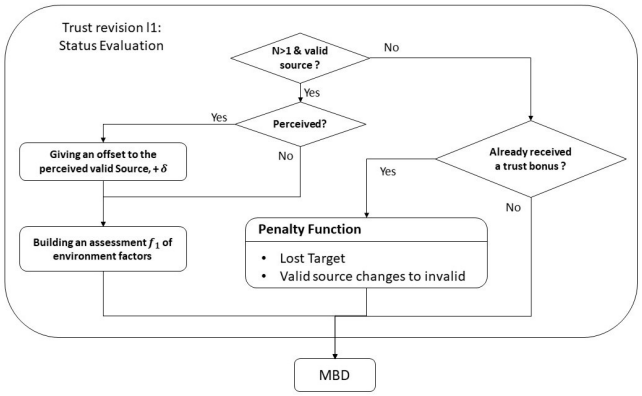


Fig. 4: Trust revision L1: Environment Assessment Module

score of the source. Next, ego considers the communication lifetime with the source as an environment state factor to compute the trust score. The environment state evaluation function f_1 is defined as follows:

$$f_1 = \frac{\alpha_1}{1 + e^{-\beta_1 n + \gamma_1}}, f_1 \in [0, \Delta_{max}] \tag{1}$$

We think that environment factors should not give a high weight to the global trust evaluation. The trust bonus is limited to $\Delta_{max} = \frac{1}{\alpha_1}$, n represents the number of received messages from the source, which gives an indication of the neighborhood communication lifetime. The parameters α_1,β_1,γ_1 are positive calibration parameters. We choose $\alpha_1 = 0.1$, $\beta_1 = 1$, $\gamma_1 = 8$. We limit the maximum trust bonus Δ_{max} related to the communication lifetime at 0.1. We assume this maximum value would not impact considerably the global trust value. In our case, ego starts to increase the trustworthiness of a valid transmitting source after 2 seconds since the target changed its own state from invalid to valid. Then, after receiving 12 seconds, the trust bonus of lifetime achieves its maximum.

Second, if the new V2X message (CPM) is from an invalid source and ego has already attributed the trust bonus to the local perception or to the environment factors to this source, we employ the penalty functions to decrease the already attributed trust bonus. We design two penalty functions as follows, that considers two situations.

- **Lost target**: when ego looses a source target which has been tracked by its perception module for a certain time, ego should reduce the trust until removing the whole given offset $+\delta$. The loss of trustworthiness over time is computed through the f_2 function as follows:

$$f_2 = -\alpha_2 n_L + \beta_2, f_2 \in [-\delta, 0] \tag{2}$$

  n_L is the number of received messages from the source since it can not be perceived by ego. δ, α_2, β_2 are positive calibration parameters. To be sure that the perceived object has left the FoV of the ego and that it is not coasting in the FoV (I,e: leaving and re-entering the FoV), we choose a function defined by parts to compute the trustworthiness T at a given time t as presented in *Eq.*3

$$\begin{cases} T_0, n_L \leq \dfrac{\beta_2}{\alpha_2} \\ T_{t+1} = T_t - |f_2(n_{L(t)}) - f_2(n_{L(t-1)})|, \dfrac{\beta_2}{\alpha_2} < n_L \leq \dfrac{\delta + \beta_2}{\alpha_2} \\ T_0 - \delta, n_L > \dfrac{\delta + \beta_2}{\alpha_2} \end{cases} \tag{3}$$

  In this work, we choose $\delta = 0.2$, $\alpha_2 = 0.04$, $\beta_2 = 0.08$. We assume that when the source object is lost for more than 2 seconds its trustworthiness starts to decrease. If the source object leaves the FoV of the ego for more than 7 seconds, the maximum trust discount is attributed.
- **Valid source changes to invalid**: In a dynamic environment, the nodes sometimes change their state (*e.g.* valid sources can change to invalid and vice versa). When a valid (considered as reliable) source changes to invalid (considered as unreliable), ego should update the created trust relationship. This penalty function is a symmetric function of *Eq.*1 to the x axis, as follows:

$$f_3 = \frac{-\alpha_3}{1 + e^{-\beta_3 n_p + \gamma_3}}, f_3 \in [-\Delta_{max}, 0] \tag{4}$$

  The parameters α_3,β_3,γ_3 are positive calibration parameters, where $\alpha_3 = \alpha_1$, $\beta_3 = \beta_1$, $\gamma_3 = \gamma_1$. n_p represents the number of received messages from the source since it changed its valid status to invalid.

3) Trust revision L2: Misbehavior detection: The second important phase of our framework is the misbehavior detection performed on the CPM data. These attacks could be kinematic data modification, ghost injection, object omission etc. This phase allows to detect the majority of naive attacks via plausibility and consistency checks. Plausibility checks verifies if an attribute value (e.g., the velocity of the perceived object) is within an acceptable interval. The consistency checks verify if two attributes are consistent or if a received attribute value is consistent with the previous ones. We use the same misbehavior detectors specified in [4]. In this paper, we are more focused on addressing advanced attacks which create conflict situations. The misbehavior detection step consists of two verification levels:

– *Source reliability verification*

Ego should ensure that the neighbor sources are reliable. The CPM has to provide plausible and consistent kinematic data about the neighbor itself, as shown in Fig.5.

- For invisible source, ego can employ some basic detectors to verify the received data. We use data plausibility checks and data consistency checks (such as checking the consistency of the current data with the previous received data from the same neighbor source) etc.
- For directly visible source and indirectly visible source, ego can employ other advanced detectors. Ego can compare the received data with its local perception data or the received perception data from other sources.

– *Perception data reliability verification*

Ego should check the plausibility of the perception data provided by the neighboring sources. Ego should also check

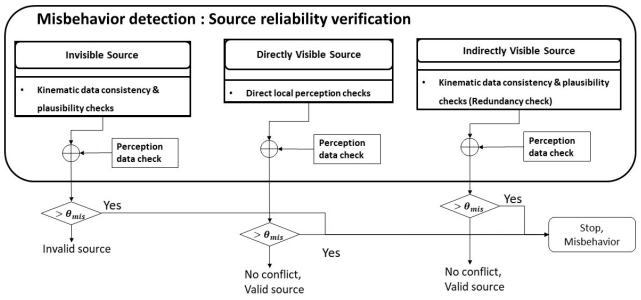


Fig. 5: Source Reliability Verification Module

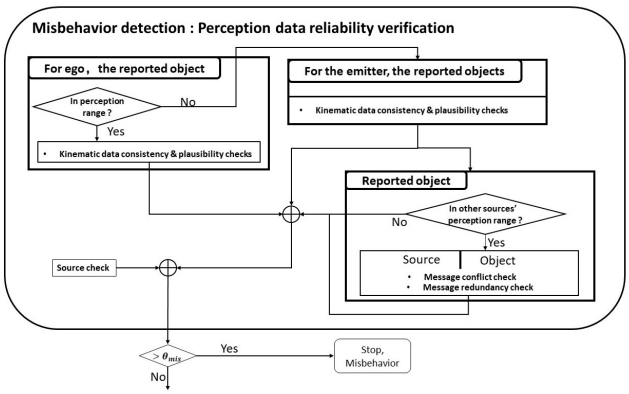


Fig. 6: Perception Data Reliability Verification Module

the consistency of the perception data over the past received perception information by the same source.

As shown in Fig.6, ego should determine the reported object by other sources, whether the reported object is in ego's local perception or not. If the reported object has already been in the local perception, the ego's local perception check has the highest priority. Then, ego should check the plausibility and consistency of the perception data with the previous data received from the same source. If this reported object is also in other sources' perception range, ego should check the conflict and the redundancy among these sources which have common perception. Hence, we use a cumulative sum operator to assess the misbehavior degree at the end of the misbehavior detection step. If the returned result is greater than the predefined misbehavior threshold θ_{mis}, we assume that the new received V2X message is sent by a misbehaving source. It needs to be noted here, this predefined misbehavior threshold θ_{mis} does not associate with the trust, but it indicates a misbehavior degree.

Trust Revision after detection of conflict

The trust in a misbehaving vehicle would decrease to zero in this step. When ego detects the conflict among the data provided by the neighboring sources, ego should reduce the trustworthiness of all sources which are associated to the conflict event. We define a conflict penalty function $g(E)$.

$$g(E) = \prod_{e=1}^{N} \alpha_{conf(e)}, e \in E \tag{5}$$

$\alpha_{conf(e)}$ represents the penalty coefficient of the conflict event, N represents the number of conflict events associated with the source, E represents a set of conflict events.

4) Trust revision L3: Opinions fusion: Opinions fusion module is the last phase of our framework. In case of conflict between the incoming CPMs, the trust into nodes should be revised when their reported opinions have more conflict degrees *Eq.*9 with other nodes' opinions reporting on the same information statement. In this paper, we used the Subjective Logic [12] approach.

The trust revision L3 consists of the following steps:

1) *Source Opinion Construction*
The ego vehicle considers each received CPM as evidence by the emitting node C of the presence or absence of the conflict object X. The Ego can evaluate the opinion w_X^C of vehicle C in the existence of conflict object X using the framework of subjective logic [12]. Let r represent positive evidence, *i.e.* the number of received CPMs from C where the conflict object X is present. Similarly, let s be the negative evidence, *i.e.* the number of CPMs where the conflict object X is absent. Then w_X^C takes the form:

$$w_X^C \Rightarrow \begin{cases} b(x) = \dfrac{r}{r+s+W} \\ b(\bar{x}) = \dfrac{s}{r+s+W} \\ u = \dfrac{W}{r+s+W} \end{cases} \tag{6}$$

The non-informative prior weight is expressed as a constant W, normally set to $W = 2$.

2) *Trust Discounting*
Ego's referral trust in C can be expressed as a binomial opinion where the opinions are defined as trust and distrust. Then, via the C's advice opinion about X, the *ego*'s derived opinion about X is expressed as follows:

$$w_X^{|Ego:C|} \Rightarrow \begin{cases} b_X^{[Ego:C]}(x) = P_C^{Ego}(t) b_X^C(x) \\ b_X^{[Ego:C]}(\bar{x}) = P_C^{Ego}(t) b_X^C(\bar{x}) \\ u_X = 1 - P_C^{Ego}(t) \sum\limits_{x \in \mathbb{X}} b_X^C(x) \\ a_X^{[Ego:C]}(x) = a_X^{Ego}(x) \end{cases} \tag{7}$$

3) *Calculation of the reference opinion*
The reference opinion is a simple average belief fusion [12] which relates to calculate the the degree of conflict.

$$w_X^{Ref} = \underline{\oplus}_{C \in \mathbb{C}} (w_X^{|Ego:X|}) \tag{8}$$

4) *Calculation of the degree of conflict*
The trust in sources should be revised as a function of the degree of conflict (DC) which was defined in [12]:

$$DC(w_x^C) = \frac{1}{2}(1 - u_X^{Ref})(1 - u_X^C) \sum_{x \in \mathbb{X}} |P_X^{Ref}(x) - P_X^C(x)| \tag{9}$$

5) *Trust Revision*
In this step, *ego* revises the trust in sources whose degree of conflict is over the average. The revision mechanism is defined in [13].

$$MC(w_X^C) = \max_{C \in \mathbb{C}} DC(w_X^C) \tag{10}$$

$$AC(w_X^C) = \frac{1}{card\{\mathbb{C}\}} \sum_{C \in \mathbb{C}} DC(w_X^C) \tag{11}$$

$$RW(w_{Ego}^C) = \begin{cases} \dfrac{MC(w_X^{C \in \mathbb{C}})(BC(w_X^C) - AC(w_X^{\mathbb{C}}))}{MC(w_X^{C \in \mathbb{C}}) - AC(w_X^{\mathbb{C}})} \\ \qquad\qquad , IF \quad (BC(w_X^C) - AC(w_X^{\mathbb{C}})) > 0 \\ 0 \qquad , \quad otherwise \end{cases} \tag{12}$$

$$\hat{w}_C^{Ego} \Rightarrow \begin{cases} \hat{b}_C^{Ego}(x) = b_C^{Ego}(x)(1 - RW(w_{Ego}^C)) \\ \hat{b}_C^{Ego}(\bar{x}) = b_C^{Ego}(x)(1 - RW(w_{Ego}^C)) + RW(w_{Ego}^C) \\ \hat{u}_C^{Ego} = u_C^{Ego}(1 - RW(w_{Ego}^C)) \\ \hat{a}_C^{Ego} = a_C^{Ego} \end{cases} \tag{13}$$

MC stands for max conflict, AC for average conflict, RW for revision weight, $\hat{w}$ for revised opinion respectively.

The node will be classified as misbehaving if the revised trust is under the predefined threshold θ_{L3}.

V. Framework Evaluation

We validate and evaluate our proposed framework by simulation on Artery [16]. We add the Collective Perception service module to Artery. We integrate the F2MD [17] framework to Artery and extend it with misbehavior attacks for Collective Perception and finally we add our trust-based framework on top of it.

A. T-junction scenario with Omission Attack

We evaluated the proposed trust framework in a T-junction scenario with obstruction as illustrated in Fig.7. The scenario includes 5 vehicles which are 4 vehicles exchanging CPMs, *i.e.* one vehicle under test (39, *ego*), two benign vehicles (63 and 69), one attacker denoted 26. Vehicle 1 is an unconnected vehicle which is incapable of transmitting V2V messages. The *ego* drives towards to the intersection with the intention to turn right. Meanwhile, unconnected vehicle 1 drives from right to left and the attacker is behind it. Then, 63 and 69 start moving from left to right and from right to left respectively. According to the traffic rule, vehicles coming from the right road have the higher priority to cross the intersection. When ego approaches the intersection, and due to the building which limits its local perception view, ego needs to choose a strategy to turn right based on the received CPMs from its neighbors. We assume that vehicle 1 is driving with a high speed, therefore in the usual case *ego* has to decelerate or to stop if it is aware about vehicle 1. In order to test the performance of the proposed framework, we implement an omission attack. The attacker 26 broadcasts a sequence of CPMs omitting the perception information

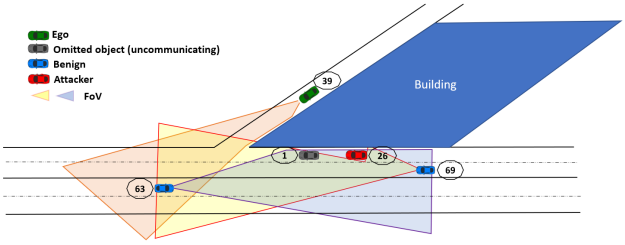


Fig. 7: T-junction scenario with omission attack

about the unconnected vehicle 1. The *ego* receives the CPMs from vehicle 26 which only contain the information about itself. Then the *ego* receives the CPMs which contain the information about vehicle 1 from vehicle 63 and 69. The vehicle 63 is configured to have a little higher speed than the other vehicles. When vehicle 1, 26, 69 approach to intersection, vehicle 63 was already in the intersection zone and was perceived by the *ego* for a short time. The *ego* received redundant information about vehicle 1 for few seconds.

In absence of an efficient solution of misbehavior detection, the *ego* can not confirm the existence of vehicle 1 and judge which vehicle (between vehicle 26, 69) is probably malicious. In other words, the *ego* can not confirm that the conflict information about vehicle 1 is caused by the omission attack from vehicle 26 or the ghost injection attack from vehicle 69. As a result, the *ego* is confused about the decision of crossing the intersection. If the *ego* believes that vehicle 69 makes a ghost injection attack then turns right, it's a very dangerous behavior that will cause a high risk collision. In opposite, if the *ego* chooses to yield to the vehicle 1 due to the existence of vehicle 1 and vehicle 69 makes a truly ghost injection attack about vehicle 1, this may cause a priority deprivation.

B. Simulation Settings

In the simulation, we set a constant velocity model. The sensors detect the object that are in their perception range. Obstacles that mask the perception are also taken into account. The simulation parameters are summarized as follows:

- Sensor range = 65m, 80m, 100m, 150m, 200m. All simulated vehicles are equipped with the same front sensor.
- CPM generation frequency = 100ms, 200ms, 500ms, 1s, 2s, 5s
- Default assigned trust score = 0.5, as the initial trust score for each new connected vehicle.
- Perceived Trust Bonus δ = 0.2, which is an offset when a vehicle is perceived by the ego.
- Conflict penalty coefficient α_{conf} = 0.9, *Eq.*5.
- Trust threshold = 0.1, if the trust on a vehicle is below this threshold, the vehicle is classified as misbehaving. Messages from this vehicle are then ignored completely.

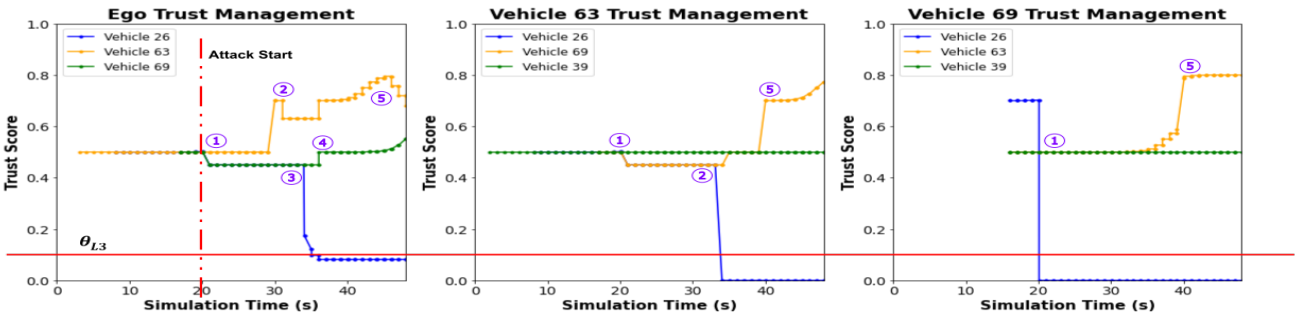
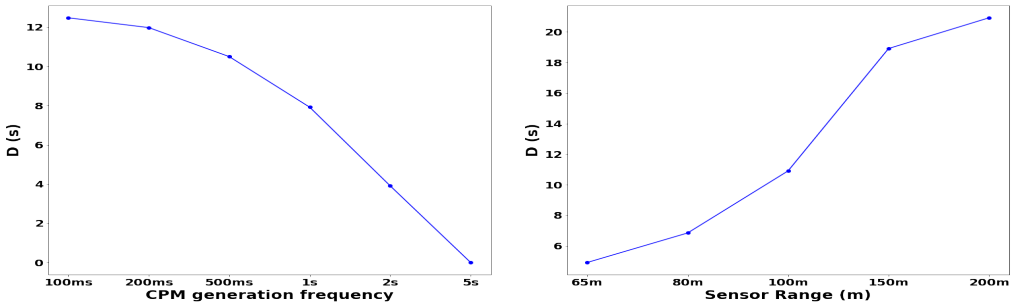


Fig. 8: Trust evaluation in each vehicle



(a) (b)

Fig. 9: the safety metric D with different CPM generation frequency and different sensor range

C. Performance Evaluation Analysis

1) Evaluation of the Trust Attribution Efficiency: To evaluate the performance of the proposed framework, we measure the trust values attributed by each vehicle to its neighbors during the simulation. We use a CPM transmission frequency of 1s and a sensor range of 65m. Omission attack starts at 20 seconds in the scenario.

As shown in Fig.8, at 20 seconds, vehicle 69 detects the unconnected vehicle 1 within its field of view (FoV). Meanwhile, vehicle 69 receives the CPMs transmitted by the attacker which don't contain the information about vehicle 1. Vehicle 69 confirmed the correctness of the kinematic attributes of the attacker (vehicle 26). However, it believes that the CPMs sent by vehicle 26 should include the perception information about vehicle 1. Hence, the trust score attributed vehicle 69 to the attacker drops to zero, as shown by Label ① . It also shows that ego and vehicle 63 detect a conflict event from the received CPMs of vehicle 69 and the attacker. This is because they are not able to perceive vehicle 1 directly through their local sensors. Therefore, they apply the conflict penalty function *Eq.*5. At 31 seconds, vehicle 63 is perceived by the *ego*. After 2 seconds, vehicle 1 enters in the FoV of vehicle 63. Vehicle 63 can confirm the kinematic attributes of vehicle 1, thus the trust score in attacker drops immediately. However, *ego* still can't perceive vehicle 1, as shown by label ② . *Ego* adds vehicle 63 in the same conflict event. When *Ego* receives enough confirmations about vehicle 1 through the CPMs of vehicle 63, *ego* starts to execute the trust revision algorithm discussed in Section IV.B.4, as shown by label ③ . Even without the perception of vehicle 1, the trust score in attacker drops below the threshold in *ego* after 3 seconds. When the *ego* eliminates the misbehaving vehicle, the *ego* returns the trust which is reduced by the conflict penalty function to other benign vehicles, as shown by Label ④ . Label ⑤ is associated to the event where vehicle 63 and vehicle 69 perceive each other. Then, the ego reduces the trust in vehicle 63 due to its disappearance from the ego's FoV.

2) Impact of misbehavior detection performance on safety risk assessment: we define the safety metric D to capture the ability of the proposed detection framework to inform higher level safety applications about safety risks. Concretely, we measure the delay between the misbehavior detection time and the time the ego reaches the intersection. This would give us an idea if our solution allows safety application to anticipate dangerous situation such as sudden braking or collision in the intersection. In another word, the earlier the erroneous information is detected (and consequently the attacker is detected), the longer would be the time to reach the intersection and thus to anticipate a potential collision for example. We formulate this delay in *Eq.*14, where $T_{Intersection}$ is the time at which the ego would reach the intersection and T_{mis} is the time the ego detects the vehicle omission.

$$D = T_{Intersection} - T_{mis} \tag{14}$$

a) The CPM generation frequency: In Fig.9(a), We evaluate 6 different CPMs frequencies ranging from 100ms

to 5s. The result shows that the proposed framework is not functional when the CPM frequency is equal or higher than 5s. It means that the ego is not able to detect that vehicle 26 is an attacker and that it omitted vehicle 1 in the CPMs before crossing the intersection. In this case, the ego does not simply receive enough observations about vehicle 1 from other vehicles to build each vehicle opinion (See Section IV.B). For this frequency, the misbehavior detection solution was not useful to anticipate the danger. The delay for the ego to arrive to the intersection since the misbehavior is detected is about 12.5 seconds when the frequency is 100ms and about 8 seconds when the frequency is 1s. For these frequencies, our framework was more reactive to detect the misbehaving vehicle (*i.e.* from the conflict assessment step to the final misbehavior detection step). As a conclusion, a higher CPM frequency in intersection scenarios with smart attackers allows a better misbehavior detection and leads to better reaction time to avoid dangerous situations.

b) The sensor range configuration: Another considered element which impacts the performance of the proposed framework is the sensor configuration. Here, we simulate the effect of sensor range configuration using the same safety metric. In Fig.9(b), the result highlights that the sensor range can improve significantly the vehicle reaction time. With a wider sensor range, vehicles have more perception. Then, they have more chance to detect the conflict (*i.e.* ghost object or omitted object). With increased sensor range, the *ego* is able to receive the contradictory CPMs when it is relatively still far from the intersection. Consequently, the *ego* is aware about the omission attack in advance. On the other hand, the fusion algorithm would get more precision and converge more rapidly due to more observations. However, it doesn't really mean that a wider sensor range is always better. [18] shows that the sensor configuration can impact negatively the object perception ratio and the packet delivery ratio and as a result the performance degrades as the sensor range increases. For this, a trade-off between sensor range configuration, communication performance and safety level needs to be found.

VI. Conclusion

In this paper, we present a misbehavior detection framework based on trust assessment for collective perception services. The framework uses different levels to update the trust score given to participating nodes. It uses the subjective logic when a conflict is detected among nodes. The proposed framework is evaluated in T-junction traffic scenario for object omission attack. The simulation results show that the proposed framework allows *ego* to quickly and effectively detect and remove the misbehaving nodes. We also show that our solution has a positive impact on safety risk anticipation in intersections scenarios when the CPM generation frequency and the sensor range are suitably chosen. This general framework opens a series of questions about the scalability, the feasibility in a complex models and etc. As a future work, we plan to evaluate our solution taking into consideration more advanced parameters such as perception uncertainty and more complex attacks on new road scenarios. We plan also to study the extension of our solution to semi-global scheme based for example on detection by RSUs.

Acknowledgments

This research work has been carried out in the framework of the Technological Research Institute SystemX, and therefore granted with public funds within the scope of the French Program *Investissements d'avenir*.

References

[1] "Intelligent Transport Systems (ITS); Cooperative Perception Services, CPS," European Telecommunications Standards Institute, Standard.

[2] A. Rauch, F. Klanner, R. H. Rasshofer, and K. Dietmayer, "Car2x-based perception in a high-level fusion architecture for cooperative perception systems." in *Intelligent Vehicles Symposium*. IEEE, 2012.

[3] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Communications Surveys Tutorials*, 2019.

[4] J. Kamel, I. B. Jemaa, A. Kaiser, L. Cantat, and P. Urien, "Misbehavior detection in c-its: A comparative approach of local detection mechanisms," in *2019 IEEE Vehicular Networking Conference (VNC)*.

[5] N. Bißmeyer, S. Mauthofer, K. M. Bayarou, and F. Kargl, "Assessment of node trustworthiness in vanets using data plausibility checks with particle filters," in *2012 IEEE Vehicular Networking Conference (VNC)*.

[6] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *2019 IEEE International Conference on Communications (ICC)*.

[7] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in vanet," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*.

[8] M. Ambrosin, L. L. Yang, X. Liu, M. R. Sastry, and I. J. Alvarez, "Design of a misbehavior detection system for objects based shared perception v2x applications," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*.

[9] M. Raya, P. Papadimitratos, V. D. Gligor, and J.-P. Hubaux, "On data-centric trust establishment in ephemeral ad hoc networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*.

[10] C. Allig, T. Leinmüller, P. Mittal, and G. Wanielik, "Trustworthiness estimation of entities within collective perception," in *2019 IEEE Vehicular Networking Conference (VNC)*.

[11] F. Geissler, A. Unnervik, and M. Paulitsch, "A plausibility-based fault detection method for high-level fusion perception systems," *IEEE Open Journal of Intelligent Transportation Systems*, 2020.

[12] A. Jøsang, *Subjective Logic*. SpringerLink, 2016.

[13] A. Jøsang, J. Zhang, and D. Wang, "Multi-source trust revision," in *2017 20th International Conference on Information Fusion (Fusion)*.

[14] R. W. van der Heijden, A. Al-Momani, F. Kargl, and O. M. F. Abu-Sharkh, "Enhanced position verification for vanets using subjective logic," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*.

[15] J. Muller, T. Meuser, R. Steinmetz, and M. Buchholz, "A Trust Management and Misbehaviour Detection Mechanism for Multi-Agent Systems and its Application to Intelligent Transportation Systems," 2019.

[16] R. Riebl, H. J. Günther, C. Facchi, and L. Wolf, "Artery: Extending Veins for VANET applications," *2015 International Conference on Models and Technologies for Intelligent Transportation Systems, MT-ITS 2015*.

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

[18] G. Thandavarayan, M. Sepulcre, and J. Gozalvez, "Cooperative perception for connected and automated vehicles: Evaluation and impact of congestion control," *IEEE Access*, 2020.