Fast and Secure Multi-hop Broadcast Solutions for Inter-Vehicular Communication

Wafa Ben Jaballah*, Mauro Conti†, Mohamed Mosbah*, Claudio E. Palazzi†
*Univ. Bordeaux, LaBRI, CNRS, France, Email: wafa.benjaballah@labri.fr, mosbah@labri.fr
†Univ. of Padua, Italy, Email: conti@math.unipd.it, cpalazzi@math.unipd.it

Abstract—**Inter-vehicular communication (IVC) is an important emerging research area that is expected to contribute considerably to traffic safety and efficiency. In this context, many possible IVC applications share the common need for fast multi-hop message propagation, including information such as position, direction, speed, etc. Yet, it is crucial for such data exchange system to be resilient to security attacks. Conversely, a malicious vehicle might inject incorrect information into the inter-vehicle wireless links leading to life and money losses, or to any other sort of adversarial selfishness (e.g., traffic redirection for the adversarial's benefit). In this work we analyze attacks to the state of the art IVC based safety applications. Furthermore, this analysis leads us to design a Fast and Secure Multi-hop Broadcast Algorithm (FS-MBA) for vehicular communication, which results resilient to the aforementioned attacks.**

I. Introduction

Inter-vehicular communication (IVC) is amongst the most promising and challenging applications of Vehicular Ad-hoc Networks (VANETs) [1], [2]. Many applications are possible in this context, yet local danger warning systems remain the most prominent ones. Most of these safety related applications, including state of the art ones, share properties that put them into the same class of solutions: IVC based vehicular safety applications [3], [4], [5], [6], [7], [8]. These common properties are as follows:

1) Communication is generally vehicle-to-vehicle (V2V), without infrastructure.
2) Vehicles exchange messages containing their position, direction, speed and possible dangers.
3) Broadcast messages have to be propagated as quickly as possible within a certain area of interest, even through multi-hop forwarding.
4) Specific algorithms are employed to choose as few forwarders as possible over the message's multi-hop path in order to fasten the propagation of alert messages over their area of interest.
5) Vehicles' information such as position, direction, speed and transmission range is used to feed the forwarder selection algorithm.

Clearly, the effectiveness of such safety related application is based on the reliability of broadcast information. Furthermore, the last property in the list indicates that even the effectiveness (speed) of the propagation mechanism is based on the reliability of the exchanged messages' content. Therefore, secure communication in this context is a crucial aspect that must not be overlooked.

To discuss attacks to IVC based safety applications, we consider a state of the art protocol which is representative of this class of applications: the Fast Multi-hop Broadcast Algorithm (FMBA) [3]. Since discussed attacks and solutions depend on the aforementioned five properties (also possessed by FMBA), FMBA allows us to clarify the explanation thanks to a practical case study, yet without loss of generality.

Contribution. Through the use of a representative case study, we analyze the security threats to state of the art IVC based safety applications also proposing countermeasures for these threats. We hence propose a solution which is both fast and secure in broadcasting safety related messages: Fast and Secure Multi-hop Broadcast Algorithm (FS-MBA).

Organization. This survey paper is organized as follows. The next section reviews main work and background related to the security issues of IVC, in particular when used to fast broadcast alert messages in vehicular networks. Section III discusses the functioning of the case study algorithm (FMBA). Sections IV, V, and VI detail possible attacks, whereas respective solutions are presented in Sections VII, VIII, and IX. Section X wraps up the solutions in a single algorithm (FS-MBA), and discusses some performance issues. Finally, in Section XI conclusions are drawn.

II. Background and Related Work

Inter-Vehicular Communication (IVC) is an important component of the intelligent transportation system [1], [9], [10], [11]. It enables a driver (or her/his vehicle) to communicate in a multi-hop fashion with other drivers located out of the radio range. As a result, information gathered from IVC can foster road safety and transportation efficiency. Benefiting from the large capacities (in terms of both space and power) of vehicles, the nodes of these networks can have long transmission ranges and unlimited lifetimes. Main IVC applications can be categorized into three classes [7]:

- Information and warning functions: Dissemination of road information (including accidents, road congestion,

[0] Manuscript received ...

[1] Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission for the PRISM-CODE project (Privacy and Security for Mobile Cooperative Devices) under the agreement n. PCIG11-GA-2012-321980. This work is also partially supported by the MIUR/PRIN ALTER-NET, the UNIPD/PRAT Web Squared projects, and the Cluster of Excellence CPU, University of Bordeaux.

etc.) to remote vehicles.
- Communication based longitudinal control: Exploiting the "look through" capability of IVC to help avoiding accidents.
- Co-operative assistance system: Coordinating vehicles at critical points such as blind crossing (a crossing without light control) and highway entries.

In IVC systems, several applications require multi-hop broadcast to inform vehicles (and drivers) about road data, delivery announcements, traffic congestion, proximity with other vehicles, accidents and even entertainment related information for passengers [3], [4], [5], [6], [7], [8], [12]. The simplest broadcasting mechanism is flooding, where messages are re-broadcast by each node that receives them. Although very simple, yet this technique may lead to high message collision probability and data redundancy, thus resulting rather inefficient in terms of radio resource usage and message delivery time.

When a message is disseminated to receivers beyond the transmission range, the multi-hopping could be used. However, multi-hop broadcast can consume a significant amount of wireless resources for unnecessary retransmissions. These facts are important motivations for many works focused on efficient multi-hop message broadcast in VANETs; as high mobility and high number of nodes make multi-hop broadcast significantly more challenging in a VANET environment.

The broadcast delivery time represents one of the main issues of IVC. Indeed, it has been proven that this characteristic is strictly related to both the number of relays of the messages (hops) and the network congestion [3], [4], [8], [13], [14], [15], [16], [17], [18]. In [13], the demand-driven transmission (DDT) protocol adjusts the timing of rebroadcast packets such that the vehicle furthest away from the source node retransmits earlier than the other nodes. Ad hoc multi-hop broadcast and urban multi-hop broadcast are proposed in [8] for vehicular networks. These protocols are designed to address the broadcast storm, hidden node, and reliability problems in multi-hop broadcast. Sender nodes try to select the furthest node in the broadcast direction, to assign the function of forwarding and acknowledging the packet without any a prior topology information. That is, senders select the furthest node without knowing the identification (ID) or position of their neighbors. FMBA [3] aims at reducing the number of hops traversed by a message, in order to minimize the propagation delay of a message. Vehicles in a car platoon dynamically estimate their transmission range and exploit this information to efficiently propagate a broadcast message with as few transmissions as possible. In essence, the farthest vehicle in the transmission range of a message sender or forwarder will be statistically privileged in becoming the next (and only) forwarder. In [4], authors have enhanced the fast broadcast algorithm using heterogeneous transmission range. Unlike [3], the authors select the forwarder of the message as the vehicle which transmission spans farther, not the farthest vehicle in the transmission range of the sender.

In summary, several multi-hop broadcast algorithms have been proposed. These algorithms generally share the set of properties mentioned in Section I, thus falling into a single class of solutions. Unfortunately, they have all been developed without security in mind, whereas security is a fundamental problem in this context which should not be overlooked [19]. Indeed, attackers might run malicious actions to inject false information or alarm, thus rendering ineffective the safety application [20], [21], [22], [23], [24], [25], [26]. More in detail, the authors in [24] classify the attacks on vehicular communications into:

- **Bogus information.** One or several legitimate members of the network send out false information to misguide other vehicles about traffic conditions. In order to cope with such misbehavior, the received data from a given source should be verified by correlating and comparing them with those received from other sources.
- **Cheating on positioning information.** Injection of a false position by a malicious vehicle pretending to be at a claimed position.
- **ID disclosure of other vehicles.** This is to track their location. A global entity can monitor trajectories of targeted vehicles and use this data for many purposes, we could take the example of some car rental companies that track their own cars.
- **Denial of Service.** The attacker may want to bring down the IVC or even cause an accident. Example of attacks include channel jamming and aggressive injection of dummy messages.
- **Masquerade.** The attacker claims to be another vehicle by using false identities.

In this work, we analyze the security of a representative algorithm for state of the art IVC based safety applications, and propose countermeasures to handle the security threats. In particular, we focus on one of the main threats to safety application: the possibility to attack the protocol to impede its useful service. For ease of exposition, but without loss of generality, we focus especially on FMBA as it embodies both a state of the art solution and a representative example of the IVC based vehicular safety applications class possessing all the five properties mentioned in Section I. Indeed, problems and possible countermeasures identified for FMBA can be adapted also to other protocols/algorithms, belonging to the same general class of applications sharing the aforementioned set of properties.

III. Preliminaries and Notation

In this section we present the notation (summarized in Table 1) and assumptions used in this paper. Furthermore, we describe in details FMBA, the case study chosen to represent IVC based vehicular safety applications.

It is worth noting that FMBA is designed to speed up multi-hop broadcast both frontward and backward. However, for the sake of clarity, in this work we refer only to the case where alert messages have to be sent only backward, with respect

Symbol	**Definition**
$CMBR$	Current Maximum Back Range
$CMFR$	Current Maximum Front Range
$LMBR$	Latest-Turn Maximum Back Range
$LMFR$	Latest-Turn Maximum Front Range
$MaxRange$	How far the transmission is expected to go backward before the signal becomes too weak to be intelligible
d	Distance between two vehicles
CW	Contention Window
$CWMax$	Maximum Contention Window
$CWMin$	Minimum Contention Window
$Hello$	Hello message transmitted by a vehicle in the estimation phase to update the transmission range
drm	declared transmission range in the $Hello$ message
P	The prover vehicle
V	The verifier vehicle
R	The geographical region

Fig. 1. Notation

to the vehicular traveling direction (the frontward case is just specular).

A. Model assumptions

To simplify the discussion we have made the following assumptions about the general model we are considering:

- We suppose that at most one malicious vehicle is on the network.
- There are no obstacles and no buildings in the road.
- The hearing communication range is symmetric. It means that if a vehicle V hears a vehicle P, then we assume that P can also hear V.
- We suppose that there are N vehicles arranged in the platoon. A platoon can be looked at as a collection of nodes/vehicles connected by a wireless local area network (LAN), and are engaged in following each other longitudinally.
- A vehicle V does not know its transmission range.
- The verifier node V communicates directly with the verified node P.
- Each vehicle knows its own location, for instance, using GPS that provides accurate information about time and position.
- All the vehicles belong to a Public Key Infrastructure [27], [28]; i.e., each vehicle has a public/private pair of keys and a unique identity certified by a Certification Authority. We assume that the certification authority corresponds to the government agency responsible to assign licence plates: a vehicle can be used only if it is provided with a unique licence plate, a PKI certificate associated to its plate ID, and the public key of the Certification Authority. We assume that certificate revocation lists are updated at given time interval (e.g., daily) by the vehicle and stored in a local memory[1].
- The power and computational resources are supposed largely adequate for our application's requirements.
- The network is loosely time synchronized.

B. Fast Multi-Hop Broadcast Algorithm (FMBA)

The aim of the Fast Multi-Hop Broadcast Algorithm (FMBA) is to reduce the time required by a message to propagate from the source to the farthest vehicle in a certain area of interest [3]. To achieve this goal, FMBA exploits a distributed mechanism for the estimation of the communication range of vehicles. These communication range estimations are obtained by exchanging a number of $Hello$ messages among the vehicles, and are then used to reduce the number of hops an alert message has to traverse to cover a certain area of interest. This leads to a decrease in the number of transmissions as well as the time required by a broadcast message to reach all the cars following the sender within a certain distance.

This scheme is composed by two phases: the estimation phase, and the broadcast phase. The former is continuously active and is meant to provide each vehicle with an up-to-date estimation of its transmission range. Instead, the latter one is performed only when a message has to be broadcast to all vehicles in the sender's area of interest. In order to forward a packet, each receiver has to compute its waiting time before attempting to forward the message. This waiting time is expressed through a contention window (CW) computed using Equation 1.

$$CW = \left| \frac{(MaxRange - d)}{MaxRange} \times (CWMax - CWMin) + CWMin \right|. \tag{1}$$

When a car has to send or forward a broadcast message it computes the $MaxRange$ value in the broadcast message as the maximum between $LMBR$ and $CMBR$ values. To avoid unnecessary transmissions, all vehicles between the original sender and the current forwarder abort their attempt to forward the message; whereas all vehicles behind the current forwarder compute a new CW based on last forward's parameters to participate in the election for the forwarder on the next hop.

In Figure 2, we present the CW of a vehicle V versus the position of different vehicles. Vehicles compute their CW through Equation 1. The farther a vehicle is from the source of a broadcast message, the smaller its CW results.

The waiting time is a value computed randomly within CW. Thus, as presented in Figure 2, if we assume distances among vehicles as $d(D,V) \geq d(C,V) \geq d(B,V) \geq d(A,V)$, then the expectation of vehicles' CWs generated by FMBA results $CW(D) \leq CW(C) \leq CW(B) \leq CW(A)$. Therefore, in the considered example D has the highest probability to become the next forwarder of the message transmitted by vehicle V, since its waiting time is randomly chosen within the smallest CW among those assigned by FMBA to vehicles A, B, C, and D. This leads to a general reduction of the number of hops

[1] Please note that the list of revoked ID for *all* passenger vehicles in USA could fit in some 1GB storage.

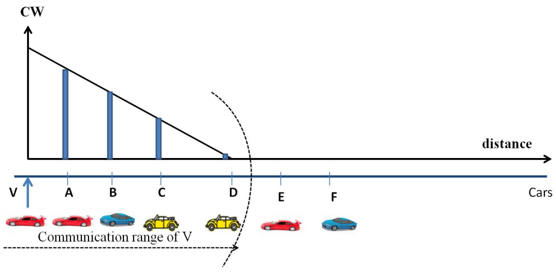


Fig. 2. Contention window versus distance

and time needed by a broadcast message to traverse its area of interest.

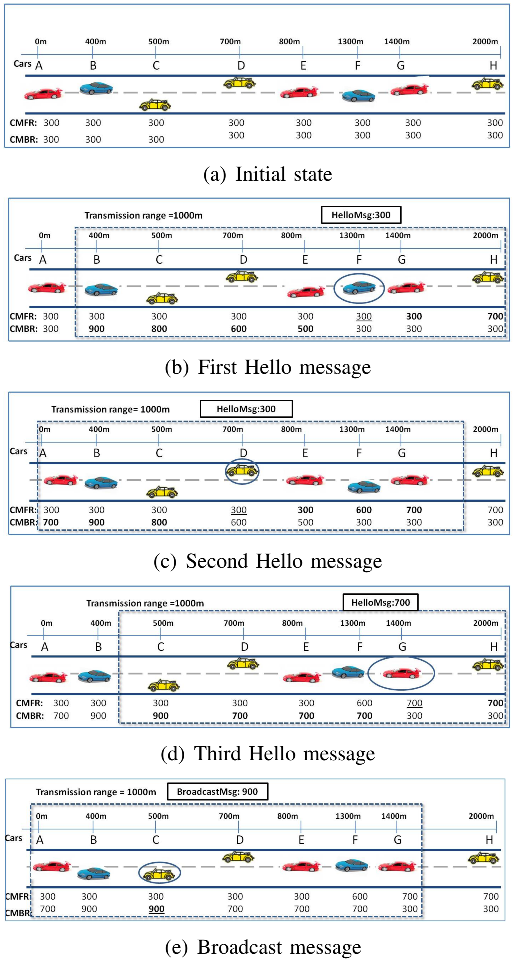
C. Example of FMBA

Since our analysis is based on the FMBA case study, in this section we present an example of execution of this algorithm. To help the readers, the example is presented in Figure 3.

We first suppose to have an initial state (Figure 3(a)), where all cars have initial values $CMFR = CMBR = 300m$; whereas the actual transmission range of each vehicle is $1000m$. We consider a $2000m$ portion of road and we suppose that car F sends the first $Hello$ message (Figure 3(b)) broadcasting its $CMFR$ (i.e., $300m$). This value is used by cars in front of car F to update their $CMBR$ values and by cars following car F to update their $CMFR$ values. The value of $CMBR$ (respectively $CMFR$) of each vehicle receiving the broadcast $Hello$ message is updated with the maximum among: i) the broadcast $CMFR$ value; ii) the previous value of $CMBR$ (respectively $CMFR$), and iii) the distance from car F. The rationale of this is that $CMFR$ represents how far a message was heard coming from the front of the car. Thus, when received in a broadcast message, this value is used to compute how far following vehicles are able to hear a message coming from their front. The value of $CMFR$ can hence be one of the parameters to determine the $CMBR$, which corresponds to the backward maximum transmission range (that will be declared when transmitting an alert message). Similar but reversed considerations could be made if we considered the frontward direction for alert message propagation.

In Figure 3(c), car D sends a second $Hello$ message broadcasting a $CMFR$ of $300m$ so that vehicles in D's transmission range can update their $CMBR$ and $CMFR$ as explained. Then, car G sends the third $Hello$ message (Figure 3(d)).

In the next step (Figure 3(e)), car C has to broadcast an alert message. We can remark that the algorithm has modified C's $CMFR$ and $CMBR$. The broadcast message issued by car C includes in its $MaxRange$ field the latest $CMBR$ value. We remark that the maximum transmission range, estimated by car C, after only three $Hello$ messages is $900m$ over an actual one of $1000m$. Cars following C and hearing the broadcast message can then compute their CW through Equation 1 so as to have a forwarder, possibly close to the end of C's backward transmission range.


0m
400m
500m
700m
800m
1300m
1400m
2000m
Cars
A
B
C
D
E
F
G
H
CMFR: 300 300 300 300 300 300 300 300
CMBR: 300 300 300 300 300 300 300 300
Transmission range =1000m
HelloMsg:300
CMFR: 300 300 300 300 300 300 300 700
CMBR: 300 900 800 600 500 300 300 300
Transmission range= 1000m
HelloMsg:300
CMFR: 300 300 300 300 300 600 700 700
CMBR: 700 900 800 600 500 300 300 300
Transmission range =1000m
HelloMsg:700
CMFR: 300 300 300 300 300 600 700 700
CMBR: 700 900 900 700 700 700 300 300
Transmission range = 1000m
BroadcastMsg: 900
CMFR: 300 300 300 300 300 600 700 700
CMBR: 700 900 900 700 700 700 300 300

(a) Initial state

(b) First Hello message

(c) Second Hello message

(d) Third Hello message

(e) Broadcast message

Fig. 3. Example of a fast broadcast algorithm

IV. Attack #1: Position-Cheating Attack

In this section, we present a position cheating attack. This attack is mainly linked to properties 2), 4), and 5) mentioned in Section I and its goal is to induce delay by increasing the CW of honest vehicles.

A. Overview

A malicious node could announce in a $Hello$ message a false position being more distant than the real one. Then, honest nodes eventually receiving an alert broadcast message will compute unnecessarily large CWs, thus slowing down the forwarding process. For ease of presentation, Figure 4 depicts the impact of this attack reporting the CWs of some vehicles depending on their distance from the original sender/forwarder (vehicle V) of the alert message.

In particular, as the CW of each vehicle is computed through Equation 1, without the malicious vehicle the CW function should vary as shown by the continuous line in the Figure 4 (from its maximum in correspondence of vehicle V to its minimum at the end of the transmission range which is assumed to be close to vehicle D). Instead, if during the estimation phase, a malicious vehicle within V's transmission range sent a $Hello$ message to declare a fake position corresponding to *M'* in the Figure 4, the transmission range estimation of vehicle *V* would be wrongly computed as the distance from *V* to *M'*, instead of the distance from *V* to *D*. This leads vehicles *A*, *B*, *C* and *D* to wrongly compute their CWs with higher values. In fact, those nodes will consider the minimum CW in correspondence of position of *M'*, as shown by the dotted line in Figure 4.

This simple, yet effective attack, modifies the computation of CW, increasing in average the contention period of each node before any forwarding transmission can take place, hence slowing down the transmission of the alert message.

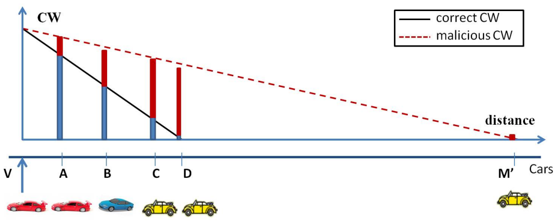


Fig. 4. Impact of distance cheating on the waiting time

B. Description

In this section, we present Algorithm 1 executed by a malicious vehicle M. In fact, it cheats about its claimed position, declaring a false position (Algorithm 1, line 2). Then, M broadcasts its $Hello$ message (Algorithm 1, line 3) indicating its claimed position.

Algorithm 1: Position-Cheating attack executed by a malicious vehicle M

```
1 Input: $real\_position$: (Real position of $M$);
    $claimed\_position$: (Claimed position of $M$);
    $vehicle\_ID$: $ID$ of the vehicle $M$;
    $drm$: declared max range of $M$;
    $Hello$ msg: $Hello$ message generated by $M$;
2 $claimed\_position > real\_position$;
3 $M \rightarrow *$: Hello msg = $< vehicle\_ID, claimed\_position, drm >$ ;
```

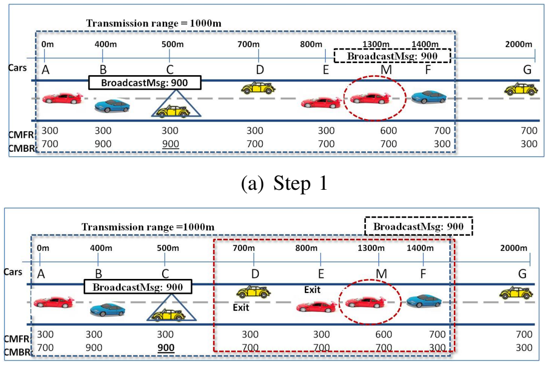
V. Attack #2: Replay broadcast message attack

In this section, we present the replay broadcast message attack which has the aim to enforce honest nodes to not collaborate and not forward the packets. This attack is mainly linked to properties 1), 2), and 3) mentioned in Section I. In particular, the adversary repeat the transmission of the same message, enforcing other vehicles to not forward packets.

A. Overview

In this scenario, we consider an honest node which broadcasts a message to all the receivers in its transmission range. We suppose that the adversary intercepts the broadcast message and rebroadcasts it without waiting. First, we remark that all the nodes that receive this same broadcast message from the front, the attacker push them to restart the broadcast procedure (as explained in the forwarding procedure of a broadcast message). Second, all the nodes that receive this message from the back, stop trying to forward this message. In fact, according to the FMBA, the messages has been already propagated over the considered vehicles, and these vehicles will exit the forwarding procedure. The adversary could repeat broadcasting the same message, pushing the nodes to not forward the packet, by just restarting at each time the broadcast process.

In more details, let us consider the scenario depicted in Figure 5(a). The honest car (C) forwards its messages. In this attack scenario, we suppose that the vehicle (M) is malicious and does not wait for the expiration of its time interval; it sends the message immediately. When receiving the message, vehicles which are behind M (car F in Figure 5(b)) restart the broadcast process, whereas nodes which are in front of M (cars D and E in Figure 5(b)) will exit the forwarding process.



(a) Step 1

(b) Step 2

Fig. 5. Example of enforcing non cooperation attack

To summarize this attack, a malicious node M could do some operations:

1) M does not modify the message but just broadcasts it.
    - Nodes which are behind M restart the forwarding process, thus wasting time.
    - Nodes which are in front of M exit the forwarding process.

   No one could forward the packet if the adversary repeats every message sent by the forwarder or the sender vehicle.
2) M modifies the broadcast message and forwards it with a high $MaxRange$ so as to generate slow forwarding hops with vehicles employing unnecessarily high CW values.
3) M forwards the message with a low $MaxRange$ in order to increase the probability that more than one vehicle will simultaneously attempt to forward the message, thus resulting in transmission collision and hence in forwarding delay.

B. Description

In this section, we present the Algorithm 2 executed by the malicious vehicle M. In fact, a source S broadcasts a message (line 2) and the malicious vehicle M retransmits it without waiting (line 4).

Algorithm 2: Enforcing non cooperation attack executed by the malicious vehicle M

```
1 Input S: the sender of the message;
  $broadcast$ msg: the broadcast message of $S$;
2 $S \rightarrow$*: $broadcast$ msg;
3 $M$ intercepts the $broadcast$ msg and does not wait for the expiration
  of its waiting time;
4 $M \rightarrow$ *: $broadcast$ msg;
```

VI. Attack #3: Interrupting forwarding attack

The goal of this attack is to degrade the network performances by impeding alert message relaying. This attack is mainly linked to properties 1), 3, and 4) mentioned in Section I.

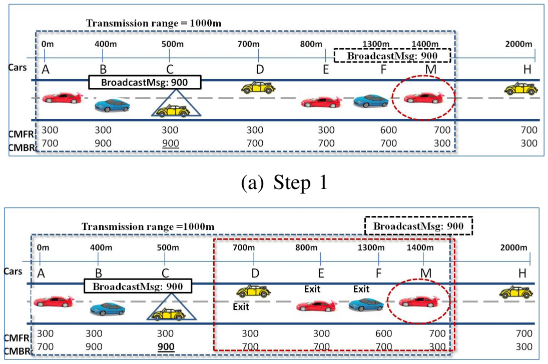
A. Overview

In this attack, the forwarder vehicle is malicious and tries to broadcast a message frontward but not backward. To do so, the malicious node has to be located at the end of the transmission range and be endowed with a directional antenna. By forwarding the alert message only frontward, vehicles in front of the malicious node will abort their forwarding procedure, as the message has already been sent farther than their position. On the other hand, vehicles behind the malicious node will simply not receive any message, thus interrupting the forwarding procedure.

Let us consider the attack scenario depicted in Figure 6(a) and Figure 6(b). The honest vehicle C broadcasts a message, while the chosen forwarder vehicle is M (using the forwarding procedure in FMBA algorithm). Assume that M is a malicious vehicle (Figure 6(a)) that forwards the message on front (not backward) by adjusting its transmission range. Vehicles (D, E, and F) will exit the forwarding process (Figure 6(b)) because they have received the same message from the back. Thus, the forwarded message will never be forwarded to the other cars. The impact of this attack in Figure 6(b) is that it avoids the transmission of messages. This attack could damage the network, especially when there are many malicious vehicles (see Figure 7) that collaborate together, in order to not forward or even limit the propagation of the alert message. These attacks can incur great security threats to vehicular communications. We observe (Figure 7) that n malicious nodes might collaborate together, in order to block the transmission of messages and not forward them in n zones.

B. Description

The malicious forwarder executes the following algorithm in order to stop the propagation of the alert message in the network. A broadcasts a message (Algorithm 3, line 2). The malicious vehicle intercepts it and transmits it to the vehicles in front of it (Algorithm 3, line 4). This attack disrupts the packet transmission process and blocks the transmission of the broadcast message.



(a) Step 1

(b) Step 2

Fig. 6. Impact of a malicious forwarder node

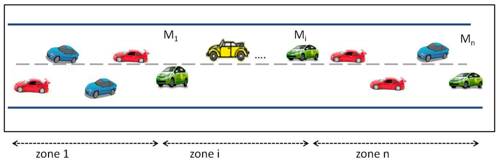


Fig. 7. Degrading performance attack executed by n malicious vehicles

Algorithm 3: Degrading performance attack executed by a malicious forwarder vehicle M

```
1 Input: $broadcast$ msg : the broadcast message;
  $A$: the sender of the broadcast message;
2 $A \rightarrow *$: $broadcast$ msg;
3 // M is the forwarder of the message;
4 $M \rightarrow$ front vehicles of $M$: $broadcast$ msg;
```

VII. Solution to attack #1: False Position Detection

Position dissemination is crucial for the fast broadcast algorithm [3]. Thus, forged position information has severe impact regarding the performance and security of the algorithm. We propose a detection mechanism that is able of recognizing nodes cheating about their position. Unlike other proposals described in the literature ([29], [30], [31]), our detection mechanism does not rely on additional hardware. Instead, our solution uses collaborative neighbors. We present an overview and a detailed description of our false position detection mechanism.

A. State of the art solutions for false position detection

Accurate information on position is crucial for IVC based vehicular safety applications. To this aim, detection mechanisms have been proposed in this context to recognize nodes cheating about their location [29], [30], [31], [32], [33], [34]. Position verification approaches can be grouped into two main categories ([29], [35]): infrastructure based and infrastructure-less based approaches.

1) Infrastructure based solutions: This approach uses special hardware dedicated infrastructure to verify the position of other vehicles. The solutions in [29] use multiple sensors to monitor and calculate trust values for position information. There are two classes of position verification sensors: autonomous and cooperating sensors. In fact, autonomous sensors work autonomously on each node and contribute their results to the overall trust ratings of neighbors. Cooperating sensors use the information exchange between the neighbors to verify positions.

In [30], the authors propose a specific use of infrastructure called "verifiable multilateration", employing three or more Road Side Units (RSUs). These RSUs send a synchronized challenge-response message to a vehicle. Then, the verification of its position is based on the consistency in response time calculation, to verify an announced location of a vehicle. Authors in [30] also exploit an history of movement of vehicles to detect forged data.

The solution in [31] uses verifiers at special locations. These verifiers define an acceptable distance between each others. The verification procedure then works as follows. First, the prover P sends a beacon containing its position. Then a verifier V replies with a challenge (a message) via radio. After receiving the challenge, P has to answer via ultrasound. The verifier computes the time required to receive the response (according to the defined acceptable distance for V). Then, P is approved to be within the region R of the verifier. This solution needs specific infrastructure: the verifiers. More specifically, these verifiers attempt to verify location claims for region R that are "near" V. Clearly, verifier nodes are assumed to be trusted and using secure communication among themselves.

An approach in [33] achieves position verification based on reception of beacons. First, the verifier nodes are divided in acceptors and rejectors. The acceptor nodes are distributed in the region R which is to be controlled. Verifier nodes could decide whether a vehicle is in a region R or not. In this approach, verifier nodes (acceptors and rejectors) are placed on distinct places, and are temporally synchronized among each other. If a vehicle P wants to prove its position (P wants to verify if it is in region R or not), it sends a beacon. As verifiers are placed in a region R, then the first verifier which receives the beacon could decide whether the position of P is acceptable or not. In fact, if the signal of P first reaches a rejector, then P is not in region R. Otherwise, if the first reached verifier is an acceptor, then P is approved to be in region R.

In [34], the proposed solution depends on two directional antennas. Each vehicle periodically sends a message containing its location together with its own two lists of front and back neighbors. A vehicle will decide on the relative positions of its one-hop neighbors based on the messages it receives.

2) Infrastructure-less approaches: Infrastructure-less approaches could be classified on parameter based and model based approaches [29].

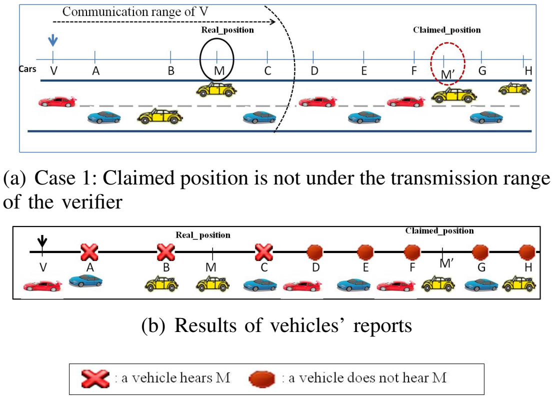
- **Parameter based approaches.** Vehicles check whether a claimed node's position is within a degree of accuracy from the actual one. This check is based on acceptable values of some network and traffic parameters, such as i) packet's timestamps consistency with current time; ii) acceptance range which assumes that no neighbor is further than the maximum transmission range. This approach assumes that the transmission range is fixed. The solution in [36] proposes distance bounding of vehicle's positions also involving verifiers and provers. This approach mainly detects distance enlarging fabrication. The verifier chooses the best common neighbor between the sender and him. This neighbor will give an estimated location of the prover. If that estimated location is not within a certain error distance of both verifier and neighbor, the verifier considers the node as malicious.
- **Model based approaches.** These solutions compare the regular behavior of the system and current actions to identify anomalies that could indicate malicious behaviors [37]. Each node periodically broadcasts its database containing information about observed nodes. When a broadcast is received, the contents are merged into the receiving node's database. Each node periodically examines events in its database searching for the scenario with the least number of malicious nodes. The disadvantage of this category of solutions is that a big search space of possible scenarios is needed to ensure efficacy.

B. Overview of the proposed cheating position detection

For the sake of clarity, we present in this section some false position attacks, distinguishing among them depending on the claimed position and the real position of the verified vehicle. The main participants are the verifier vehicle V, the prover vehicle M and the other vehicles in the road. M claims a position in the $Hello$ message. The verifier vehicle V uses information, collected by collaborative vehicles, to decide whether M is a cheater or not. We elaborated three scenarios to deal with the false position attacks. The first scenario considers that there is at most one malicious vehicle in the road. The second scenario assumes that there is one malicious vehicle (either the prover vehicle M or one of the reporting vehicles). The third scenario assumes that the overhearing capabilities of the vehicles are asymmetric.

1) Scenario 1: In this attack scenario, we consider that there is at most one malicious vehicle. M could be either a malicious vehicle or an honest one. We distinguish three cases based on the real position and the claimed position of M.

A report is a message sent by a vehicle, to indicate that it does hear another vehicle message or not. In the first case, as presented in Figure 8(a), M claimed a position M' that is not included in the actual transmission range of the verifier V. Vehicle V collects information from neighbor vehicles in order to decide whether the claimed position of M is fake or not. The notation used to indicate whether a vehicle is hearing or not message M is reported in Figure 8(c). This legend is also applied to the other following figures. In Figure 8(b), the reports of the vehicles demonstrate that only A, B and C have heard the node M. Based on this reported information, the verifier V can determine that M is cheating about its position.



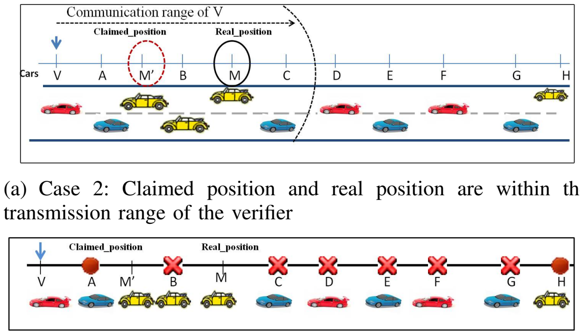
(a) Case 1: Claimed position is not under the transmission range of the verifier

(b) Results of vehicles' reports

(c) Legend

Fig. 8. Scenario 1 - Case 1

In Figure 9(a), we discuss the second case. In this case, the claimed position of M (position M') is in the communication range of the verifier V. Vehicles A, B, and C report their information about their neighbors (Figure 9(b)). These reports confirm that the considered vehicles (A, B, and C) received M's message, whereas vehicles D, E, F, G, and H did not hear it. Based on the collected information, the verifier node decides that the received information is consistent.



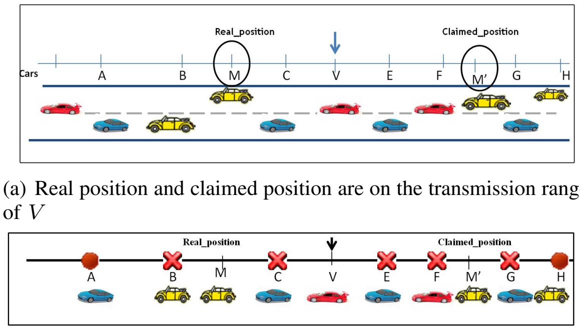
(a) Case 2: Claimed position and real position are within the transmission range of the verifier

(b) Results of vehicles' reports

Fig. 9. Scenario 1 - Case 2

Considering the third case, as depicted in Figure 10(a), the verifier V is located between the real position and the claimed position of M. The verifier, in the third case, could confirm that the reported information is consistent. Thus, M might be successfully cheating; yet, the impact of this false location does not lead to successfully modify the CW of V as the claimed position is between the positions of F and G, thus not affecting the computation of the maximum transmission range (Figure 10(b)).

We could summarize these cases in Figure 11, which represents V's CW as a function of the distance of the different neighbors. In fact, the CW is divided into three regions based on the different collected reports and position of vehicles. Let us consider the three regions denoted by $(R1)$, $(R2)$, and $(R3)$. Region $(R1)$ represents the regular CW values (represented by the continuous line) of the vehicle V without an attack.



(a) Real position and claimed position are on the transmission range of V

(b) Results of vehicles' reports

Fig. 10. Scenario 1 - Case 3

Furthermore, if M's claimed position exceeds C's position (for example the position M'_{R_2}), this information could have an effect on the waiting time of other vehicles (A, B, C); until the claimed position of the prover reaches the position of D (represented by the dotted line). If the claimed position of M exceeds D, in this situation the attack is detected. Region $(R2)$ is limited by the positions of C and D, then a claimed position of M in this Region $(R2)$ has a small effect on V's CW. Region $(R3)$ represents the position of vehicles superior than the position of D. Vehicles D, E, F, G, and H are in Region $(R3)$. If a malicious vehicle claims to be in Region $(R3)$ (for example M'_{R_3} in Figure 11), it will be detected by V because most of the vehicles in this region report the non overhearing of M. Thus, the verifier V will detect M as a cheater.

To summarize, if the vehicle M claims a position in Region $(R1)$, the verifier V found that: (i) the reported information is consistent, (ii) M could be cheating, and (iii) even if it cheats, there is no effect on V. In the second case, we consider that vehicle M claims a position in the Region $(R2)$. Thus, the verifier V proceeds to the following assumptions: (i) the reports of the observed nodes are consistent, (ii) V could not detect whether M is cheating or not, and (iii) the claimed position of M, situated between the positions of C and D, changes the diagram and has an effect on V. In Region (R3) in Figure 11, the prover M claims a position located in this region. In this case, V decides that M is cheating based on the received reports of vehicles.

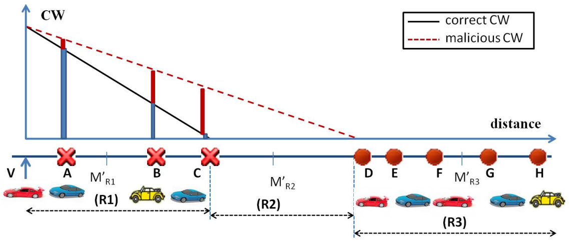


Fig. 11. Scenario 1: Verifier waiting time versus distance

2) Scenario 2: In this attack scenario, we consider that the malicious vehicle could be at most either M or one of the reporting vehicles. Let us focus on Figure 12. We split the regions into four parts: $(R1)$, $(R2)$, $(R3)$ and $(R4)$. $(R1)$ includes the vehicles A, B, and C that they hear the message, and could be in the transmission range of the verifier V. $R1$ represents the regular CW values of the vehicle V without an attack (represented by the continuous line). The region $(R2)$ represents the region between the last vehicle that heard the message and the first vehicle that did not hear the message. Region $(R3)$ is the area including the first vehicle that did not hear the message, and the second vehicle that did not hear also the message. Region $(R4)$ is the area including the vehicles that did not hear the message.

We consider that one of the reporting vehicles (B) is in Region $(R1)$. The reporter (B) claims that it does not hear M, however it should normally heard the message. Thus, the verifier V could detect B as the only cheating vehicle because we assumed that there is one malicious node. A vehicle located in Region $(R2)$ (for example the vehicle D) declares that it does not hear M. In this case, neither M nor the reporting vehicle (D) are detected as malicious. Vehicle V could not determine whether one of them is cheating or not. If the reporting vehicle is located in Region $(R3)$, it does not hear the vehicle M. Thus, the verifier V considered that one of the two vehicles is cheating either M or the reporting vehicle. In another situation, let us consider that the reporting vehicle (G) is located in Region $(R4)$, it transmits a report indicating that it does not hear M. Thus, the verifier V considers that M is the malicious vehicle.

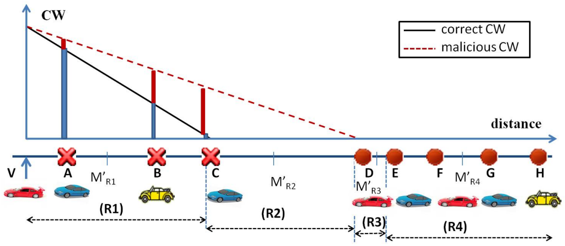
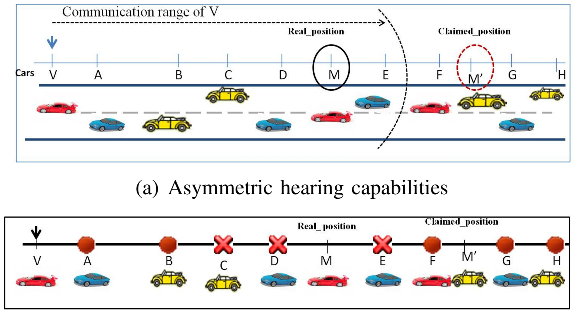


Fig. 12. Scenario 2: Verifying waiting time versus distance

3) Scenario 3: We consider in this scenario asymmetric overhearing capabilities: even if a vehicle V can receive messages from M, the reverse could not be true. In this case, we consider Figure 13(a) in which V hears M, and M does not hear V. In Figure 13(b), we represent the reports of vehicles about their ability to receive messages from M. In fact, A, B, F, G, and H do not hear M. Moreover C, D, and E have heard the message of the prover M. The hearing capabilities of the vehicles are not the same. M does not hear V and also A and B do not hear M's message.

Vehicle V decides whether M is a cheater or not based on the received claimed position in the three regions (Figure 14). In fact, if the prover M claims to be in Region $(R3)$, its claimed position (M'_{R_3}) will be detected as false. If M claims to be in Region $(R2)$, the announcement of this location (M'_{R_2}) has an effect on the diagram of the verifier. However, M could not be detected as malicious. Finally we also consider the case where M claims to be in Region $(R1)$ (for example in position M'_{R_1}). Thus, V could not detect M as cheating node, but the announcement of the location of M has no negative impact on the waiting time of V.



(a) Asymmetric hearing capabilities

(b) Results of vehicles' reports

Fig. 13. Scenario 3

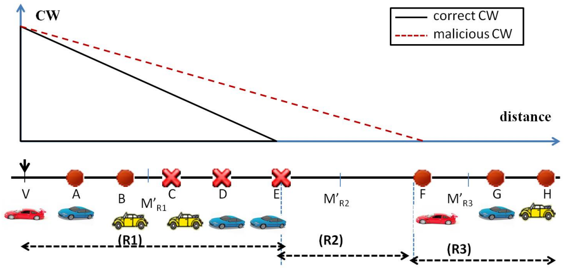


Fig. 14. Verifier waiting time versus distance

C. Description of the proposed cheating position detection

In this section, we describe our proposed position verification scheme. Our solution requires no infrastructure but only distributed messages exchanged between nodes to detect the malicious nodes. In fact, we discuss how the information of the vehicles could be propagated to other vehicles, in order to have a complete and a local observation. First, we discuss the structure of the transmitted message, and the timing of forwarding or transmitting the information. Second, these data could be collected from direct neighbors (in case nodes communicate directly) or from multi-hop neighbors (in case of indirect or asymmetric communication) between vehicles. Yet, we should limit the multi-hop propagation of this information within a limited region. Collected information should be recent and limited to the participating nodes. Third, in order to guarantee the authenticity of messages, nodes proceed to an authentication mechanism. To do this, we propose to transmit the information of vehicles in a modified $Hello$ message. We present the sending and receiving procedure of $Hello$ message. After receiving the different reports (the modified $Hello$ messages) from vehicles, each verifier executes locally a position verification procedure. Then, the verifier vehicle could detect whether the claimed position of a vehicle M is false or not.

1) Structure of the modified $Hello$ message: In order to have the position of the vehicles and their neighbors, we propose to include these data into the $Hello$ message instead of using additional structures. The first reason supporting this choice is that the $Hello$ message is transmitted by a vehicle in each time interval (100 ms). A second reason is that the sender of the $Hello$ message is chosen randomly. A third reason is to reduce the communication overhead and not generate another structure or another type of message. The modified $Hello$ message includes the $vehicle\_id$, the $vehicle\_position$, $declared\_max\_range$, $timestamp$, $list\_neighbors$ which is the list of two hop neighbors, and a signature generated by the sender private key. We refer the reader to Figure 15, in which we present an element (neighbor) in the list of neighbors. This list is constructed by a set of elements of type Neighbor. Then, we clarify in each element of type Neighbor its $vehicle\_id$, $vehicle\_position$, and a list of indirect neighbors. In fact, a list of indirect neighbors is composed by a set of elements of type Indirect Neighbor. Each Indirect Neighbor (see Figure 16) is a structure which includes its $vehicle\_id$, $vehicle\_position$, $declared\_max\_range$, and $timestamp$. Moreover, the packet in Figure 17 contains a signature of this information by the sender private key. The Figure 17 illustrates the structure of the modified $Hello$ message.

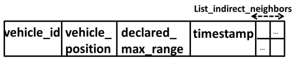


Fig. 15. Neighbor structure

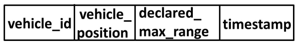


Fig. 16. Indirect neighbors structure

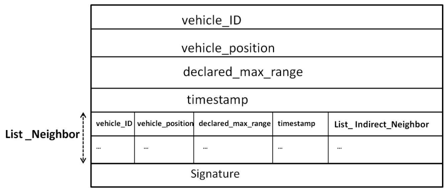


Fig. 17. A modified structure of $Hello$ message

2) Sending $Hello$ message procedure: We focus on the $Hello$ message sending procedure (Algorithm 4). In every turn, each vehicle determines a random waiting time (lines 1 and 2). After this waiting time, if neither other transmission is heard nor collision happened (line 6), it proceeds with transmitting a $Hello$ message. This $Hello$ message includes $vehicle\_ID$ (line 7), the $timestamp$ (line 9), the $vehicle\_position$ (line 8), the $declared\_max\_range$ (line 10), the list of neighbors and their two hop neighbors: $list\_neigh\_S$ (line 11). Furthermore, the sender uses its private key to generate a signature to the message (line 12), then it transmits the message (line 13).

Algorithm 4: Sending Hello message algorithm (executed by vehicle X)

```
1  Input: $list\_neighbors$: list of neighbors of $V$,
   $vehicle\_X$: the identity of the sender $X$,
   $position\_X$: the sender position,
   $current\_Time\_X$: current time of the sender,
   $LMFR$, $CMFR$, $list\_neigh\_X$: the list of neighbors of $X$,
   $K^X_{private}$: private key of the sender $S$,
   $H$: hash function;
2  Output: $Hello$ message ;
3  For each turn ;
4  $sending\_time := random(turn\_size)$;
5  wait ($sending\_time$);
6  if not ($heard\_Hello\_msg()$ or $heard\_collision()$) then
7      $Hello\_msg.vehicle\_ID$:= $vehicle\_X$;
8      $Hello\_msg.vehicle\_position$:= $position\_X$;
9      $Hello\_msg.timestamp$:= $current\_Time\_X$;
10     $Hello\_msg.declared\_max\_range$:= $max(LMFR, CMFR)$;
11     $Hello\_msg.list\_neighbor$:= $list\_neigh\_X$;
12     $Hello\_msg.signature$:= $K^X_{private}$
       (H($Hello\_msg.vehicle\_ID$, $Hello\_msg.vehicle\_position$,
       $Hello\_msg.timestamp$,
       $Hello\_msg.declared\_max\_range$,
       $Hello\_msg.list\_neighbor$)) ;
13     transmit ($Hello\_msg$);
14 EndFor
```

3) Receiving $Hello$ message procedure: The $Hello$ message receiving procedure is depicted in Algorithm 5. In particular, a vehicle receiving a $Hello$ message in line 2, generates the public key (line 3) using $f(sender\_id\_X)$ where f is a hash function. In line 4 of Algorithm 5, the receiver verifies the signature of $Hello$ message. Then, it checks for the freshness of message (line 6). Indeed, it could be an old message transmitted to the vehicles. This check is performed verifying the coherence between the time inserted in the message by the claiming vehicle (the sender of the $Hello$ message) and the current time of the receiver. Then, for each message that passes the previous checks, the receiver vehicle extracts the information ($sender\_id\_X$). The receiver checks whether this is the first received $Hello$ message carrying the $sender\_id\_X$. Then, the receiver simply stores ($sender\_id\_X, sender\_position\_X, declared\_max\_range$, $timestamp, list\_neighbor\_X$) to its list of neighbors (algorithm algoReceivinghello, line 9). Then, the receiver determines its own position (line 12), extracts from the $Hello$ message the $sender\_position$ (line 12), and the included estimation of the maximum transmission range (line 11), and determines the distance between itself and the sender (line 13). If the $Hello$ message is received from ahead, the value of $CMFR$ is updated (lines 14 and 15), otherwise $CMBR$ is updated (lines 16 and 17). In both cases, the new value is obtained as the maximum among the old one, the distance between the considered vehicle and the $Hello$ message sender, and the sender's transmission range estimation provided by the $Hello$ message.

4) Position Verification Procedure: After receiving reports ($Hello$ messages) from other vehicles, a vehicle V could decide whether the claimed position announced by a vehicle is correct or not. In Algorithm 6, we present the position verification algorithm executed by a vehicle V. In fact, V collects N reports (Algorithm 6, line 3). For each received report, it checks whether the claimed vehicle M is in the list

Algorithm 5: Receiving Hello message algorithm (executed by vehicle V)

```
Input: $list\_neighbors$: list of neighbors of $V$,
  $current\_Time\_V$: the current time of $V$,
  $sender\_id\_X$: the identity of the sender,
  $sender\_position\_X$: the field corresponding to sender position,
  $currentTime\_X$: current time of the sender included in the message,
  $drm\_X$: the declared maximum range received, $list\_neigh\_X$: the
  list of neighbors in the received message,
  $signedHelloMsg\_X$: the received signature ;
$< sender\_id\_X, sender\_position\_X, currentTime\_X, drm\_X,$
  $list\_neigh\_X, signedHelloMsg\_X >$ ;
$K_X^{Pub} \leftarrow f(sender\_id\_X)$;
if $H(sender\_id\_X, sender\_position\_X, currentTime\_X,$
  $drm\_X, list\_neigh\_X) \neq K_X^{Pub}(signedHelloMsg\_X)$ then
    handle this exception ;
if IsNotCoherent $(current\_time\_X, current\_time\_V)$ then
    handle this exception ;
if IsNotPresent $(list\_neighbors, sender\_id\_X)$ then
    add($list\_neighbors, < sender\_id\_X, sender\_position\_X,$
    $currentTime\_X, drm\_X, list\_neigh\_X, signedHelloMsg\_X >$)
    ;
$mp := my\_position()$ ;
$sp := sender\_position$ ;
$drm := declared\_max\_range$ ;
$d := distance(mp, sp)$ ;
if $(received\_from\_front(Hello\_msg))$ then
    $CMFR := max(CMFR, d, drm)$ ;
else
    $CMBR := max(CMBR, d, drm)$ ;
```

Algorithm 6: Position verification algorithm (executed by vehicle V)

```
1  Input: $V$: the verifier node executes the verification algorithm
     $M$: the vehicle for which $V$ wants to verify its claimed position
     $M_1$,...$M_N$: $N$ messages collected by $V$
     $Claimed\_position$ of $M$
     $M_i =< sender\_i, sender\_position\_i, timestamp\_X,$
     $drm\_X, list\_neigh\_i >$;
2  Output: State of $M$ is malicious or suspicious;
3  //i is the sender of the report $M_i$
     For all $i \in 1, ...Ndo$
     extract $(list\_neigh\_i)$ from the report $M_i$;
4  if $M \in list\_neigh\_i$ then
5      $i$ hears $M$ ;
6  else
7      $i$ does not hear $M$ ;
8  End For.
     $V$ sets its CW with respect to the different information;
9  if claimed_position $\in$ Region $(R1)$ of $V$ then
10     $M$ is not detected as malicious and the claiming distance has no
       effect on $V$;
11 else
12     if claimed_position $\in$ Region $(R2)$ of $V$ then
13         $M$ is not detected and has an effect on $V$ ;
14     else
15         if claimed_position $\in$ Region $(R3)$ of $V$ then
16             $M$ is detected as malicious;
```

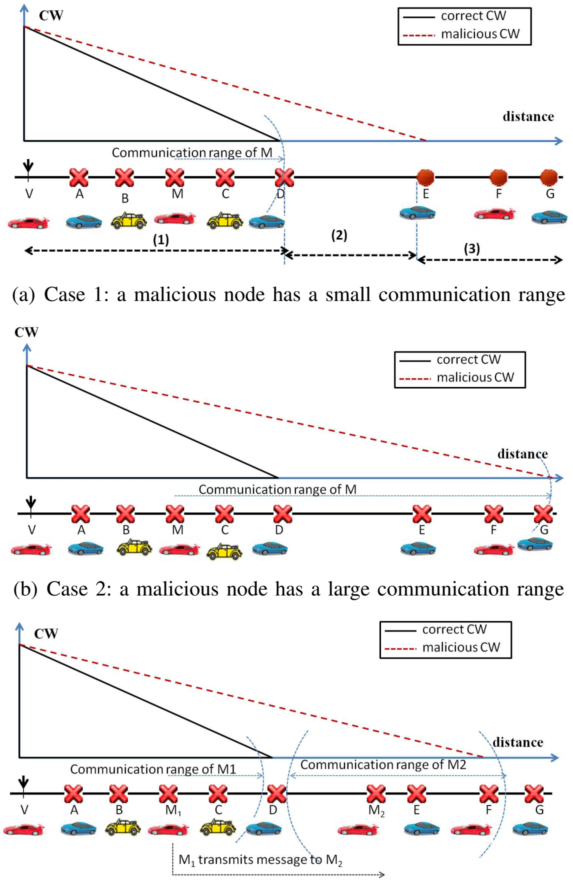
of neighbors of these vehicles. Based on this information, V could decide if the claimed position is in a certain Region. If the claimed position is in Region ($R1$) (line 9), then the vehicle could be a malicious one, but this has no negative effect on V. Otherwise, if the claimed position is in Region ($R2$) (line 12), then the position of M has an effect on V, and M is classified as suspicious. Finally, if the claimed position is in Region ($R3$) (line 15), M is detected as a cheater. If there is at most one malicious node, we refer to the case 1 of scenario 1. If there is one malicious vehicle in the road (M or one of the reporting vehicle), then we refer to the scenario 2.

5) The utility of the timestamp and the vehicle position in the modified $Hello$ message: In this section, we present the utility of some received transmitted information and their role in preventing the propagation of the adversary's message. We consider the scenario where two honest vehicles A and G are distant from each other. We consider that we have the following scenario depicted in Figure 18(a). If M is a malicious vehicle and has a communication range as presented in Figure 18(a), then it has no effect in modifying the list of neighbors of other vehicles. But, if the communication range of M is as presented in Figure 18(b) then E, F, and G will hear M. Thus, the diagram of V changes, and E, F, and G indicate that M is their neighbor. We could also consider the case where there are two malicious nodes collaborating together (Figure 18(c)). Vehicle M_1 is a malicious vehicle located near A, whereas M_2 is a malicious vehicle located near G. A sends a $Hello$ message signed by its private key. M_1 takes the same message and forwards it to M_2. Then, M_2 forwards it to G. When receiving this message, G will hear the message of A and think that A is its neighbor. Then, E and F suppose that M_1 is their neighbor, and thus the diagram of V changes. This scenario could be used by malicious nodes collaborating together, in order to influence vehicles that they are neighbors of one malicious vehicle.

In order to detect this attack or to limit its impact, a vehicle can check the sending time of the message. When receiving the message, if it was sent at a late time with respect to the receiving time, then the vehicle could detect it. Even if the message of the malicious node has propagated to some vehicles, the verification of the current time and the time included in the message could prevent the forwarding of this message. In fact, based on time, the vehicle can limit the propagation of the message (the malicious vehicle tries to forward it, in order to convince other vehicles that they are neighbors of V). At a certain time, a vehicle can detect that it is a late message (because forwarding takes time). The vehicle could use also the position information of the sender of the message. Thus, if the position of the sender is too distant from the receiver, then the received information is not consistent and the verifier can detect the malicious vehicle.

VIII. Solution to attack #2: anti-replay protection

In this section, we present how an honest vehicle could detect a non cooperation attack or a replay message attack presented in section V. Our proposed solution is based on appending a $timestamp$ to the broadcast message, and storing



(a) Case 1: a malicious node has a small communication range

(b) Case 2: a malicious node has a large communication range

(c) Case 3: two malicious nodes collaborating together

Fig. 18. Falsing list neighbors (a), (b), (c)

a table in each vehicle in order to protect against this attack. This storing table contains a list of items. Each item contains the identity of the sender or the forwarder of the broadcast message, the timestamp, and the broadcast message.

A. State of the art solutions of anti-replay protection

Anti-replay protection allows a receiving node to identify replayed messages and discard them to guarantee of weak or sequential freshness. IPSec, for example, uses an incrementing counter included with each packet to ensure sequential freshness [38]. IPSec's sliding window allows for out-of-order packet arrival; as long as packets arrive in order, or within the sliding window, they are accepted regardless of the time interval between packets. A more strict guarantee is strong freshness, which ensures that a message was sent within a certain, usually short, preceding time period. Strong freshness can be provided in two ways: either via a challenge or response sequence, usually involving a nonce value, or using a timestamp that is added to the packet during transmission and compared by the receiver with the global clock. All of these anti-replay techniques require traffic authentication so that counter values and timestamps cannot be modified without detection.

One of the proposed anti-replay mechanism is included in the SPINS protocol for wireless sensor networks [39]. In this mechanism, weak freshness is a byproduct of CTR-mode encryption. A monotonically increasing counter is used as a nonce value by encrypting outgoing packets, and this counter is also used as an anti-replay counter.

Timestamps are a practical alternative for anti-replay support in VANETs. Timestamps require that nodes maintain fine grained synchronization, and that all nodes synchronize with a global clock. Synchronization in vehicular communication is not a problem as vehicles could use the GPS clock information. Thus, in our approach to prevent this kind of replay attacks, we use timestamps.

B. Overview of our proposed solution

Let us assume the malicious node received a broadcast message, and then transmitted this message, without waiting for the expiration of any contention avoidance delay. In this case, the nodes receiving this broadcast message from their front restart the broadcast procedure. At the same time, nodes receiving the message from their back abort their message forwarding procedure, since the message has been already propagated over them along the direction of propagation.

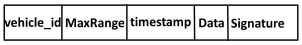
In order to detect this malicious behavior, each sender of the broadcast message should indicate in the transmitted broadcast message a $timestamp$. Then, the broadcast message includes a $vehicle\_id$, a $MaxRange$, a $timestamp$ (indicating the current forwarding time). Furthermore, each vehicle maintains a stored table in which it indicates in each line of the table the following fields: $vehicle\_id$, a $timestamp$, and a $broadcast$ message. Using these stored information, the vehicle could detect if it received the same message from its front vehicles, in a short period. The message will not be taken into consideration as it is a replayed message.

C. Description

In this section, we provide a detailed description of our proposed solution to detect malicious vehicles based on message exchange. To this aim, we discuss the structure of the broadcast message and the need for a stored table to detect replayed messages. Furthermore, in order to guarantee the authenticity of messages, nodes proceed to an authentication mechanism. To do this, we propose to transmit the timestamp in the broadcast message, and to store in each vehicle a table containing the last previous transmitted broadcast messages. We present the sending and the receiving procedure of a broadcast message. After receiving the broadcast message from a forwarder vehicle, a receiver of the message executes locally a verification procedure. Then, this vehicle could detect whether the transmitted message is a replayed message or not. The verifier vehicle uses the information stored in its table and the forwarding time to determine if it is a replayed broadcast message or not.

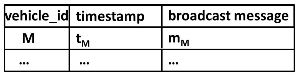
1) Broadcast message and Stored Table: Figure 19 and Figure 20 describe the structure of the broadcast message, as well as the stored table in each vehicle.

When receiving a broadcast message, the vehicle should verify whether it is a fresh message or not. In the latter case, it drops the message. Therefore, the structure of the modified broadcast message includes a $timestamp$ added by the sender.



vehicle_id	MaxRange	timestamp	Data	Signature

Fig. 19. Modified broadcast message



vehicle_id	timestamp	broadcast message
M	t_M	m_M
…	…	…

Fig. 20. Stored Table

2) Sending broadcast message procedure: A vehicle sending the broadcast message proceeds as follows. With the help of Algorithm 7, we explain our scheme's behavior during the procedure for sending broadcast messages. Each vehicle proceeds with transmitting a broadcast message; this broadcast message includes the $vehicle\_ID$ (line 2), the $MaxRange$ (line 3), the $timestamp$ (line 4), and the $data$ (line 5). Furthermore, the sender uses its private key to generate a signature for the message (line 6), before broadcasting it (line 7).

Algorithm 7: Sending broadcast Message algorithm (executed by a vehicle X)

```
1  Input: $broadcast\ msg$ : the broadcast message;
   $vehicle\_Id\_X$: vehicle ID of the sender $X$;
   $MaxRange\_X$: MaxRange of vehicle $X$;
   $timestamp\_X$: timestamp or current time of vehicle $X$;
   $data\_X$: generated data;
   $Kpriv\_X$: private key of sender $X$ ;
2  $vehicle\_ID \leftarrow vehicle\_Id\_X$;
3  $MaxRange \leftarrow MaxRange\_X$;
4  $timestamp \leftarrow timestamp\_X$;
5  $data \leftarrow data\_X$;
6  $signature \leftarrow$
   $\{vehicle\_Id\_X, MaxRange\_X, timestamp\_X, data\_X\}$
   $\_Kpriv\_X$ ;
7  $broadcast\ msg \leftarrow$
   $\langle vehicle\_ID, MaxRange, timestamp, data, signature\rangle$;
```

3) Receiving broadcast message procedure: The broadcast message receiving procedure is presented in Algorithm 8. In particular, a vehicle receiving a broadcast message (line 2), checks for the freshness of message (line 4). This check is performed by verifying the coherence between the timestamp, included in the message, by the claiming vehicle (the sender of the broadcast message) and the receiver's current time. In line 7 of Algorithm 8, the receiver verifies the signature of the broadcast message. Then, it checks whether the received broadcast message had already been transmitted before, using its stored table ($verified\_stored\_table()$). If the message is received for the first time, then the vehicle simply stores ($vehicle\_id\_X$, $timestamp\_X$, $broadcast\ msg$) in its stored table which is the role of $add\_message\_to\_stored\_table()$ (line 10). Otherwise, the $broadcast\ msg$ is a replayed message, and thus it will be dropped (line 12).

IX. Solution to Attack #3: Interrupting forwarding Attack Detection

To detect malicious vehicles attempting to jeopardize FMBA by impeding the propagation of an alert message, we propose a mechanism that detects when vehicles along the road have not received the message. Our solution requires a vehicle acting as a verifier (as described in Section IX-B and IX-C). Each forwarder vehicle should give a proof of relaying the broadcast message. Please note that we do not need any specific mechanism to elect the verifier, since any node can act as such (in fact, messages for the verifier are not intended for any specific receiver). We can just assume that any node in the area will also act as verifier. Of course a verifier which is a malicious node would not report any problem, even if an attack is detected. However, to detect the attack and take further actions (e.g., rebroadcast the blocked message) it is enough that at least one non compromised verifier detects the attack. In the following, we present an overview of the security scheme as well as a description of it.

Algorithm 8: Receiving broadcast message algorithm (executed by a vehicle V)

```
1  Input: $broadcast\ msg$: the received broadcast message;
   $f$: hash function;
2  $broadcast\ msg$ =
   $\langle vehicle\_Id\_X, \_, timestamp\_X, Data\_X, Signature\_X\rangle$;
3  $K_{pub\_X} = f(vehicle\_Id\_X)$ ;
4  if not (verified_timestamp(current_time, timestamp_X)) then
5      $V$ drops the message;
6  else
7  if verified_signature() then
8      if $verified\_stored\_table()$ then
9          $accept\_message()$;
10         $add\_message\_to\_stored\_table()$ ;
11     else
12         drop packet;
13         handle exception();
14 else
15     drop packet;
16     handle exception();
```

A. State of the art solutions of detecting misbehaving forwarders

Malicious vehicles are compromised vehicles that are willing to put an effort to introduce some damage. Hence malicious nodes are different from selfish ones, since selfish just do not want to use their resources for the sake of protocol's success. Many studies enhance the cooperation between vehicles and incentive them to forward messages. Major contributions in incentive cooperation in multi-hop communications use either reputation based schemes ([40], [41], and [42]) or credit based schemes [43], or hybrid schemes [44]. We underline that the type of attack that we described, in section VI, deals with malicious behavior of nodes, not selfishness. In fact, the malicious node tries to transmit packets only frontward and not backward, in order to stop the propagation of the message. Many efforts ([40], [41], and [42]) have been done to detect misbehaving nodes that do not forward the packets. We classify these works on reputation based schemes and credit based schemes.

1) Reputation based mechanisms: The reputation of a node increases when it carries out correctly the task of forwarding the packets. The main problem of the reputation based mechanisms is that they work in the inefficient promiscuous mode [45]. Once a node's reputation degrades to a threshold, the node is identified as dishonest or selfish. In [46], a watchdog and a path-rater are implemented in each node. The watchdog overhears the medium to detect whether the next-hop node forwards the packets or not. The path-rater module chooses the path that avoids selfish nodes based on the watchdog's notifications. The major disadvantage of this scheme is that it incurs extra loads on the honest nodes. In [47], it is shown that the reputation based mechanisms punish the selfish nodes at the expense of decreasing the throughput of cooperative ones.

2) Credit based mechanisms: In the credit based mechanisms, the cooperative nodes earn credits in order to relay packets generated from other nodes, and spend them to relay packets. In Nuglets [48], a tamper proof device is installed on each node, thus it allows nodes to store their credits and secure their operations. However, this is not secure and realistic because they assume that the device can not be tampered. In [49], the authors propose SIP, in which the destination node sends a payment receipt to the sender which issues a REWARD packet. The REWARD packet increments the credit counters stored in the intermediate nodes. The disadvantage of this scheme is that each packet needs three trips between the source and destination nodes.

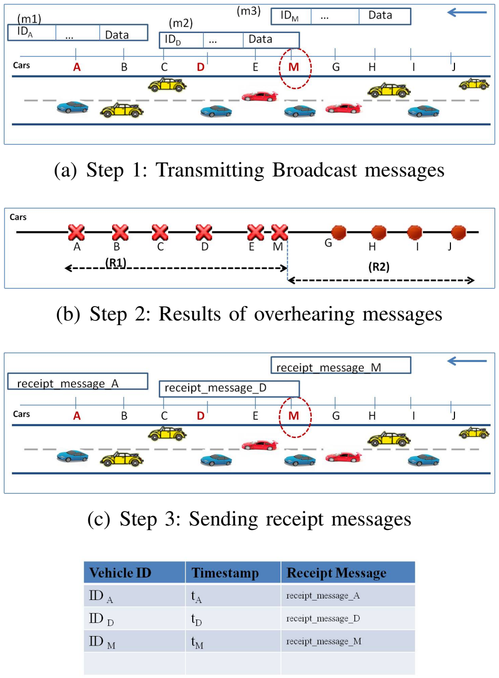
The credit based schemes proposed in the literature, to detect misbehaving or selfish nodes could not be applied to our specific problem, since malicious nodes could forward only on one direction (frontward) and not backward. The goal of the compromised nodes is to stop the propagation of the alert message, thus it could induce damages and accidents. The approach we propose, is based on the fact that each forwarder node has to sent a proof of receipt to a verifier node. The receipt message includes information that helps the verifier node to detect the misbehavior of the node.

B. Overview

The kind of attacks that we described, in section VIII, deals with malicious behavior of nodes, not selfishness. In fact, the malicious node tries to transmit packets only frontward and not backward (by adjusting its transmission range). The aim of this malicious vehicle is to stop the propagation of the message. To prevent this attack, neighbor nodes have to collaborate together to confirm whether this node is misleading or not. Therefore, we have to define a strategy able to confirm that backward neighbors have not received the message. To reclaim this, each forwarder vehicle should transmit a receipt message to a verifier node, indicating that its message was relayed. The verification will be done by a verifier node able to use its knowledge of the vehicles on the road, to decide whether there are some malicious vehicles, that are trying to impede the correct propagation of the broadcast message.

In the following, we discuss the structure of the receipt message, as well as the transmission of this message by the forwarder nodes, and the verification operation done by the verifier node. The attack is more specific to $FMBA$, and different from what was studied in the literature [43], [44].

To simplify our presentation, we refer the reader to Figure 21(a) in which we have the vehicle A sending a broadcast message (represented by $m1$), which forwards this message to the forwarder vehicle D. Then, D transmits the broadcast message ($m2$) to the receivers in its communication range. As a consequence, M is chosen as a forwarder when applying the forwarding procedure in FMBA. Unfortunately, M is a malicious vehicle and thanks to a directional antenna tries to send messages only frontward (broadcast message ($m3$)). The goal of M is to stop the propagation of the broadcast message, by blocking vehicles in the front from forwarding the alert message, while backward vehicles (G, H, I, and J) will never receive the alert message (see Figure 21(b)).



(a) Step 1: Transmitting Broadcast messages

(b) Step 2: Results of overhearing messages

(c) Step 3: Sending receipt messages

Vehicle ID	Timestamp	Receipt Message
ID $_A$	t_A	receipt_message_A
ID $_D$	t_D	receipt_message_D
ID $_M$	t_M	receipt_message_M

(d) Step 4: Verifier process

Fig. 21. Example of non propagation of broadcast message detection (a), (b), (c), (d)

C. Description

In this section, we detail the transmission of a receipt message by a forwarder vehicle, as well as the verification performed by the verifier node.

1) Receipt message: In order to send a proof of packet relay, the forwarder vehicle has to create a receipt message. In fact, forwarders contact the verifier node at least once during each time interval to send their receipts. After forwarding a broadcast message, the forwarder sends to the verifier node a receipt message containing the $vehicle\_id$, the $vehicle\_position$, the $timestamp$, and a $signature$ generated by the forwarder private key. We refer the reader to Figure 21(c), in which we present an example of receipt messages generated by a forwarder. The verifier node collects authenticated receipts (see Figure 21(d)), and performs some verifications to detect possible malicious forwarders trying to stop the propagation of the broadcast message.

2) Sending receipt message procedure executed by a forwarder vehicle: A vehicle sending the receipt message proceeds as described by Algorithm 9. Focusing on the receipt message sending procedure, each forwarder transmits a receipt message after relaying a broadcast message. This receipt message includes the $vehicle\_ID$ (line 5) and the $timestamp$ (line 6). Furthermore, the sender uses its private key to generate a signature to the message (line 9) before transmitting it (line 10).

Algorithm 9: Sending receipt message algorithm (executed by a vehicle X)

```
1  Input: $receipt\ msg$ : the receipt message;
   $A \rightarrow *$: $broadcast$ msg;
2  $vehicle\_id\_X$: the vehicle Id of $X$;
3  $timestamp\_X$: the timestamp of $X$;
4  $K\_priv\_X$: the private key of $X$;
5  $vehicle\_ID \leftarrow vehicle\_id\_X$ ;
6  $timestamp \leftarrow timestamp\_X$ ;
7  $signature \leftarrow \{vehicle\_ID, timestamp\}\ \_Kpriv\_X$ ;
8  //generating the receipt message
9  $receipt\_message \leftarrow \langle vehicle\_ID, timestamp, signature \rangle$ ;
10 $transmit\ (vehicle\_ID, timestamp, signature)$;
```

3) Receiving receipt message procedure executed by a verifier node: The receipt message receiving procedure executed by a verifier node is depicted in Algorithm 10. In particular, a vehicle receiving a broadcast message (line 2) generates the public key (line 6) using $f(vehicle\_id\_X)$ with f is a hash function. In line 10 of Algorithm 10, the receiver verifies the signature of the receipt message; then, for each message that passes the previous checks, the receiver vehicle extracts the information ($vehicle\_id\_X$) and ($timestamp\_X$), and stores ($vehicle\_id\_X, timestamp\_X, signature\_X$) (line 13). At this point, the verifier node can verify using the different collected receipts whether there is a propagation of the broadcast message or not, since it knows the position of the different vehicles. If the information carried by the last forwarder vehicle is not consistent with the dispersion of vehicles on the road, this means there are neighbors that have not received the broadcast message. In this case, the verifier can detect that one of the forwarder vehicles is malicious, as it does not forward packets backward.

X. FS-MBA

In this section, we present FS-MBA, i.e., a global view of the solutions in sections VII, VIII, and IX. In the remainder of this section, we provide a security overhead analysis for the FS-MBA algorithm (Section X.A), and we evaluate its performance under $Hello$ message loss (Section X.B). Furthermore, we study the behavior of FS-MBA under attacks run by more than one malicious vehicle (Section X.C).

Algorithm 10: Receiving receipt message algorithm (executed by a vehicle V)

```
1  Input: $receipt\ msg$: the receipt message of vehicle $X$;
   $vehicle\_id\_X$: the received vehicle Id ;
2  $timestamp\_X$: the received timestamp;
3  $signature\_X$: the received signature ;
4  $K\_pub$: public key;
5  $receipt\_message \leftarrow$
   $\langle vehicle\_ID\_X, timestamp\_X, signature\_X \rangle$ ;
6  $K\_pub \leftarrow f(vehicle\_id\_X)$ ;
7  if not (verified_timestamp(current_time, timestamp_X)) then
8      $V$ drops the message;
9  else
10 if verified_signature() then
11     if $verified\_consistency$ then
12         $accept\_message()$;
13         $Add\_message\_to\_stored\_table()$ ;
14     else
15         handle exception;
16 else
17     drop packet;
```

To make a global view of our secure solution, Algorithm 11 illustrates the merge of the algorithms presented before (to tackle different type of attacks).

Algorithm 11: FS-MBA

```
1  Input: $type\_message$: the type of the received message, i.e $Hello$
   message, or $broadcast$ message or $receipt$ message;
2  $msg$: the received message;
3  if ⟨ type_message== Hello message⟩ then
4      Receiving $Hello$ message algorithm (Algorithm 5) ;
5      Position verification algorithm (Algorithm 6);
6  else
7      if ⟨ type_message == broadcast message⟩ then
8          Receiving broadcast message algorithm (Algorithm 8);
9      else
10         Receiving receipt message algorithm (Algorithm 10);
```

In particular, upon receiving a message (Algorithm 11, line 2), a vehicle does the following checks. First, the receiver checks whether the message is a $Hello$ message (line 3). If this condition holds, the vehicle verifies the presence of the cheating position attack (line 5). If the received message is a $broadcast$ message (check done in line 7), the vehicle verifies whether there is a replay broadcast message attack. The verification is done by executing Algorithm 8. Finally, if the message is a $receipt$ (check done in line 9), the node verifies whether there is an interrupting forwarding attack (line 10), by running Algorithm 10.

A. Security Overhead

Security always comes at a price, which translates into processing, bandwidth and storage overhead. Securing FMBA also costs some additional communication and computation overhead, due to the generation and the verification of packet signatures. Moreover, vehicles send information (e.g., position) very frequently. Each $Hello$ message has to be signed, and each vehicle has to validate, (e.g., every 100 milliseconds),

$Hello$ messages from all neighboring vehicles in the communication range. In Figure 22, we summarize the communication, the computation and the storage overhead experienced by a vehicle X for each message. In Figure 22, we denote the size (in terms of bytes) of a field S by $\Vert S\Vert$. We denote by $sign$ the signature of the message, $sign\_gen\_op$ represents the signature generation operation, and $sign\_verif\_op$ is the signature verification operation. Let N_1 (N_2) be the number of direct (indirect) neighbors (see Section VII-C1) for a vehicle X. Let V_id be the vehicle identity, and V_pos indicates the vehicle position.

Overhead		
Communication	per Hello msg (send./rec.)	$\Vert timestamp\Vert$+$\Vert sign\Vert$ +N_1 (V_id+V_pos +drm +$\Vert timestamp\Vert$+ N_2($\Vert V\_id\Vert$+$\Vert V\_pos\Vert$ +$\Vert drm\Vert$+$\Vert timestamp\Vert$))
	per broadcast msg (send./rec.)	$\Vert timestamp\Vert$+$\Vert sign\Vert$
	per receipt msg (send./rec.)	$\Vert receipt\_msg\Vert$
Comp.	per each msg (send.)	$sign\_gen\_op$
	per each msg (rec.)	$sign\_verif\_op$
Storage	per Hello msg (send./rec.)	-
	per broadcast msg (send./rec.)	$\Vert broadcast\_msg\Vert$
	per receipt msg (send./rec.)	$\Vert receipt\_msg\Vert$

Fig. 22. FS-MBA Overhead

B. $Hello$ Message Loss

In this section, we evaluate the performance of our security solution FS-MBA, and the original FMBA under $Hello$ message loss. As a metric to evaluate the performance of our algorithms, we consider the average number of slots required to propagate a broadcast message. We measure the time variables in slots as done in [3]. We considered a scenario with a strip-shaped road and considered an area-of-interest of 2 km with a factual transmission range equal to 300 m. We simulated several scenarios, and for each of them we run 60 simulations and we average their outcomes to produce charts. We used 50 vehicles per kilometer, and their speeds were uniformly distributed in the range 72-144 Km/h. In the remainder of this section, we denote by p the probability of a $Hello$ message loss. Taking inspiration from the implementation and analysis done in [50], we have chosen three different loss probabilities (p = 10%, p = 25%, and p = 50%). To assess whether message loss has an impact, we run several simulations (see Figure 23). In our simulation, we considered two attacks for false position cheating. In Figure 23, *FMBA Attack* 1 (respectively *FS-MBA Attack* 1) refers to a position cheating attack claiming 1000 m when running FMBA (respectively FS-MBA) algorithm. *FMBA Attack* 2 (*FS-MBA Attack* 2) refers to a position cheating attack claiming 1500 m when running FMBA (respectively FS-MBA) algorithm. We evaluated FMBA and FS-MBA under the two attacks: FS-MBA performances are always better than FMBA.

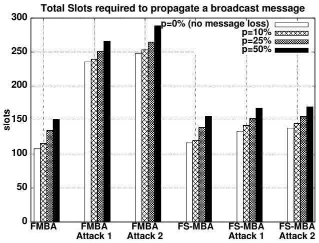


Fig. 23. Impact of Hello message loss

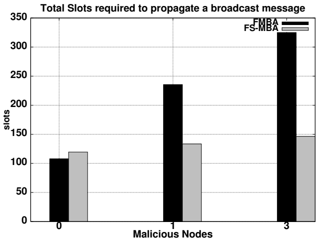


Fig. 24. Impact of malicious vehicles

Each protocol has been evaluated under different probability of losses (p = 0% i.e. with no message loss, p = 10%, p = 25%, and p = 50%). From the results in Figure 23, we can see that the performances get worst when the probability of $Hello$ message loss increases. The number of slots required to propagate a broadcast message is 107 slots for FMBA without message loss, and roughly 150 slots for FMBA with a probability of loss p = 50%. Furthermore, if we consider the case of FS-MBA attacked by a malicious vehicle claiming a position of 1000 m (*FS-MBA Attack* 1 in Figure 23), we see that the number of slots varies from 133 slots with p = 0% to 167 slots with p = 50%. However, it is interesting to note that these performances are not worse compared to FMBA under the two different attacks. If we consider *FMBA Attack* 2, the number of slots is still higher varying the probability of $Hello$ message loss. It is straightforward to note that the performances (in terms of the number of slots) of the two protocols with $Hello$ message loss is still reasonable compared to FMBA with attacks. These results show that FS-MBA is sound and resilient to message loss, which make it feasible under realistic conditions.

C. Impact of the number of malicious vehicles

In order to assess the impact of the number of malicious vehicles on degrading the performances of our solutions, we have done some simulations using three different scenarios. In the first scenario, we consider that there is no malicious vehicle in the road. In the second scenario, one malicious vehicle is cheating about its position. In the third scenario, three malicious nodes are randomly distributed on different hops. We concentrate on evaluating the performances of FMBA and FS-MBA under a position cheating attack of 1000 m. In particular, we consider the number of slots that a vehicle has to wait before forwading a broadcast message as our evaluation metric. This metric will allow us to compare the efficiency of the two protocols. Figure 24 reports the average number of slots required to propagate a broadcast message. In Figure 24, we see that the performances of FMBA with a position cheating attack degrades significantly when the number of malicious nodes increases. In fact, the number of slots is 235 for one malicious vehicle, and 325 for three malicious vehicles. This simulation demonstrates first that even with one malicious node, the performances of FMBA decreases, and the number of slots required to propagate a message increases very significantly (see Figure 24). Secondly, it is interesting to note that FS-MBA is resistant to the attack run by the three malicious vehicles. Hence, the number of slots is still much lower (146 slots) compared to FMBA attacked by one malicious vehicle.

Due to the space limit, we show the behavior of the protocols only for 50 vehicles per kilometer. We consider that this simulation could be applied to other different vehicles' density. Hence, we envision to do extensive simulations studying the position and the number of malicious nodes, and we consider this perspective as a future work.

XI. Conclusion

The main goal of Inter Vehicular Communications (IVC) consists in increasing people's safety by exchanging warning messages between vehicles. This survey paper scratches the surface of what is promising to be a new and fertile area of research in IVC security. We have elaborated on security issues in IVC considering a general class of applications based on multi-hop broadcast; yet, without loss of generality, we have chosen a representative case study for this class, FMBA, to concretely discuss issues and possible solutions. In this context, we have provided an overview of the different attacks and security weaknesses, also proposing possible countermeasures. In Figure 25 we provide a synthetic summary of the different attacks on the IVC based vehicular safety applications class, and in particular on the representative case study we chose for the discussion.

Attacks on FMBA	**State of the art solutions**	**Our proposed solution**	**Class of Attack [24]**
Position cheating	Infrastructure, Parameter, and Model based approaches.	Infrastructure-less and cooperative neighbors technique.	Cheating on positioning information; Denial of Service.
Replay broadcast message	selfish behavior, replay and duplicate message detections and preventions.	Storing the information of a broadcast message for a certain period to avoid its transmission by another node.	Denial of Service.
Interrupting-forwarding	Reputation or Credit based approaches.	Malicious node behavior detection based on receipt messages as a proof of forwarding alert messages.	Denial of Service.

Fig. 25. Overview of attacks

References

[1] M. L.Sichitiu and M. Kihl, "Inter-vehicle communication systems: a survey," *IEEE Communications Surveys & Tutorials*, vol. 10, no. 2, pp. 88–105, 2008.

[2] C. Wu and Y. Liu, "Queuing network Modeling of Driver Workload and Performance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 3, pp. 528–537, 2007.

[3] C. E. Palazzi, S. Ferretti, M. Roccetti, G. Pau, and M. Gerla, "How Do You Quickly Choreograph Inter-Vehicular Communications? A Fast Vehicle-to-Vehicle Multi-Hop Broadcast Algorithm, Explained," in *IEEE CCNC*, 2007.

[4] A. Amoroso, M. Ciaschini, and M. Roccetti, "The farther relay and oracle for VANET. preliminary results," in *WICON*, 2008.

[5] M. D. Felice, A. Ghandour, H. Hartail, and L. Bononi, "Enhancing the performance of safety applications in IEEE 802.11p/WAVE Vehicular Networks," in *IEEE WoWMoM*, Jun. 2012.

[6] C. E. Palazzi, M. Roccetti, and S. Ferretti, "An Intervehicular Communication Architecture for Safety and Entertainment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, pp. 90–99, Mar. 2010.

[7] J. Luo and J.-P. Hubaux, *A Survey of Research in Inter-Vehicle Communications*. Embedded Security in Cars –Securing Current and Future Automotive IT Applications, Springer-Verlag, 2005.

[8] G. Korkmaz, E. Ekici, F. Özgüner, and Ü. Özgüner, "Urban multi-hop broadcast protocols for inter-vehicle communication systems," in *1st ACM Workshop VANET*, 2007, pp. 76–85.

[9] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, "Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 584–616, 2011.

[10] F. Qu, F.-Y. Wang, and L. Yang, "Intelligent transportation spaces: Vehicles, Traffic, Communications, and BeyondLiuqing Yang," *IEEE Communications Magazine*, vol. 48, no. 11, pp. 136–142, Nov. 2010.

[11] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, "A survey of Inter-Vehicle Communication Protocols and Their Applications," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 2, 2009.

[12] A. Broggi, P. Cerri, S. Ghidoni, P. Grisleri, and H. G. Jung, "A New Approach to Urban Pedestrian Detection for Automatic Braking," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 4, pp. 594–605, 2009.

[13] M.-t. Sun, W.-c. Feng, K. Fujimura, T.-H. Lai, H. Okada, and K. Fujimura, "GPS-Based Message Broadcasting for Inter-vehicle Communication," in *ICPP'00*, vol. 6, 2000, pp. 2685–2692.

[14] M. Roccetti and G. Marfia, "Modeling and Experimenting with Vehicular Congestion for Distributed Advanced Traveler Information Systems," *Computer Performance Engineering Lecture Notes in Computer Science*, vol. 6432/2010, pp. 1–16, 2010.

[15] N. Ravi, S. Smaldone, L. Iftode, and M. Gerla, "Lane Reservation for Highways (Position Paper)," in *IEEE ITSC 2007*, 2007.

[16] C. E. Palazzi, M. Roccetti, S. Ferretti, G. Pau, and M. Gerla, "Online Games on Wheels: Fast Game Event Delivery in Vehicular Ad-hoc Networks," in *Proc. of IEEE V2VCOM 2007*, Jun. 2007.

[17] M. Fazio, C. E. Palazzi, S. Das, and M. Gerla, "Facilitating Real-time Applications in VANETs through Fast Address Auto-configuration," in *IEEE CCNC* . IEEE Communications Society, 2007.

[18] T.-S. Dao, K. Y. K. Leung, C. M. Clark, and J. P. Huissoon, "Markov-Based Lane Positioning Using Intervehicle Communication," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 4, pp. 641–650, Dec. 2007.

[19] P. Papadimitratos, L. Buttyan, T. Holcze, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J.-P. Hubaux, "Secure vehicular communication systems: design and architectures," *IEEE Communications Magazine*, vol. 46, no. 11, pp. 100–109, Nov. 2008.

[20] A. Weimerskirch, J. J. Haas, Y.-C. Hu, and K. P. Laberteaux, *VANET: Vehicular Applications and Inter-Networking Technologies Kenneth P Laberteaux*, h. hartenstein and k. p. laberteaux ed. Chichester, UK: John Wiley & Sons,, Nov. 2009, ch. 9 :Data Security in Vehicular Communication Networks.

[21] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer, "A Scalable Robust Authentication Protocol for Secure Vehicular Communications," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 1606–1617, May 2010.

[22] B. Mishra, Priyadarshini Nayak, S. Behera, and D. Jena, "Security in vehicular adhoc networks: a survey," in *ICCCS*, 2011.

[23] G. Calandriello, P. Papadimitratos, J. P. Hubaux, and A. Lioy, "On the Performance of Secure Vehicular Communication Systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, pp. 898–912, 2011.

[24] M. Raya and Jean-Pierre Hubaux, "The security of vehicular ad hoc networks," in *3rd ACM workshop on SASN*, 2005.

[25] A. Aijaz, B. Bochow, F. Dötzer, A. Festag, M. Gerlach, R. Kroh, and T. Leinmüller, "Attacks on Inter Vehicle Communication Systems - an Analysis," in *3rd International Workshop on Intelligent Transportation (WIT 2006)*, Mar. 2006.

[26] Q. Wu, J. Domingo-Ferrer, and Ú. González-Nicolás, "Balanced Trustworthiness, Safety and Privacy in Vehicle-to-Vehicle Communications,"

IEEE Transactions on Vehicular Technology, vol. 59, no. 2, pp. 559–573, Mar. 2010.

[27] W. Diffie and M. E. Hellman, “New directions in cryptography,” *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[28] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[29] T. Leinmüller, C. Maihöfer, E. Schoch, and F. Kargl, “Improved security in geographic ad hoc routing through autonomous position verification,” in *3rd international workshop on Vehicular ad hoc networks VANET*, 2006.

[30] J.-P. Hubaux, S. Capkun, and J. Luo, “The security and privacy of smart vehicles,” *IEEE Security & Privacy*, vol. 2, pp. 49–55, May 2004.

[31] N. Sastry, Umesh Shankar, and D. Wagner, “Secure verification of location claims,” in *Proc. of the 2nd ACM workshop on Wireless security*, 2003.

[32] G. Yan, S. Olariu, and M. C.Weigle, “Providing vanet security through active position detection,” *Journal of Computer Communications*, vol. 31, no. 2, pp. 2883–2897, Jul. 2008.

[33] A. Vora and M. Nesterenko, “Secure Location Verification Using Radio Broadcast,” *IEEE Transactions on Dependable and Secure Computing*, vol. 3, Oct. 2006.

[34] Z. Ren, W. Li, and Q. Yang, “Location Verification for VANETs Routing,” in *IEEE WIMOB*, Oct. 2009, pp. 141–146.

[35] T. Leinmüller, E. Schoch, and F. Kargl, “Position Verification Approaches for Vehicular Ad Hoc Networks,” *IEEE Wireless Communication Magazine*, vol. 13, no. 5, pp. 16–21, Oct. 2006.

[36] J.-H. Song, V. W. Wong, and V. C. Leung, “Secure Location Verification for Vehicular Ad-Hoc Networks,” in *IEEE GLOBECOM*, 2008, pp. 1–5.

[37] P. Golle, D. Greene, and J. Staddon, “Detecting and Correcting Malicious Data in VANETs,” in *ACM international workshop on Vehicular ad hoc networks*, 2004, pp. 29–37.

[38] S. Kent and R. Atkinson, “Security Architecture for the Internet Protocol,” *IETF RFC 2401*, Nov. 1998.

[39] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, “SPINS: Security protocols for sensor networks,” *Wireless Networks*, vol. 8, no. 55, pp. 521–534, Sep. 2002.

[40] S. Buchegger and J.-Y. L. Boudec, “Performance analysis of the CONFIDANT protocol,” in *ACM MobiHoc 2002*, Jun. 2002, pp. 226–236.

[41] P. Michiardi and R. Molva, “CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks,” in *Proc. of IFIP Communication and Mutimedia Security in Mobile Ad Hoc Networks*, Sep. 2002, pp. 107–121.

[42] Q. He, D. Wu, and P. Kholsa, “A secure incentive architecture for ad hoc networks: Research Articles,” *Journal in Wireless Communications and Mobile Computing - Wireless Network Security*, pp. 333–346, May 2006.

[43] A. Weyland, T. Staub, and T. Braun, “Comparison of motivation-based cooperation mechanisms for hybrid wireless networks,” *Journal of Computer Communications*, vol. 24, no. 13-14, pp. 2661–2670, Aug. 2006.

[44] M. E. Mahmoud and X. S. Shen, “Stimulating cooperation in multi-hop wireless networks using cheating detection system,” in *INFOCOM*, 2010, pp. 776–784.

[45] L. M. Feeney, “An Energy-Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad Hoc Networks,” *Mobile Networks and Applications*, vol. 3, no. 6, pp. 239–249, 2001.

[46] S. Marti, T. J. Giuli, K. Lai, and M. Baker, “Mitigating routing misbehavior in mobile ad hoc networks,” in *MobiCom*. ACM, 2000, pp. 255–265.

[47] J. J. Jaramillo and R. Srikant, “DARWIN: Distributed and Adaptive Reputation mechanism for WIreless ad-hoc Networks,” in *ACM MobiCom’ 07*, Sep. 2007, pp. 87–98.

[48] L. Buttyán and J.-P. Hubaux, “Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks,” *Mobile Networks and Applications*, vol. 8, no. 5, pp. 579–592, Oct. 2003.

[49] Y. Zhang, W. Lou, W. Liu, and Y. Fang, “A Secure Incentive Protocol for Mobile Ad Hoc Networks,” *Wireless Networks*, vol. 13, no. 5, pp. 569–582, Oct. 2007.

[50] G. Marfia, M. Roccetti, A. Amoroso, and G. Pau, “Safe Driving in LA: Report from the Greatest Intervehicular Accident Detection Test Ever,” *IEEE Transactions on Vehicular technology, IEEE Vehicular Technology Society*, vol. 62, no. 2, Feb. 2013.

Wafa Ben Jaballah is currently a PhD Student at the University of Bordeaux-1, France. Her main research interest is in the area of security in wireless sensor networks and vehicular communications. In 2012, she was a visiting student at the university of Padua, Italy.

Mauro Conti received the Ph.D. degree from Sapienza University of Rome, Italy, in 2009. In 2008, he was a Visiting Researcher at the Center for Secure Information Systems, George Mason University, Fairfax, VA, USA. After earning his Ph.D. degree, he was a Postdoctoral Researcher at Vrije Universiteit Amsterdam, The Netherlands. From 2011, he is an Assistant Professor at the University of Padua, Italy. In 2012, he was a Visiting Assistant Professor at the University of California, Irvine, CA, USA. His main research interest is in the area of security and privacy. In this area, he has published more than 40 papers in international peer-reviewed journals and conferences. Dr. Conti was a Panelist at ACM CODASPY 2011. He served as program committee member of several conferences, and he is General Chair for SecureComm 2012 and ACM SACMAT 2013. In 2012, he has been awarded by the European Commission with a Marie Curie Fellowship.

Mohamed Mosbah received hid Ph.D. degree from the University of Bordeaux 1, France, in 1993. He was an associate professor between 1994 and 2002. He is a full professor in computer science since 2003 at Polytechnic Institute of Bordeaux, France. His research interests include distributed algorithms and systems, formal models, security, and ad hoc and sensor networks. He participated to several national and European research projects, including collaborations with industry. He wrote more than 60 research papers published in international journals and conference proceedings. He is involved in various technical program committees and organisations of many international conferences.

Claudio E.Palazzi is an Associate Professor in Computer Science of the Department of Mathematics, University of Padua, Italy. He received his M.S. degree in Computer Science from UCLA in 2005, his Ph.D. degree in Computer Science from University of Bologna in 2006, and his Ph.D. degree in Computer Science from UCLA in 2007. From 2007 to 2010 he was an Assistant Professor at the Department of Pure and Applied Mathematics of the University of Padua. His research is primarily focused on protocol design and analysis for wired/wireless networks, with emphasis on network-centric multimedia entertainment and vehicular networks. On these topics, he is active in various technical program committees in prominent international conferences and is co-author of more than 90 papers, published in international conference proceedings, books, and journals.