Secure Verification of Location Claims on a Vehicular Safety Application

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

***Abstract*—Traffic safety through inter-vehicular communication is one of the most promising and challenging applications of Vehicular Ad-hoc Networks. In this context, information such as position, direction, and speed, is often broadcast by vehicles so as to facilitate fast multi-hop propagation of possible alert messages. Unfortunately, a malicious vehicle can inject bogus information or cheat about its position. In this work, we analyze the impact of a position cheating attack on an alert message application. We show that this weakness we found could be leveraged by an adversary in a very effective way. Furthermore, our analysis leads us to design a countermeasure to this threat. Finally, we run a set of simulations which confirm our findings.**

I. Introduction

Inter-vehicular communication (IVC) is an emerging research area considerably contributing to traffic safety and efficiency [1]. In this context, many applications are possible; yet, vehicular safety exploiting the fast propagation of alert messages represents one of the most prominent and challenging. Indeed, vehicular safety applications exploiting IVC are often based on 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 [2], [3], [4].

Safety related applications are based on the reliability of broadcast information. Several multi-hop broadcast algorithms have been proposed [2], [3], [5], [6]. Unfortunately, they have all been developed without security in mind, whereas security is a fundamental issue in this context which should not be overlooked. Indeed, attackers might run malicious actions to inject false information or alarm, thus rendering ineffective the safety application [7]. A malicious vehicle can inject false information to misguide other vehicles about traffic conditions, or pretending to be at a claimed, false position, so as to jeopardize the effectiveness of the vehicular safety application. Instead, the correctness of this information is crucial as the efficiency of solutions for alert message propagation is often based on it [2], [3].

The main contribution of our work consists in analyzing the security threats to fast multi-hop broadcast algorithm (FMBA) proposed in [2] for safety application. In particular, we highlight the impact of a position cheating attack on this algorithm. Furthermore, we propose a countermeasure for this threat, developing a solution which is both fast and secure against position cheating in broadcasting safety related messages.

This paper is organized as follows. The next section reviews main works and backgrounds related to the position cheating attacks on IVC, in particular when used to fast broadcast alert messages in vehicular networks. Section III discusses the notation and the functioning of the fast broadcast algorithm FMBA [2]. Section IV details position cheating attack on FMBA algorithm. Section V presents countermeasures for this threat. Section VI discusses the experimental results for the performance of the solution, under different probabilities of message loss. Finally, we give concluding remarks in Section VII.

II. Related Work

In intelligent transportation system [8], IVC (Inter-Vehicular Communication) is an fundamental building block component. In fact, it enables a vehicle to communicate in a multi-hop fashion with other vehicles located out of its transmission range. Minimizing the broadcast delivery time is one of the main challenge for IVC. In the literature, it has been proven that this broadcast time is strictly related to both the number of relays of the messages (hops) and the network congestion [2], [3], [5].

In [2], the authors propose a fast broadcast algorithm (FMBA). It 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 [3], authors have enhanced the fast broadcast algorithm using heterogeneous transmission range. Unlike [2], 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.

Accurate information on position is crucial for IVC based vehicular safety applications. To this aim, detection mecha-

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 has been partially supported by the TENACE PRIN Project (n. 20103P34XC) funded by the Italian Ministry of Education, University and Research. This work is also partially supported by the MIUR/PRIN ALTER-NET and the UNIPD/PRAT Web Squared projects.

nisms have been proposed in this context to recognize nodes cheating about their location [9], [10]. Position verification approaches can be grouped into two main categories ([9], [11]): infrastructure based and infrastructure-less based approaches.

1) Infrastructure based approach: This approach uses special hardware dedicated infrastructure to verify the position of other vehicles. The solutions in [9] 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. The solution in [12] uses verifiers at special locations. This solution needs specific infrastructure: the verifiers. More specifically, these verifiers attempt to verify location claims for region R that are "near" a verifier V. In [10], 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 approach: Solutions of this type can be further classified in parameter based and model based approaches [9]. In parameter based approaches, vehicles check whether a claimed node's position is within a degree of accuracy from the real position. 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 [13]. On the other side, model based solutions compare the regular behavior of the system and current actions to identify anomalies that could indicate malicious behaviors [14]. 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.

III. Preliminaries and Notation

In this section, we introduce the assumptions used in our work (Section III-A), and we give a brief overview of the algorithm FMBA [2] (Section III-B), the protocol which security is addressed in this work.[1] The notation used in this paper is summarized in Table 1.

A. Model assumptions

In the remaining part of this work, we assume the following. We assume that at most one malicious vehicle is on the network, while there are no obstacles and no buildings in the road. The hearing communication range is considered to be symmetric: 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 considered as a collection of 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, while the verifier node V communicates directly with the verified node P. We assume 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 [15]; i.e., each vehicle has a public/private pair of keys and a unique identity certified by a Certification Authority. Finally, the power and computational resources are supposed largely adequate for our application's requirements, and the network is loosely time synchronized.

[1]While FMBA is designed to speed up broadcast both backward and frontward [2], for easy of exposition we consider only the fist case (the other is specular).

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 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

B. Fast Multi-Hop Broadcast Algorithm

Fast Multi-Hop Broadcast Algorithm (FMBA) [2] has the goal of reducing the time required by a message to propagate from the source to the farthest vehicle in a certain area of interest [2]. To achieve this goal, this algorithm 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.

FMBA 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.

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

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 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.

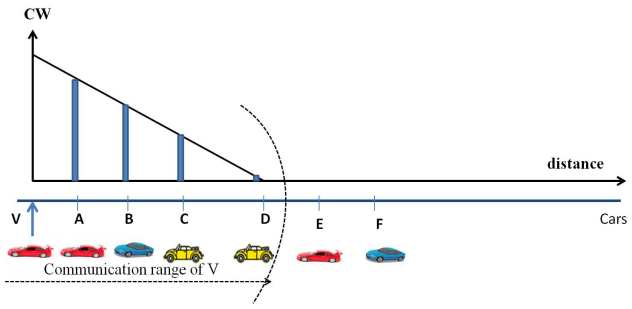


Fig. 2. Contention window versus distance

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 the algorithm 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 the algorithm to vehicles A, B, C, and D. This leads to a general reduction of the number of hops and time needed by a broadcast message to traverse its area of interest.

IV. Position-Cheating Attack

In this section, we show that FMBA is vulnerable to a position cheating attack. In particular, the goal of this attack is to induce a delay in the alert broadcast by increasing the CW of honest vehicles.

To run the position cheating attack, a malicious node announces in a $Hello$ message a false position, i.e. claiming to be more far away in the direction of the alert message (backward in this paper). Hence, honest nodes receiving an alert broadcast message will compute unnecessarily large CWs, thus slowing down the forwarding process. For ease of presentation, Figure 3 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.

Since 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 3 (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 3, 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 3.

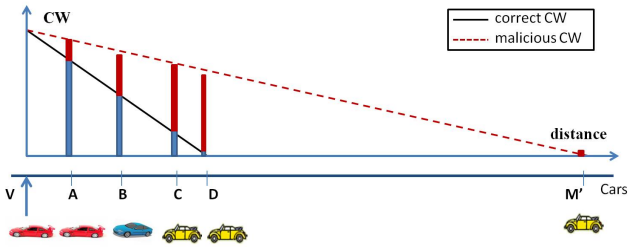


Fig. 3. Impact of distance cheating on the waiting time

The simple position cheating attack described 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. Algorithm 1 describes more in details the attack, as executed by a malicious vehicle M. In fact, M 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. Position Cheating Detection

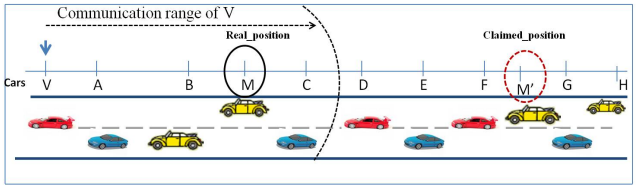
Disseminating the information about vehicle positions is fundamental for FMBA [2]. Hence, vehicles (successfully) cheating about their position can have a severe impact regarding the performance and security of the algorithm. In this work, we propose a detection mechanism that is able

of recognizing nodes cheating about their position. Unlike other proposals described in the literature ([9], [12]), 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.

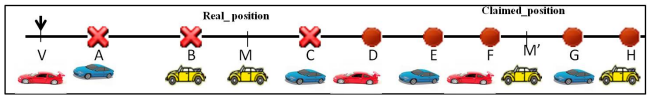
A. Overview

In this section, we present different position cheating attacks, differentiated by the claimed position and the real position of the verified vehicle. The entities involved 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 distinguish three cases based on the real position and the claimed position of M.

In the first case (Case 1), as presented in Figure 4(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 4(c). This legend is also applied to the other following figures. In Figure 4(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



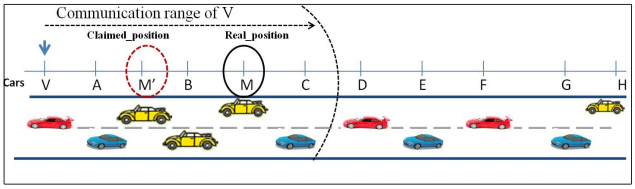
(b) Results of vehicles' reports

(c) Legend

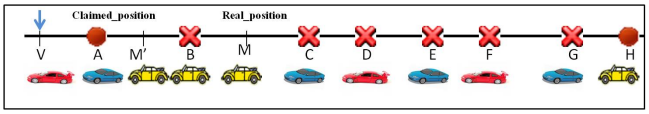
Fig. 4. Case 1

We discuss the second case (Case 2) with the support of Figure 5(a). 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 5(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.

In the third case (Case 3), depicted in Figure 6(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 6(b)).

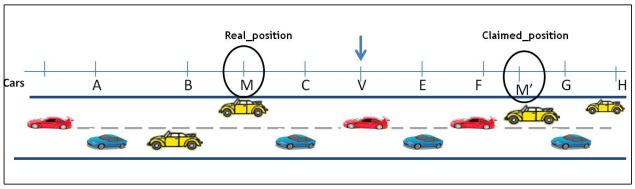


(a) Case 2: Claimed position and real position are within the transmission range of the verifier

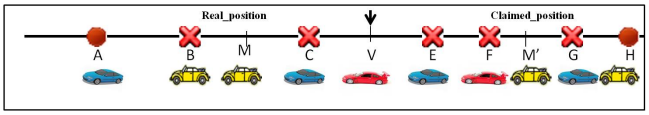


(b) Results of vehicles' reports

Fig. 5. Case 2



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



(b) Results of vehicles' reports

Fig. 6. Case 3

We could summarize these cases in Figure 7, 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. 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 7), 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.

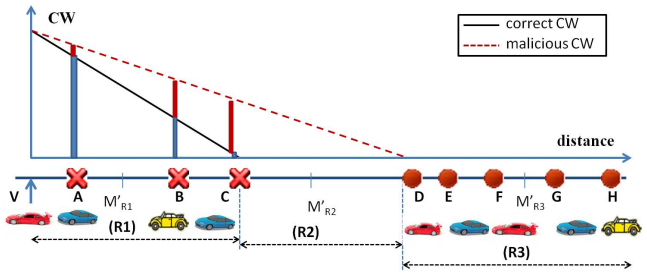


Fig. 7. Verifier waiting time versus distance

B. Description

In this section, we describe our proposed position verification scheme. 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. The Figure 8 illustrates the structure of the modified $Hello$ message. Each element (of type Neighbor) in the $list\_neighbors$ has $vehicle\_id$, $vehicle\_position$, and a list of indirect neighbors. In turn, a list of indirect neighbors is composed by a set of elements of type Indirect Neighbor, i.e. a structure which includes $vehicle\_id$, $vehicle\_position$, $declared\_max\_range$, and $timestamp$.

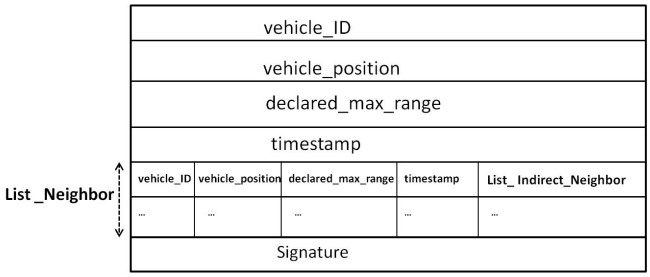


Fig. 8. A modified structure of $Hello$ message

2) Sending $Hello$ message procedure: We focus on the $Hello$ message sending procedure (Algorithm 2). 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 2: 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 3. 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 3, 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 3, 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.

Algorithm 3: Receiving Hello message algorithm (executed by vehicle V)

1 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 ;
2 $< sender\_id\_X, sender\_position\_X, currentTime\_X, drm\_X, list\_neigh\_X, signedHelloMsg\_X >$;
3 $K_X^{Pub} \leftarrow f(sender\_id\_X)$;
4 **if** $H(sender\_id\_X, sender\_position\_X, currentTime\_X, drm\_X, list\_neigh\_X) \neq K_X^{Pub}(signedHelloMsg\_X)$ **then**
5 handle this exception ;
6 **if** *IsNotCoherent* $(current\_time\_X, current\_time\_V)$ **then**
7 handle this exception ;
8 **if** *IsNotPresent* $(list\_neighbors, sender\_id\_X)$ **then**
9 add($list\_neighbors, < sender\_id\_X, sender\_position\_X, currentTime\_X, drm\_X, list\_neigh\_X, signedHelloMsg\_X >$) ;
10 $mp := my\_position()$;
11 $sp := sender\_position$;
12 $drm := declared\_max\_range$;
13 $d := distance(mp, sp)$;
14 **if** $(received\_from\_front(Hello\_msg))$ **then**
15 $CMFR := max(CMFR, d, drm)$;
16 **else**
17 $CMBR := max(CMBR, d, drm)$;

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 4, we present the position verification algorithm executed by a vehicle V. In fact, V collects N reports (Algorithm 4, line 3). For each received report, it checks whether the claimed vehicle M is in the list 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.

Algorithm 4: 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 $i \in 1, ...N do$
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;

VI. Impact of $Hello$ Message Loss: Experimental Evaluation

In this section, we evaluate the performance of FMBA with a position verification (FMBA+PV), and the original algorithm FMBA under $Hello$ message loss. As a metric to evaluate the performance of our algorithms (FMBA, FMBA+PV), we consider the average number of slots required to propagate a broadcast message. We measure the time variables in slots as done in [2]. 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. The main tool utilized for our simulations is NS-2 simulator (version NS-2.29) [16]. 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 [17], 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 9). In our simulation, we considered three attacks for position cheating. In Figure 9, *FMBA Attack* 1 (respectively *FMBA+PV Attack* 1) refers to a position cheating attack claiming 1000 m when running FMBA (respectively FMBA+PV) algorithm. *FMBA Attack* 2 (*FMBA+PV Attack* 2) refers to a position cheating attack claiming 1500 m when running FMBA (respectively FMBA+PV) algorithm. *FMBA Attack* 3 (*FMBA+PV Attack* 3) refers to a position cheating attack claiming between 0 and 2000 m when running FMBA (respectively FMBA+PV) algorithm. We evaluated FMBA and FMBA+PV under the two attacks: FMBA+PV performances are always better than FMBA.

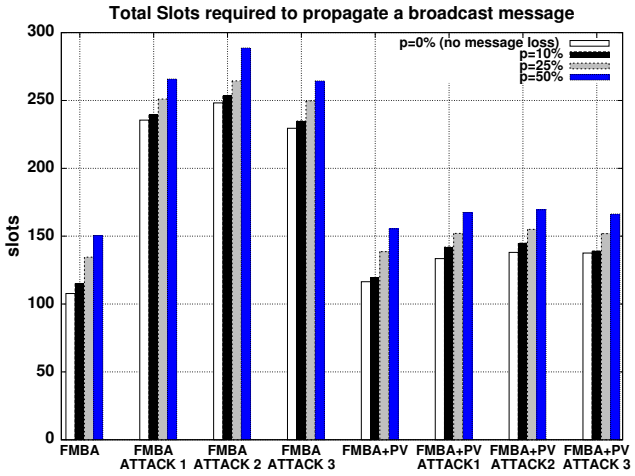


Fig. 9. Impact of Hello message loss

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 9, 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 when FMBA+PV attacked by a malicious vehicle claiming a position of 1000 m (*FMBA+PV Attack* 1 in Figure 9), 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 three 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 FMBA+PV is sound and resilient to message loss, which make it feasible under realistic conditions.

VII. Conclusion

Inter-vehicular communication (IVC) is a fundamental building block for emerging traffic safety applications. In these applications, information such as position, direction, and speed, is usually broadcast to other vehicles to facilitate fast multi-hop propagation of alert messages. In this work we shown how a malicious vehicle could subvert the protocol via cheating about its position. We analyzed the impact of position cheating attack and proposed a way to improve FMBA (a state of the art solution for fast broadcasting) to be resilient to such an attack. We evaluated our solution also with experiments, simulating different probabilities of message loss.

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. 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

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

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] Q. Chen, D. Jiang, V. Taliwal, and L. Delgrossi, "IEEE 802.11 based Vehicular Communication Simulation Design for NS-2 ," ser. VANET 2006, 2006, pp. 50–56.

[17] 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.