DTCF: A Distributed Trust Computing Framework for Vehicular Ad hoc Networks

Tahani Gazdar[1], Abdelfettah Belghith[2,*] and Ahmad AlMogren[2]
[1]HANA Research Laboratory, University of Manouba, Tunisia
tahani.gazdar@isi.utm.tn
[2]College of Computer and Information Sciences, King Saud University, Saudi Arabia
abelghith@ksu.edu.sa, ahalmogren@ksu.edu.sa
[*]Corresponding author: Abdelfettah Belghith

Received June 12, 2016; revised October 27, 2016; revised December 27, 2016; accepted January 22, 2017; published March 31, 2017

Abstract

The concept of trust in vehicular ad hoc networks (VANETs) is usually utilized to assess the trustworthiness of the received data as well as that of the sending entities. The quality of safety applications in VANETs largely depends on the trustworthiness of exchanged data. In this paper, we propose a self-organized distributed trust computing framework (DTCF) for VANETs to compute the trustworthiness of each vehicle, in order to filter out malicious nodes and recognize fully trusted nodes. The proposed framework is solely based on the investigation of the direct experience among vehicles without using any recommendation system. A tier-based dissemination technique for data messages is used to filter out non authentic messages and corresponding events before even going farther away from the source of the event.

Extensive simulations are conducted using Omnet++/Sumo in order to investigate the efficiency of our framework and the consistency of the computed trust metrics in both urban and highway environments. Despite the high dynamics in such networks, our proposed DTCF is capable of detecting more than 85% of fully trusted vehicles, and filtering out virtually all malicious entities. The resulting average delay to detect malicious vehicles and fraudulent data is showed to be less than 1 second, and the computed trust metrics are shown to be highly consistent throughout the network.

Keywords: Trust, security, mobile computing, communication networks, vehicular Networks.

The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no RGP-1436-031

https://doi.org/10.3837/tiis.2017.03.016 **ISSN : 1976-7277**

1. Introduction

Trust based approaches are emerging solutions providing an adequate framework for various safety critical applications in vehicular environments. Such applications require vehicles to authenticate each others as well as to check and control the authenticity of exchanged data in a timely manner. As safety applications involve human lives, designing an appropriate, efficient and accurate trust framework remains a major concern and one of the most important challenges to be able to build cooperative safety applications in VANETs.

In VANETs, trust establishment and management must be tailored to the network characteristics and fitted to take into consideration different challenges [1,2]. First of all, vehicular networks have dynamic topologies. Second, vehicular ad hoc networks do not have a centralized third party. The only possible communications with infrastructures take place with road side units which are not yet deployed at a large scale. As such, it is still not suitable to use mechanisms based on a centralized authority to establish trust in VANETs [3]. As the density of vehicles may get very high in traffic jams or around accidents, a trust model must be scalable [2] providing the same performances independently of the size and the density of the network. The rapid change in road conditions, constitutes another important challenge facing trust establishment in VANETs. The time to react to a given situation is very critical and a vehicle must be able to trust the received information in a reasonable real time.

Numerous methodologies [3, 4, 5] are used to compute trust such as probabilistic models, game theory, Bayesian networks, Fuzzy logic, Swarm intelligence, etc. Most of the existing proposals [6, 7, 8] are based on recommendation systems where vehicles share their opinions about their neighbors. Recommendation systems are commonly used to allow monitors to compute the same trust metric about a given node. However, this approach creates additional signaling traffic that might affect the efficiency of Safety applications, yet the transitivity in recommendation systems accumulates the trust error thus leading to false trust metrics. Furthermore the non detection of malicious nodes constitutes another limitation of most existing approaches. In addition, other approaches such as [8] rely on road side units to efficiently compute the trust metrics, however, the unavailablity of RSUs throughout the network restricts the applicability of such approaches.

In this paper, we propose a new fully Distributed Trust Computing Framework for vehicular ad hoc networks (DTCF). DTCF relies solely on the investigation of direct experiences and observations between neighbors without using a recommendation system or resorting to RSUs. Each vehicle (i.e.; node) monitors the authenticity of the data messages transmitted by its neighbors. Each vehicle (hereafter called monitor) computes a trust metric for each one of its neighbors. Vehicles have to cooperate and to transmit reliable data in order to enhance their perceived trust metrics. Vehicles acquiring the lowest trust level will be revoked and declared malicious throughout the network. In order to enhance the data authenticity check, we use a tier-based technique to disseminate safety messages. Trust metrics are dynamic and updated over time. Monitors (i.e.; neighboring vehicles) of a given vehicle must have a consistent view about this monitored node. The main objective of our proposed trust computing protocol is to provide an adequate framework for trust computing to be used by VANET applications. In particular, such a framework is necessitated to build a distributed public key infrastructure [9, 10, 11] where some fully trusted vehicles play the role of certification authorities (CA) and registration authorities (RA). In general, while malicious vehicles and fraudulent messages should be detected and capturd, only a set (usually small) of fully trusted vehicles are designated to set up a secure framework for the entire network.

The remainder of the paper is organized as follows. In section 2, we present some of the relevant existing trust computing protocols in VANETs. In section 3, we detail the different components of our proposed trust computing framework. In section 4, we evaluate the performance of the proposed DTCF under highway and urban scenarios. Finally, section 5 concludes the paper and presents some viable future directions and further investigations.

2. Related Work

In [7], the authors proposed a trust model useful to decide whether or not to accept a data message from a neighbor. Each vehicle computes a trust value for each neighbor based on the rate of correctly forwarded and transmitted messages. The correctness of messages is compared to the information held by a trusted authority. This induces a huge signaling traffic which limits the practicality of the scheme. Authors in [8] proposed a fuzzy approach where vehicles are classified according to their reputation scores into three fuzzy sets. At first, each vehicle v computes a trust score for each other vehicle j transmitter of warning messages. The scheme takes into account recommendations from other vehicles about j, the old reputation score that v has on j and eventually the recommendations of road side units (RSUs) about j. According to the resulting fuzzy set to which j is mapped to, v decides upon the information received from j. The scheme is time consuming and necessitates a huge signaling traffic, yet it can discard a high number of correct messages when the number of recommendations from neighbors and RSUs is below a predefined threshold. Authors in [12] proposed a data-based trust model for ephemeral networks. First, each vehicle v computes a trustworthiness report about an event taking into account the type of the vehicle and the type of the event. Then all the reports about the same event are combined and their validity is inferred by a decision module (e.g.; Weighted Voting or Bayesian Inference) to decide whether the event reported in the warning messages has really occurred. The scheme only checks the trust on data and does not establish the trust between vehicles.

Shaik and Alzaharani proposed in [13] a new approach aiming to filter out malicious vehicles based on the detection of false location and time information. The scheme is based on three phases. First a vehicle calculates a confidence value for each data message received from its neighbors. This confidence value is computed as a function of the proximity of the location to the event and the message reception time as compared to the instant of event occurrence. Then, a trust value for each message is computed based on the confidence value. This trust value is used to decide about the forwarding of the message. This scheme has a rather high false positive rate. Furthermore, the location metric is not sufficient to decide about the legitimacy of the transmitted data and the trustworthiness of the transmitter. In [6] and [14], proposals are based on the Dempster-Shafer Theorem (DST) [15]. DST is used in order to combine many independent neighboring vehicles beliefs about a given vehicle to compute the trust metrics. However, beliefs received from other nodes might be obsolute and do not reflect the vehicle recent behavior. Furthermore, false trust beliefs might induce erroneous computed trust metrics. Routing is another application of trust [16, 17]. In [18] for example, authors proposed TROUVE which is a trust based routing protocol for vehicular networks. Each node stores information about its neighbors including the Packet Drop Ratio (PDR) and Packet Sent Ratio (PSR). In order to compute the trust metric, each vehicle compares the PDR and the PSR to a given predefined threshold and updates the trust metrics accordingly.

Our proposed trust model is fully-distributed and is based on a monitoring process of both the transmitting nodes and the transmitted messages. We combine vehicles cooperativeness and messages authenticity. Our purpose is manyfold as we aim to filter out all malicious

vehicles and fraudulent data, decide on the trust metric of each and every node within the network, and detect the largest set of fully trusted vehicles that could serve to establish a security architecture as in [10, 11]. Furthermore, special attention is given to the consistency of the computed trust metrics and the time required to detect fully trusted nodes and to declare malicious nodes.

3. DTCF: A Distributed Trust Computing Framework

3.1. Network Model

One of the major applications of vehicular networks is to ensure the safety of vehicles as well as that of road users [19]. This is mainly based on the exchange of messages between vehicles to disseminate road conditions and information relative to traffic (congestion/fluidity), weather conditions, presence of road work, etc,. Vehicles exchange messages to fluidize the road traffic and alert each other of urgent events on the roads. However, some malicious vehicles might tamper the content of these messages or even disseminate false information within the network to influence the decisions of other drivers. Malicious vehicles may stipulate the existence of certain (fake) events on the roads [2]. These fake events should be filtered out and malicious nodes should be detected. Each vehicle monitors its neighboring vehicles (namely vehicles within its transmission range). A monitor updates the trust metrics of its neighbors based on the perceived authenticity assessment of their tramsmitted messages about the state of the road.

To model occurring events (e.g.; accidents, jams, presence of work, etc.,), we consider a set of N_e (virtual) generators of events placed randomly throughout the network. There are two types of events: *authentic* events that really did occur on the roads and *fictitious* events that did not occur in reality. While *authentic* events are perceptible and treated by all vehicles in their immediate surroundings, *fictitious* events are only treated by surrounding malicious vehicles, meaning that non malicious vehicles just ignore these fake events. Authentic events serve to model the real events that occur on the different roads of the modeled network. Fictitious events, on the other hand, serve to model false or fake events that are stipulated and forwarded only by malicious vehicles. Event e has a unique identifier id_e, a position (x,y), and an occurrence instant t_e. For each event corresponds an authentication value $auth(e)$ that indicates whether event e is authentic ($auth(e) = 1$) or fictitious ($auth(e) = 0$).

For each occurring event, we define a zone of interest that comprises all nodes to which the event should be disseminated. The zone of interest is denoted by Z_I, has a radius z_i from position (x, y) of event e. The radius z_i is usually multiple of the transmission range. Furthermore, we define a detection zone Z_d with radius $z < z_i$ around position (x, y) of event e. Usually, the radius of Z_d is a fraction of the transmission range. Only vehicles within this detection zone of e can directly detect (observe) this event. The other vehicles within geographic zone Z_I but not in zone Z_d should learn about the event using multi hop communications originated from vehicles within Z_d. If event e is authentic ($auth(e)=1$), all vehicles in zone Z_d whether malicious or not observe the event using their sensors, for instance, and then treat the event. However, if event e is fictitious $auth(e)=0$, it is only treated by malicious vehicles within zone Z_d, non-malicious nodes just ignore such fictitious events.

3.2. Attacker Model

We consider two misbehavior categories of vehicles: selfish vehicles and malicious vehicles. In fact, applications in VANETs are generally based on the cooperation between vehicles, and require a high level of cooperation for efficiency matters. However, there are selfish vehicles that refrain from forwarding messages to others, in the quest of example to enhance their access to services of their own interests.

In our network model, each vehicle v transmits messages with a probability $F(v)$. If $F(v)=1$ then vehicle v is fully cooperative, otherwise it is said to be partially cooperative. Furthermore and in addition to fictitious false events, there are some malicious vehicles that may alter transmitted messages or transmit fraudulent information which harms the proper functioning of running applications specially those involving safety [2]. In our framework, we define two aspects of malicious behavior. First, a malicious vehicle v is each vehicle that inverses the authenticity of an event either from 1 to 0 or from 0 to 1 with probability $P_m(v)$. $P_m(v)$ represents in our model the probability of a malicious node to tamper or alter the content of a message. To each vehicle corresponds its own value of $P_m(v)$ chosen independently from other nodes and randomly in the interval [0,1]. A second aspect of the malicious behavior is when a malicious vehicle broadcasts messages about fictitious events as defined in Section 3.1.
Note here that malicious vehicles having P_m equal to 0, do not tamper the content of messages but transmit all messages about fictitious events as well as all messages about authentic events.

3.3. Trust Metric

The ground principle is that each vehicle monitors the behavior of its 1-hop neighbors. It assesses the authenticity of received messages and then updates accordingly the trust metric assigned to each neighbor. We denote by $T_m(M, v)$ the trust metric of vehicle v calculated by its monitor (i.e.; neighbor) M. It is a value in [0, 1] that relies on the authenticity of the data broadcast by vehicle v. v is considered malicious by M if $T_m(M, v) = 0$, and fully trusted if $T_m(M,v) = 1$. Values of $T_m(M, v)$ in the interval]0, 1[correspond to intermediate trust levels. Initially, a monitor assigns a default trust metric value $T_m = 0.1$ to unknown vehicles. Unlike many existing proposals such as [7, 6, 13] where no initial trust metric is assigned to vehicles, in our model the default trust metric value allows avoiding the cold-start problem [20]. Monitor M updates the trust metrics of its neighbors based on the perceived authenticity of the messages received from the monitored node. The trust metric of a monitored node is increased each time the monitor receives an authentic data message from this node and it is nullified upon the reception of a non-authentic message.

3.4. A Tier-based data dissemination

3.4.1. Perception of events in Z_d

Upon the occurrence of an event on a road, vehicles within zone Z_d of such an event must announce it to other vehicles. The event dissemination is organized into layers (tiers). Zone of interest Z_I is then divided into $L + 1$ tiers, tier k has a radius r_k from position (x, y) of event e:

$$r_k = kR \qquad k = 1, \dots, L \tag{1}$$

where R represents the transmission range. Zone Z_d is tier 0 with a radius $z < R$ (recall that z is a fraction of the transmission range). Zone Z_k ($k = 1, \ldots, L$) is the zone that contains all nodes between layer k and layer $k - 1$. **Fig. 1** portrays the disposition of the tiers or layers in Z_I.
All events (authentic and fictitious) are triggered according to a Poisson process with parameter $1/\mu$ at each event generator. Each event lasts for a time period δ during which warning messages denoted *W-Message* are periodically broadcast by the event generator in Z_d with time period τ. The reception of *W-messages* by a vehicle within zone Z_d models the sensing (namely the direct detection) of the event by this vehicle.

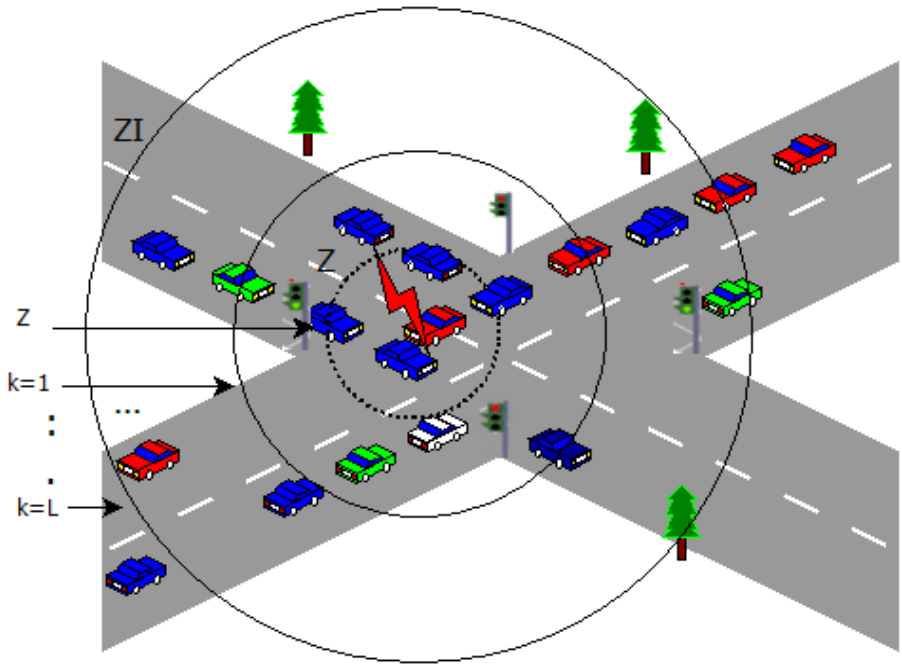


Fig. 1. Schematic Tiers in zone Z_I for an authentic event

W-Messages have the following content:

$$W\text{-}Message = (id_e, x, y, te, auth(e), seq)$$

where field *auth(e)* indicates whether event e is authentic or not, and *seq* denotes the sequence number of the message, $seq = 1, \ldots, (\delta / \tau)$.
A vehicle v perceives an event e in the corresponding zone Z_d by detecting at least one *W-Message* that describes it. We distinguish here two cases depending on the authenticity of the event. If field *auth(e)* indicates that event e is **authentic** ($auth(e) = 1$) then regardless of its nature (malicious or not) vehicle v broadcasts an alert message denoted *A-Message* containing a description of the event such as the position and the instant of its occurrence, and a reputation value denoted *Rv(e)* that reveals the authenticity of e set by vehicle v:

$$A\text{-}Message = (id_e, x, y, t_e, R_v(e), v.id, seq)$$

Reputation field $R_v(e)$ is set by vehicle v before transmitting its *A-Message*. However, the value of $R_v(e)$ depends on the behavior of v: if v is malicious then reputation $R_v(e)$ is set to 0 with probability $P_m(v)$ (lines 2-5 in **Algorithm 1**), otherwise $R_v(e)$ is set to 1. Nevertheless, if

field *auth(e)* indicates that event *e* is **fictitious** then *v* broadcasts an *A-Message* only if it is malicious (lines 6-9 in **Algorithm 1**). Recall that non-malicious vehicles ignore the fictitious *W-Message* in Z_d. Each vehicle *v* transmits all its messages with probability *F(v)* representing its degree of selfishness. The transmissions to the upper tier are randomized on a transmission interval denoted τ_1 in order to avoid collisions as it will be detailed later on. **Algorithm 1** provides the action accomplished by node *v* located in zone Z_d upon the reception of *W-Messages*. **Algorithm 1** calls a function called "*Broadcast*" useful for vehicle *v* to broadcast *A-Messages*. In order to model the malicious behavior of node *v* with identifier *v.id* according to $P_m(v.id)$, node *v* picks a random value *r* each time it transmits an *A-Message* as detailed in **Algorithm 1** (lines 2-10).

Algorithm 1. DETECT_W_Message($v.id$, id_e, $auth(id_e)$)

1. **If** (*v.id is malicious*) **Then**
2. **If** ($auth(id_e) == 1$) **Then** $r = random(0,1)$;
3. **If** ($r \leq P_m(v.id)$) **Then** $R_{v.id}(id_e)=0$;
4. **Else** $R_{v.id}(id_e) = 1$;
5. **EndIf**
6. **Else** $r = random(0,1)$;
7. **If** ($r \leq P_m(v.id)$) **Then** $R_{v.id}(id_e) = 1$;
8. **Else** $R_{v.id}(id_e)=0$;
9. **EndIf**
10. **EndIf**
11. *Broadcast* (*A-Message*(id_e, $R_{v.id}(id_e)$, $v.id$));
12. **Else If** ($auth(id_e)==1$) **Then** $R_{v.id}(id_e) = 1$;
13. *Broadcast*(*A-Message*(id_e, $R_{v.id}(id_e)$, $v.id$)) ;
14. **EndIf**
15. **EndIf**

3.4.2. Reception of messages in Z_d

Reception of *A_Message*: Consider a monitor *M* in zone Z_d of an event *e*. Upon the reception of an *A-Message* from vehicle *v* in zone Z_d, one of the four cases enumerated in **Table 1** prevails depending on the value of $R_v(e)$ received from *v* in its *A-Message* and the field *auth(e)* of the *W-Message* that monitor *M* has either detected or assumed about *e*. Recall that if *e* is fake then non malicious nodes within zone Z_d ignore it and therefore assume *auth(e)= 0*.

Table 1. The receipt of *A-Message* in zone Z_d

auth(e)	$R_v(e)$	**Action of monitor *M***
0	0	*M* declares *v* as malicious, sets $R_M(e) = 0$ and sends a *D-Message*.
0	1	*M* declares *v* as malicious, sets $R_M(e) = 0$ and sends a *D-Message*.
1	0	*M* declares *v* as malicious, sets $R_M(e) = 1$ and sends a *D-Message*.
1	1	*M* sets $R_M(e) = 1$ and increments $T_m(M, v)$

A special message called the *D-Message* is used to declare the identities of malicious vehicles (field *v.id*) and eventually fictitious events (field id_e) throughout the network. It has the following specification:

$$D\text{-}Message = (v.id, id_e, M.id)$$

D-Messages are broadcast over a number of hops in the network until reaching a trusted authority, such as a certification authority in a public key infrastructure or a road side unit in a vehicular network, which in turn takes the responsibility to disseminate the information. Notice that for the first three entrees of **Table 1**, monitor *M* declares vehicle *v* as malicious and sends a *D-Message* to broadcast this information throughout the network. The only entry of **Table 1** that increases the value of $T_m(M, v)$ is the fourth entry where the directly perceived authenticity by *M* is the same as the one received from vehicle *v*. Notice here that the reception of an *A-Message* in zone Z_d of event *e* can generate the transmission of *D-Messages* as specified in **Table 1** but does not generate any further *A-Messages* to be transmitted. The action of monitor node *M*, upon the receipt of *A-messages* in Z_d, is described in **Algorithm 2** (lines 3-14)**.**

Reception of *D-Message*: When a vehicle in zone Z_d receives a *D-Message*, it admits that vehicle *v.id* is malicious and it no longer considers its *A-Messages*.

3.4.3. Transmission and reception of messages in tiers *k=1, ... , L*

Reception of *A-Message*: Messages received from already declared malicious vehicles are ignored. Each vehicle maintains a black list containing the identities of already declared malicious vehicles. Upon the reception of an *A-Message* from the neighborhood, vehicle *M* first determines the tier to which it belongs within Z_I. A vehicle ignores messages received from its neighbors within its own tier as well as messages from upper tiers. If vehicle *M* is in tier $k=1$ and the *A-Message* about event *e* is transmitted by a vehicle *v* in zone Z_d, then it waits a maximum period of time τ_2 during which it might receive other similar messages from other vehicles in Z_d or a *D-Message* (lines 15-23 in **Algorithm 2**).
The period τ_2 serves two objectives: the collecting of *A-Messages* emanating from zone Z_d about event *e*, and eventually the reception of *D-Messages* declaring malicious nodes and fake events detected in zone Z_d. The collecting of *A-Messages* reduces the number of *A-Messages* to be transmitted; each monitor in Z_1 transmits only one *A-Message* as a response to the entire set of *A-Messages* received during τ_2. Upon the expiration of τ_2 without receiving any *D-Message* about the same event, each monitor *M* verifies the authenticity of the received messages and updates the trust metric of each vehicle transmitter of an *A-Message* concerning event *e*. Then, if event *e* is authentic, *M* transmits to nodes in zone Z_2 a unique *A-Message* concerning event *e*. However, if monitor *M* is in a tier $k > 1$ (lines 25-36 in **Algorithm 2**), then it immediately transmits an *A-Message* if the event is authentic (without any waiting as done in layer 1). The value of the reputation $R_M(e)$ in the transmitted message is set equal to 0 with probability $P_m(M)$ if *M* is malicious, and equal to 1 otherwise. The action taken by monitor *M* (*having identifier M.id*) upon the reception of *A-Messages* is described in **Algorithm 2** where pos_M and *n* denote respectively *M*'s current tier and the number of *A-Messages* received by *M* about event *e*.

The time axis shown in **Fig. 2a** corresponds to the dissemination of messages in the case of an authentic event happening at instant t_e. Period τ_1 corresponds to the duration to randomize the transmissions of messages to the next tier. Period τ_2 is the maximum time period, used only at tier $k=1$, to accumulate the *A-Messages* received from zone Z_d. Instants t_i, $i=1,\dots, L-1$ correspond to the end of periods τ_1 in the corresponding i^{th} tier. Regarding fake events, *A-Messages* should not be transmitted out of tier $k = 1$. The corresponding time axis is shown in **Fig. 2b**. Indeed, *A-Messages* about a fictitious event emanate only from malicious nodes in

zone Z_d. These *A-Messages* generate automatically *D-Messages* by non-malicious nodes in Z_d or eventually by non-malicious nodes in Z_1. In summary any *A-Message* with $R_v(e) = 0$ will be certainly captured and its transmitter will be declared as a malicious node.

Algorithm 2. Receive_A_Message (id_e, t_e, $R_v(e)$, *v.id*, pos_M, *M.id*)

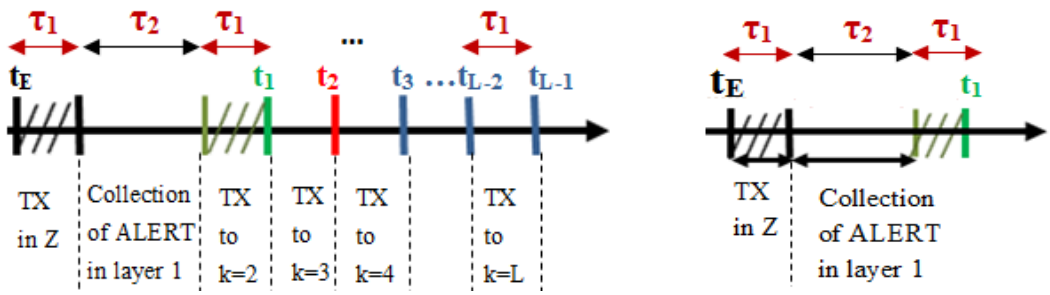
```
1. If (v.id is not in the black list of M.id) Then
2. Switch pos_M do:
3.{Case Z_d: If (v.id is in Z_d) Then If(R_v.id(id_e)==1)Then
4.                                  If(M.id didn't perceive id_e) Then
5.                                          MONITOR(0,id_e,v.id);
6.                                  Else        MONITOR(1,-1,v.id);
7.                                  EndIf
8.                                  Else If(M.id perceived id_e) Then
9.                                          MONITOR(0,-1,v.id);
10.                                         Else MONITOR (0,id_e,v.id);
11.                                     EndIf
12.                             EndIf
13.         EndIf
14. Break;
15. Case 1: If(v.id is in Z_d) Then If(M.id already detected id_e) Then
16.                                 If(R_M.id(id_e)==1 & R_v.id(id_e)==1) Then
17.                                         MONITOR(1,-1,v.id);
18.                                 Else    MONITOR (0, -1, v.id);
19.                                 EndIf
20.                                 Else If (τ_2 is active) Then  n++ ; EndIf
21.                             EndIf
22.         EndIf
23. Break;
24. Default:  If (v.id is in tier (pos_M -1)) Then
25.             If (R_v.id(e)==1) Then  MONITOR (1,-1,v.id);
26.                             If (L>pos_M & τ_1 is active) Then
27.                                     If (M.id is malicious) Then r= random(0,1);
28.                                             If(r ≤ P_m(M.id)) Then R_M.id(id_e)=0;
29.                                                     Else R_M.id(id_e) =1;
30.                                             EndIf
31.                                         Else R_M.id(id_e) =1;
32.                                     EndIf
33.                                     Transmit (A-Message(id_e ,R_M.id(id_e), M.id));
34.                             EndIf
35.                         Else    MONITOR (0, -1,v.id);
36.             EndIf
37.         EndIf
38.  Break; }
```

Algorithm 3. MONITOR (*increase*, id_e, *v.id*)

1. **If** (*increase==0*) **Then** $T_m(M.id,v.id)= 0$;
2. *Broadcast (D-Message (v.id, id_e));*
3. **Else If** ($T_m(M.id,v.id)<1$) **Then** $T_m(M.id,v.id)= T_m(M.id,v.id)+\gamma$;
4. **EndIf**
5. **EndIf**

The purpose of using the concept of tiers in the forwarding of messages is to filter out messages containing a wrong value of the event reputation as well as to declare malicious vehicles before moving away from the event position. In addition, it allows creating waves of similar messages generated in the same spatio-temporal context of events which improves the accuracy of the verification of authenticity. Moreover, this technique guaranties that there are no possible collisions between waves of *A-Messages* coming from different tiers about the same event given no two nodes from three consecutive tiers transmit simultaneously. This is satisfied if the following condition holds:

$$\min\{3\tau_1 + \tau_2, (L+1)\tau_1 + \tau_2\} \leq \tau \quad \textbf{(2)}$$



a. Authentic event **b. fictitious event**

Fig. 2. Time division

Reception of *D-Message*: A monitor that receives a *D-Message* indicating that vehicle *v* is malicious deduces immediately that the event reported by vehicle *v* is fictitious and suspends the timer τ_2 if it is already scheduled.

3.5. Authenticity verification and trust metrics update

Recall that each event lasts for δ seconds during which *W-Messages* are periodically broadcast by the event generator in Z_d with time period (τ). Since an *A-Message* with $R_v(e) = 0$ emanates only from a malicious node and since this node works with a chosen P_m, a malicious vehicle should be detected the latest after δ/τ transmitted *A-Message*. The number δ/τ should be large enough to allow such a detection. Without loss of generality, we may assume that $\delta/\tau \geq 10$ as usually is the case in VANETs. The probability to detect a malicious node is given by:

$$P_{detection} = 1 - \left(1 - P_m(v)\right)^{\delta/\tau} \quad \textbf{(3)}$$

In **Table 2**, we present some values of the probability of detection $P_{detection}$ for various $P_m(v)$ and different values of the number of *A-Messages* sent during δ by a given node *v* in Z_d about the same event.

Table 2. Probability $P_{detection}$ as a function of the number of transmitted messages

δ/τ	$P_m(v)=0.8$	$P_m(v)=0.5$	$P_m(v)=0.1$
1	0.8	0.5	0.1
2	0.96	0.75	0.19
3	0.992	0.875	0.271
4	0.998	0.937	0.34
5	1	0.999	0.4
10	1	1	0.65
20	1	1	0.87
30	1	1	0.95

3.5.1. Authenticity verification and trust metrics update in zone Z_d

A monitor M in Z_d, upon the reception of an *A-Message* from vehicle v within Z_d, behaves according to **Table 1** where $T_m(M, v)$ is incremented only if the conditions stated at the fourth entry are verified. For all other three cases, monitor M sets $T_m(M, v) = 0$ and declares v as a malicious node and consequently broadcasts a *D-Message*. The question naturally arises as to whether all malicious nodes within zone Z_d are going to be detected. The answer is definitely affirmative if two or more nodes are residing in zone Z_d since then **Table 1** is in effect. The question pertains when a unique node is within zone Z_d and this node is malicious. In this case there is no other node in Z_d that will declare it according to **Table 1**. But ultimately this single malicious node, say v, in Z_d will be caught and detected by a node in layer *1* upon its transmission of the first *A-Message* with $R_v(e) = 0$ as indicated by **Table 2**.

3.5.2. Authenticity verification and trust metrics update in zone $k = 1$

Vehicles in Z_1 do not directly perceive the happening of the event and therefore do not detect the *W-Messages* about such an event. As such, vehicles in Z_1 cannot just forward the received *A-Messages* but instead should wait a period of time τ_2 larger than τ_1 to be able to decide which *A-Messages* really emanated from malicious vehicles. The rationale behind using a waiting period τ_2 is twofold. Firstly, it enforces the inspection of all received *A-Messages* and *D-Messages* emanating from zone Z_d to properly decide upon the authenticity of the *A-Messages*. Secondly, only one unique *A-Message* and/or *D-Message* is then transmitted about the same event. The later fact has the nice property to reduce the required signaling traffic of our proposed DTCF. Normally no *A-Message* about a fake event is transmitted from Z_1 unless the network is sparse and one and only one vehicle is in Z_d that is malicious. In such a case, it will be detected after some time according to **Table 2** and depending on its P_m and the dynamics of the network as further vehicles might enter zone Z_d and detect it as malicious. When a monitor receives a *D-Message* indicating that a vehicle v is malicious, it deduces immediately that the event is fictitious and suspends timer τ_2. Besides any malicious vehicle in Z_1 would be detected by its neighbors at this same tier as soon as it transmits an *A-Message* with $R_v(e) = 0$.

After the authenticity assessment step, monitor M updates the trust metrics of vehicles from which it has received A-Messages about event e. The initial value of $T_m(M, v)$ is set to *0.1*. If the authenticity received in an *A-message* is correct (i.e.; equal 1 for an authentic event) then monitor M increments the trust metric of vehicle v by a step γ (with 1 a multiple of γ). That is:

$$T_m(M, v) = T_m(M, v) + \gamma \tag{4}$$

Otherwise, the trust metric of vehicle v is nullified ($T_m(M, v)=0$), v is declared malicious and a *D-Message* is scheduled for transmission. If vehicle v is declared malicious, other vehicles will permanently ignore all its transmitted messages. **Algorithm 3** describes the updating of the trust metric of vehicle v by monitor M.

3.5.3. Authenticity verification and trust metrics update in zone $k > 1$

There is no need for a collecting period similar to τ_2. If monitor M is in tier $k>1$ and it receives an *A-Message* containing $R_v(e) = 1$ from vehicle v belonging to tier $k - 1$, then the trust metric of vehicle v, $T_m(M, v)$, is incremented by γ. However, if $R_v(e) = 0$ then vehicle v is declared to be malicious, its $T_m(M, v)$ is set to zero, and a *D-Message* is scheduled for transmission.

4. Performance Evaluation

In order to investigate the efficiency of our proposed framework and the consistency of its results, we conducted a set of simulations using the network simulator OMNET++ [21] conjointly with the road traffic micro-simulator SUMO [22]. We simulated both highway and urban scenarios.

4.1. Simulation setup in the Urban Scenario

We consider an urban model composed of 9 crosses (intersections) separated by a distance of 1000m in a network area of (4000m x 4000m) as sketched on **Fig. 3**. In this model, vehicles speed up to a maximum of 15m/s (e.g., 54 Km/h) towards different directions.
Vehicles enter into the network from the different 12 endpoints, with an arrival rate of 0.1 vehicles/s from each entry point and each vehicle arbitrary chooses its own destination. The event generators are localized at the crosses as portrayed on **Fig. 3**. The number of event generators is denoted by N_e, which is either 5 (left sub-figure) or 9 (right subfigure). The number of non-authentic events generators is set to around 25% of N_e (*1* for $Ne = 5$ and *2* for $N_e = 9$ in the conducted simulations). Malicious nodes randomly choose their P_m in [0-1], unless otherwise stated. The chosen values of F and/or P_m are kept unchanged during the entire simulation. Simulations are run for 1000s and repeated enough to attain 95% of confidence interval. The rest of the simulation parameters are as given in **Table 3**.

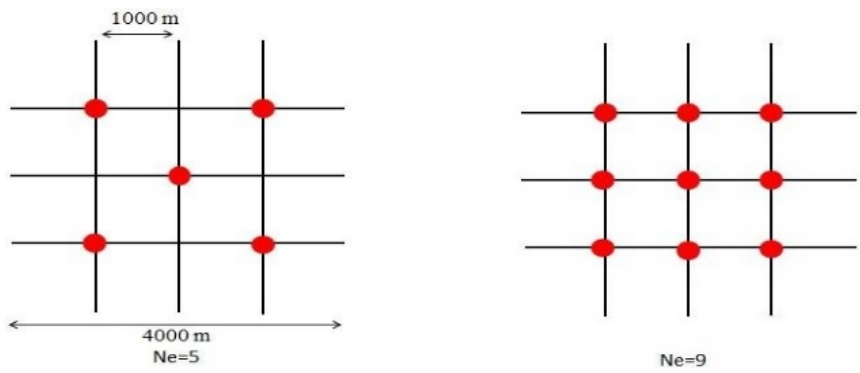


Fig. 3. The urban network

Table 3. Parameters Values

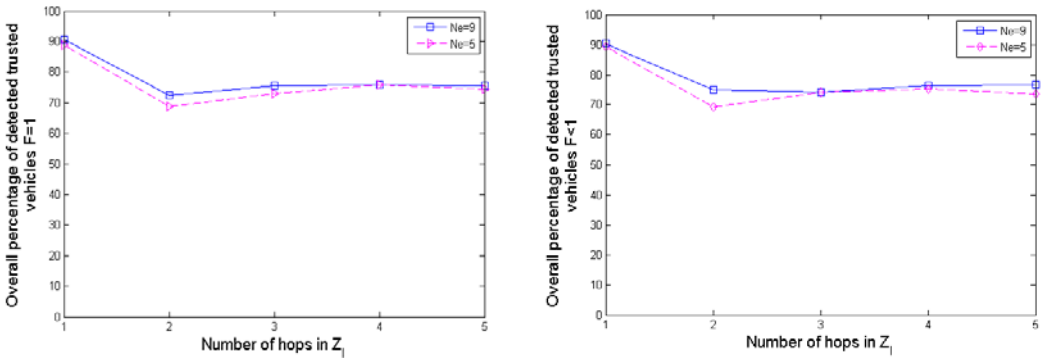
Parameters	Values
The average number of vehicles	700
L	1, 2, 3, 4, 5
z (meters)	150
μ (seconds)	300
Transmission radius R (meters)	350
Percentage of selfish non malicious vehicles	10%
Percentage of malicious non selfish vehicles	10%
Percentage of malicious and selfish nodes	10%
δ (seconds)	150s
τ_1	0.1s
τ_2	0.3s
τ	As per the equality in **Eq. 2**

4.2. Simulation results of the Urban Model

We first focus on the overall percentage of detected trusted vehicles and detected malicious vehicles per monitor. Let E_M denotes the set of vehicles encountered by monitor M. The percentage of detected trusted vehicles per monitor M, denoted by $\%TR_M$, is defined as the fraction of vehicles that reached the trusted state at monitor M from the total number of non-malicious and cooperative (F = 1) vehicles encountered by M. Let the function ψ be such that $\psi(c)=1$ if condition c is true and 0 otherwise. $\%TR_M$ is then expressed as given by **Eq. 5**.

$$\%TR_M = \frac{\sum_{v \epsilon E_M} \psi(T_m(M,v)=1 \;\&\; F(v)=1)}{\sum_{v \epsilon E_M} \psi(F(v)=1 \; and \; v \; is \; non \; malicious)} 100 \tag{5}$$

We portray in **Fig. 4a** and **Fig. 4b** the overall percentage (per monitor among all monitors) of detected trusted vehicles as a function of the number of hops in Z_I and the number of encountered events N_e, and that for both cases of fully cooperative ($F=1$) and selfish ($F=0.8$) vehicles respectively.



a. F=1 **b. F=0.8**

Fig. 4. The overall percentage of detected trusted vehicles in the urban model vs. the number of hops in Z_I and the number of non authentic events generators N_e

We clearly observe in **Fig. 4a** that the average percentage of trusted vehicles is around 90% for 1 hop, and decreases to 70% for 2 hops and then increases to around 75% between 3 and 5 hops. Recall that in our urban scenario, events generated at crosses create congestions. As a result, more vehicles remain away from the detection zone Z_d and are then able to resume their speed (15m/s). Consequently, the average encounter duration becomes shorter compared to that of the first hop of Z_I. In addition, vehicles change directions frequently which results in very dynamic neighborhoods. This involves also vehicles departing from the simulated area. The increase from 70% to 75% when we pass from 2 to more hops is essentially due the overlapping different zones Z_I corresponding to different events. The difference in 2 hops between N_e=5 and N_e = 9 is about 3%, and overall the number of hops in Z_I does not have a tangible impact on $\%TR_M$ as *A-Message*s are not transmitted out of tier 1 for fake events.

Let us now consider **Fig. 4b** portraying $\%TRS_M$ the same as the previous **Fig. 4a** but for selfish vehicles ($F<1$).

$$\%TRS_M = \frac{\sum_{v\epsilon E_M} \psi(T_m(M,v)=1 \;\&\; F(v)=0.8)}{\sum_{v\epsilon E_M} \psi(F(v)=0.8 \; and \; v \; is \; non \; malicious)} 100 \tag{6}$$

We found approximately the same result as in **Fig. 4a** with a slight decline for $N_e = 5$. There is a sufficient number of events, that even selfish vehicles can reach the trusted state.

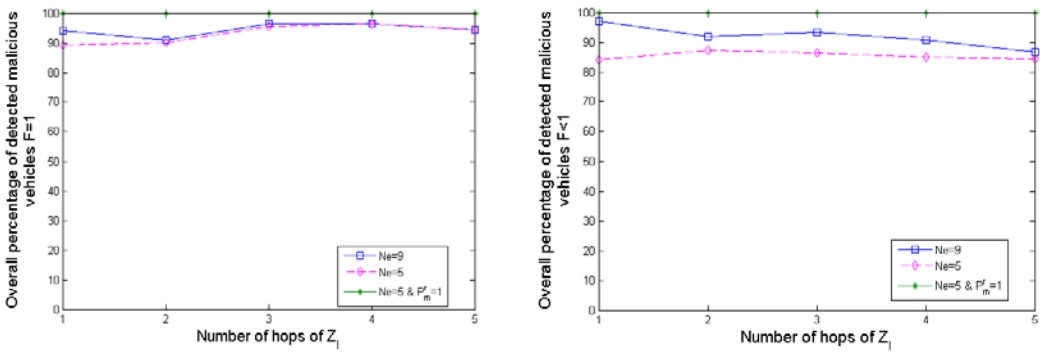
We turn now to the number of declared malicious vehicles. We plot in **Fig. 5a** and **Fig. 5b** the overall percentage of declared malicious vehicles (per monitor over all monitors) for both cases of $F = 1$ denoted $\%ML_M$ and given by **Eq. 7** and $F < 1$ denoted $\%MLS_M$ and given by **Eq. 8** as a function of the number of hops in Z_I and the number of events generators N_e.

$$\%ML_{\mathrm{M}} = \frac{\sum_{v\epsilon E_M} \psi(T_m(M,v)=0 \;\&\; F(v)=1)}{\sum_{v\epsilon E_M} \psi(F(v)=1 \; and \; v \; is \; malicious)} 100 \tag{7}$$

$$\%MLS_M = \frac{\sum_{v\epsilon E_M} \psi(T_m(M,v)=0 \;\&\; F(v)=0.8)}{\sum_{v\epsilon E_M} \psi(F(v)=0.8 \; and \; v \; is \; malicious)} 100 \tag{8}$$

Fig. 5a shows that the percentage of declared malicious vehicles is above 90% and increases to more than 95% as the number of hops increases. This is mainly due to the entry of vehicles into overlapping zones of interest. The reason of not attaining 100% is due to those malicious vehicles adopting very small values of P_m and as such behaving more like non-malicious nodes. Furthermore, there are certainly some malicious vehicles that encountered their monitors for very short periods of time while they were exiting from the network. Had they stayed in the network, they would have been certainly detected. The excellent detection capability of DTCF is exacerbated when $P_m = 1$, we see in **Fig. 5a** that all malicious nodes are detected independently of the number of layers in Z_I.

Fig. 5b shows the percentage of declared malicious selfish vehicles as per **Eq. 8**. We notice a decrease in the curves compared to those of **Fig. 5a**. However the selfish behavior has no effect when malicious nodes use $P_m = 1$, showing once again the DTCF excellent detection capability of malicious vehicles.



a. *F=1* **b. *F=0.8***

Fig. 5. The overall percentage of detected malicious vehicles in the urban model vs. the number of hops in Z_I and the number of non authentic events generators N_e

In contrast to the above, we now investigate the detection capability of our proposed DTCF per layer (within each layer) of the zone of interest Z_I. We adopt here $L = 3$ and a random P_m chosen individually by each malicious vehicle.

Fig. 6 portrays the percentage of trusted and malicious vehicles detected in layers 0, 1 and 2. The closer is the monitor from the event position the more efficient is the protocol in detecting malicious vehicles. In fact in tiers 0 and 1, a monitor detects 100% of encountered malicious vehicles, however only 91% of encountered malicious vehicles in layer 2 are detected. For trusted vehicles, there is a slight difference between layer 0 and layer 2. However, the percentage of detected trusted nodes in tier 1 is lower than that in the other two layers, because in tier 1 a monitor must wait τ_2 to collect all *A-Messages* emanating from tier 0 which provides enough time for vehicles to exit the tier.

We now study the average delay needed by a monitor to detect a trusted vehicle. Consider vehicle v monitored by vehicle M. M receives the first *A-Message* from v at instant t_v. At instant ($t'_v \geq t_v$), M detects that v becomes trusted after receiving a sufficient number of *A-Messages* which allows the transitions of $T_m(M, v)$ from the non-trusted state to the trusted state. The average latency per monitor M is calculated over the set of all its monitored vehicles denoted S_M, and is computed as given by **Eq. 9**, where $|S_M|$ denotes the cardinality of the set S_M

$$delay(M) = \frac{\sum_{v \in S_M}(t'_v - t_v)}{|S_M|} \tag{9}$$

Fig. 7 portrays the average latency (over all monitors) to detect a trusted vehicle as a function of the number of hops (tiers) in Z_I and the number of event generators N_e.

Interestingly, we notice that for one hop the latency is just below 5s which corresponds to the minimal average duration needed by the same vehicle v (with $F(v) = 1$) to transmit 9 A-*Messages* to monitor M about the same event. This enables monitor M to successively increment $T_m(M, v)$ up to the trusted state. Recall that the periodicity τ of *W-Messages* is computed based on the equality in **Eq. 5** with an additional +0.05s which amounts to $\tau = 0.55s$ for 1 hop. The overall average delay to detect a fully cooperative trusted vehicle increases as a function of the number of tiers in Z_I. It evolves from 5s when $L = 1$ to 11s for $L = 5$. Indeed the periodicity τ of *A-Messages* based on the equality in **Eq. 5** passes to 0.95s for 5 hops. More interestingly, we note that the latency is always close to the average delay required to transmit an average of 9 *A-Messages* about the same event.

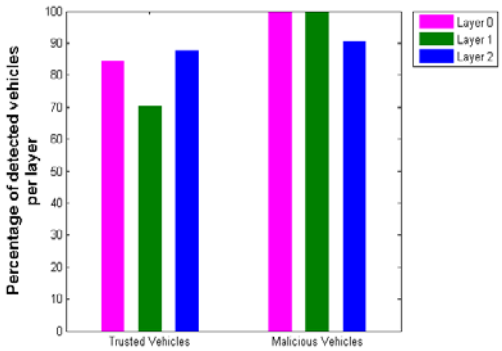


Fig. 6. Percentage of trusted and malicious nodes detected per layer ($F = 1$), $L=3$ and $N_e=5$

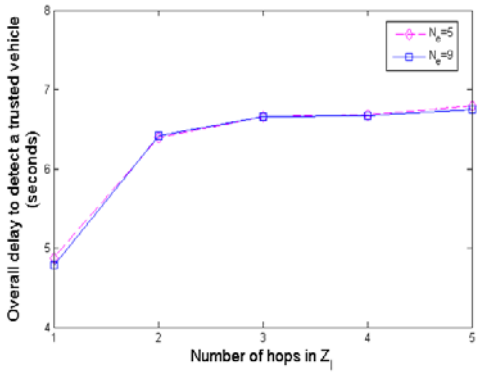


Fig. 7. Average duration to detect a trusted vehicle ($F=1$) vs. number of hops in Z_I and Ne

Fig. 8 portrays the overall (over all monitors) average delay required to detect a malicious vehicle as a function of the number of hops in Z_I and for $F = 1$. The two upper curves relate to $N_e = 5$ and $N_e = 9$ where each malicious vehicle chooses its own P_m randomly. The middle curve relates to $N_e = 5$ with a fixed P_m equal to 0.5, and the bottom curve relates also to $N_e = 5$ but with a fixed P_m equal to 1. First, we clearly notice the dependency of the average delay on P_m. When $P_m = 1$, the average delay is at its lowest as a malicious vehicle is immediately detected upon sending its first non-authentic *A-Message*. In the upper curves, the average delay goes from around *1.5s* for $L = 1$ to around 3s for $L = 2$ due essentially to the increasing value of τ according to **Eq. 5.** Recall also that for layers $L \geq 2$, the detection of a malicious is instantaneous upon its transmission of the first non-authentic *A-Message*. This is clearly portrayed in **Fig. 9**. As a result, the delay decreases when the number of hops in Z_I increases.

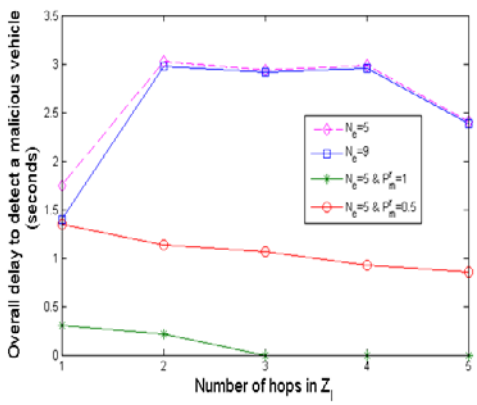


Fig. 8. Average time to detect a malicious vehicle ($F = 1$) vs. number of hops in Z_I and Ne

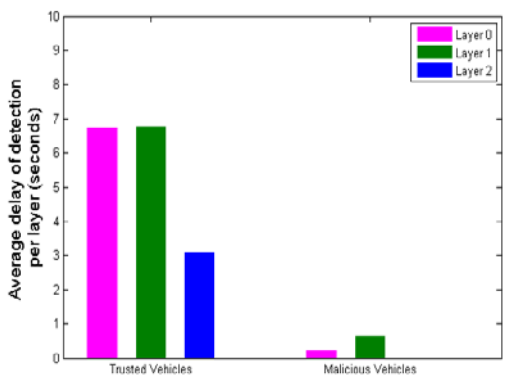


Fig. 9. Average time to detect a trusted malicious vehicle for $F=1$, $L=3$ and $N_e=5$

In contrast to **Fig. 8** portraying the overall average delay among all layers of zone Z_I, **Fig. 9** shows rather the average delay to detect both trusted and malicious vehicles within each layer of zone Z_I for $L = 3$. For trusted vehicles, the average delay for layers 0 and 1 is just under 7 seconds, this is essentially due to the large value of τ when $L=3$. For trusted nodes in layer 2, the average delay is much lower as the transmission of *A-Messages* within this layer does not undergo the waiting period τ_2.

We investigate now the consistency of the trust metrics of vehicle v among all its monitors. Let $T_m(M_j,v)$ be the average trust metric of v computed by monitor j, and $T_m(v)$ be the overall average of the trust metrics computed by all monitors for vehicle v. Let MS_v be the set of all vehicles that have monitored vehicle v. The consistency is here expressed as the variance of the trust metric of vehicle v as given by **Eq. 10**:

$$Var(T_m(v), t_0) = \frac{\sum_{j \in MS_v} \left(T_m^2(M_j, v) - T_m^2(v)\right)_{t_0}}{|\mathrm{MS_v}|} \tag{10}$$

where t_o is the sampling periodicity, M_j denotes the *jth* monitor of v, and $|MS_v|$ represents the cardinality of the set MS_v. We are rather interested in the average trust metric $T_m(v)$ when it is equal to 0, 0.8, 0.9, and 1.

Fig. 10 portrays the overall consistency of the trust metrics for $T_m = 0.8$ and $T_m = 0.9$ as a function of the number of hops in Z_I and the number of events N_e. We do not show the variance of $T_m = 0$ and $T_m = 1$ as they are null for different scenarios which clearly indicates that our proposed DTCF computes indeed exact trust metrics.

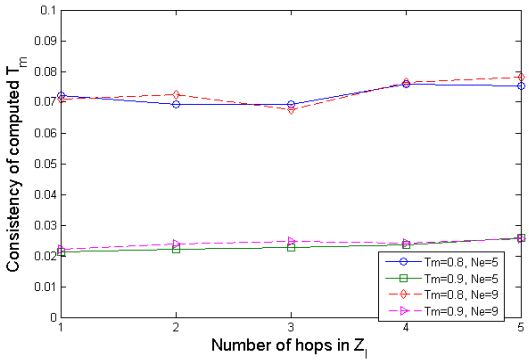


Fig. 10. consistency for T_m=*0.8* and T_m=*0.9* vs. the number of hops in Z_I and N_e

We notice that the consistency of $T_m = 0.9$ is almost stationary at 0.02 with a slight increase for 5 hops. The consistency of $T_m = 0.8$ is about 0.07 until 3 hops then it increases to 0.08 in 4 and 5 hops. The slight increase in the consistency for high numbers of hops is essentially due to the corresponding increase of τ. In summary, even for a large zone of interest Z_I, the trust metrics computed by the different monitors converge towards the same value of any given monitored vehicle.

4.3. Simulation setup of the Highway Scenario

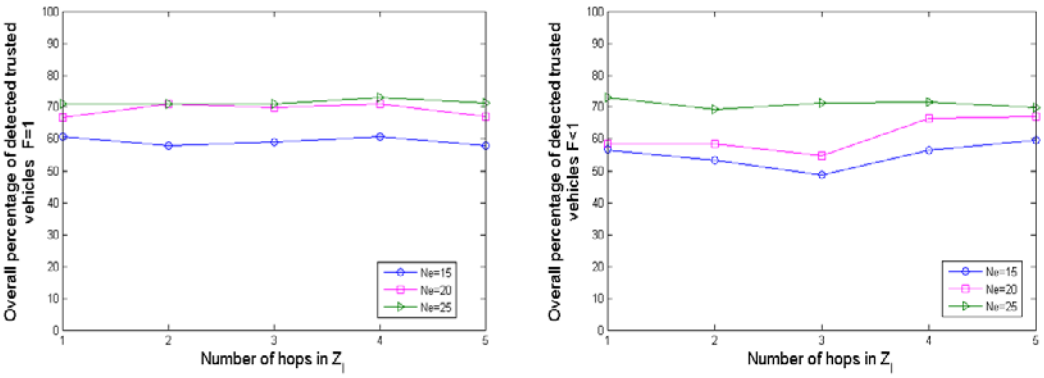
We consider a 20km segment of a highway with two lanes in each direction. The vehicle input (entering) rate per lane is assumed to be 0.35veh/s. Vehicles travel at a maximum speed of 30m/s. Event generators are assumed to be uniformly distributed along the road at both sides. The number of non-authentic event generators is set to around 25% of N_e. Furthermore, we consider two special authentic event generators respectively located at positions x=3000m and x=17000m of the highway segment where vehicles must stop for a period of 5s and then resume their trips. The rest of the simulation parameters are as given in **Table 3**.

4.4. Simulation Results in the Highway Scenario

The same previous performance metrics are investigated for this highway scenario. **Fig. 11a** plots the overall percentage (per monitor over all monitors) of detected trusted vehicles (as per **Eq. 5**) as a function of the number of hops in Z_I and for different values of N_e.

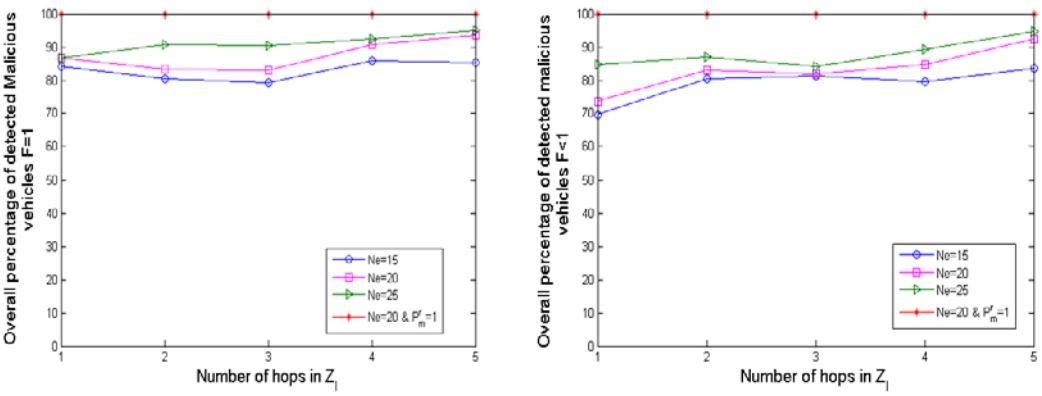
The percentage of detected trusted vehicles for all considered values of N_e remains virtually stable independently of the number of tiers within zone Z_I. However, this percentage depends on the number of events on the road. Indeed, as vehicles speed up to 30m/s in opposite directions, they quickly move away from zones Z_I as well as from several of their encountered monitors. As a result, vehicles acquire intermediate trust levels until monitors and monitored vehicles enter into a new zone Z_I of another event. Both the short encounter durations and the high speed impact the percentage of detected trusted vehicles, though around 72% in the case of $N_e = 25$ is considered very appropriate in such a situation. More interestingly, this percentage is more than sufficient to build a security framework like the one proposed in [10].

On **Fig. 11b**, we portray the overall percentage per monitor of detected trusted selfish vehicles (as per **Eq. 6**). We notice the same result for $N_e = 25$ as in **Fig. 11a** where vehicles are fully cooperative. However for $N_e = 15$ and $N_e = 20$, the percentage of detected trusted selfish vehicles declines slightly as compared to that of the fully cooperative case. Recall that selfish vehicles deny transmitting 20% of *A-Messages*. Moreover, we notice an increase for 4 and 5 hops as in **Fig. 11a** which is essentially due to the entry of vehicles into an overlapping of different zones Z_I which compensates for the non-transmitted messages.



a. *F=1* **b. *F=0.8***

Fig. 11. Overall percentage of trusted vehicles vs. the number of hops in Z_I and N_e



a. *F=1* **b. *F=0.8***

Fig. 12. Overall percentage of declared malicious vehicles vs. number of hops in Z_I and N_e

Let us now consider **Fig. 12a** and **Fig. 12b** portraying respectively the overall percentage per monitor of declared malicious vehicles (**Eq. 7**) and selfish malicious vehicles (**Eq. 8**). First of all, we observe for the case of $P_m = 1$ the excellent full detection capability (100%) of our proposed DTCF in both cases of $F = 1$ and $F=0.8$. This is due to the immediate detection of malicious vehicles upon sending their first non-authentic *A-Messages*. In both **Fig. 12a** and **Fig. 12b**, the percentage of detected malicious vehicles increases slightly with the number of hops. Not detected nodes are vehicles that have adopted small values of P_m. Recall from **Eq. 7** that for $P_m \leq 0.1$ for instance, a vehicle needs to transmit more than 30 *A-Messages* to its monitors in order to be declared malicious. The overall performance can get better had we put a larger number of non-authentic event generators.

In **Fig. 13**, we rather plot the percentage of both trusted and malicious vehicles detected in each layer of the zone of interest Z_I for $L = 3$. 100% of malicious nodes in both layers 0 and 1 are declared. Regarding the percentage of detected trusted vehicles, the same as for the urban scenario, it slightly decreases in tier 1 compared to the other two layers. This is essentially caused by the adopted collecting period τ_2.

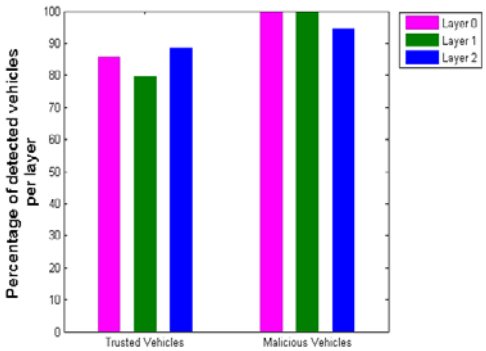


Fig. 13. Percentage of trusted/malicious detected vehicles per layer ($F=1$), $L=3$ $N_e=20$

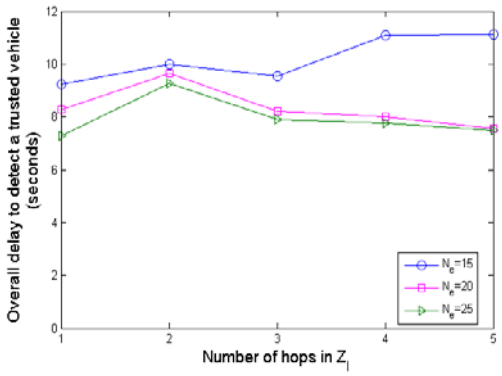


Fig. 14. Average duration to detect a trusted vehicle ($F=1$), $L=3$ and $N_e=20$

Fig. 14 portrays the latency (as per **Eq. 9**) per monitor to detect a trusted vehicle. We remark that the latency depends on the number of hops in Z_I as well as the number of events N_e. We notice also that for a sufficient number of events ($N_e \geq 20$), the latency is rather decreasing as the number of hops in Z_I gets larger than 2. This is mainly due to the overlapping among different zones of interest. As a result, vehicles in upper layers receive more *A-Messages* than those in tiers 0 and 1, and consequently the convergence to the trust state is faster. This fact can be clearly observed on **Fig. 15** portraying the average latency spent separately in each layer to detect trust nodes and to declare malicious nodes for $L = 3$, $F = 1$, and $N_e = 20$. We remarkably notice here the very small delay required to detect malicious vehicles when they all adopt a constant P_m equal to 1. Again, the delay to detect malicious vehicles in tier 2 is null as they are immediately detected upon transmitting their first *A-Message*.

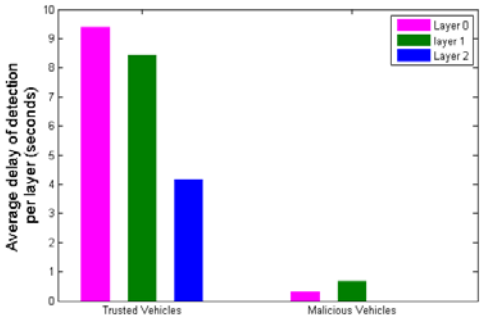


Fig. 15. Average duration to detect a trusted vehicle/malicious vehicle in each layer ($F=1$), $L = 3$ and $N_e = 20$

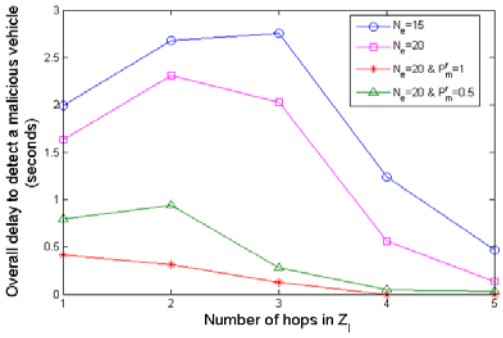


Fig. 16. Average duration to detect a malicious vehicle ($F=1$) vs. N_e and number of hops in Z_I

Fig. 16 portrays the overall average delay to declare a malicious vehicle as a function of the number of hops in Z_I, the number of events and the probability P_m. The upper curve relates to the case of random selection of P_m. As such, some vehicles may choose low P_m values and consequently require more time to be detected as malicious. The lower curve relates to $P_m=1$. In the latter, we obtain the lowest possible delay to detect malicious nodes. This is indeed a very short delay that goes to zero as the number of tiers in Z_I gets larger than 3. Recall when $P_m=1$, a malicious vehicle in any layer but layer 1 is detected immediately upon the transmission of its first non-authentic *A-Message*.

The middle curve shows the average delay to detect a malicious vehicle when $P_m=0.5$ is constantly used by every malicious vehicle. Large delays are only obtained when malicious vehicles are allowed to choose very low values of P_m, otherwise our proposed DTCF insures a very short detection delay less even than 1 second.

We investigate, now, the consistency of the trust metrics of vehicle v as computed by all its monitors. This consistency is expressed by the variance as in **Eq. 10**. In the same way as in the urban scenario, we are here interested in the average trust metric $T_m(v)$ when it is equal to 0, 0.8, 0.9, or 1. Remarkably, we observe from **Fig. 17** that the consistency is virtually independent from both the number of considered events and the number of hops in Z_I. For $T_m(v)$ equal 0 and 1, we get null values indicating the excellent capability of our proposed DTCF to compute exact and consistent trust metrics.

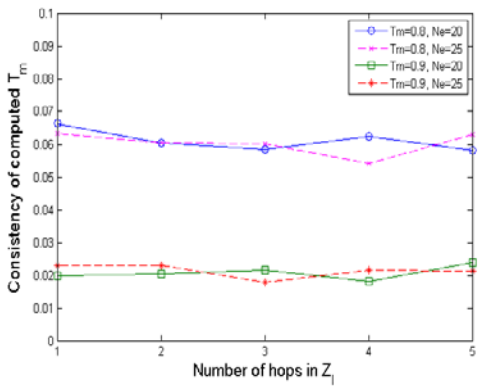


Fig. 17. $Tm = 0.8$ and $Tm = 0.9$ Consistency vs. number of hops in Z_I and N_e

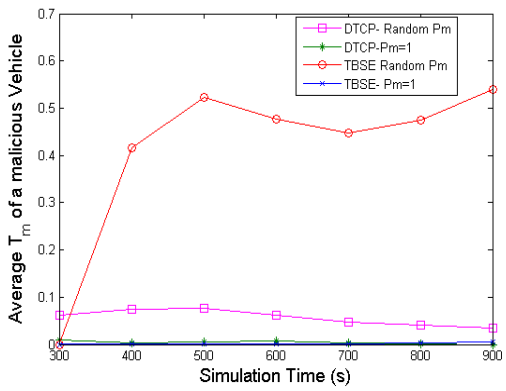


Fig. 18. Average trust metric of a malicious Vehicle

Now, we focus on the evaluation of the average trust metric of malicious vehicles. To this end, we plot in **Fig. 18** the average trust metric computed of malicious vehicles as a function of the simulation time. We compare the average trust metric computed by DTCF to that computed by the TBSE protocol proposed in [6]. For both proposals, we consider the case where a node behaves maliciously in a random and continuous fashion. A random malicious behavior means that the value of P_m is picked randomly in $[0-1[$, otherwise $P_m = 1$ and the malicious behavior is continuous.

We clearly remark from **Fig. 18**, that DTCF and TBSE compute the same value of trust metrics in the case of a continuous malicious behavior. In both curves, the average trust metric of a malicious node is virtually equal to 0. This is an expected behavior since all transmitted data messages are non-authentic. Nevertheless, when the value of P_m is randomly selected in $[0-1[$, we observe the huge difference between our proposal and TBSE. The average trust is around 0.5 for TBSE but less than 0.1 for DTCF.

4.5. Qualitative comparison

We here propose a qualitative comparison between our proposed DTCF and three other proposals [13], [6] and [7]. To this end, we consider the following criteria defined in [2, 8]:

- *Distribution*: as it has been mentioned in section 2, a distributed trust computing framework is more efficient in VANETs than a centralized framework.
- *Scalability*: the number of vehicles in both urban and highway scenarios is rather dynamic and may get very significant; a scalable and evolutionary trust computing framework is then required.
- *Accuracy*: a trust computing model should accurately compute the trust metrics of vehicles. More interestingly, it should allow the detection of all malicious vehicles.
- *Time Scarcity*: in VANETs, the speed of vehicles is significant, the network topology is dynamic, and the encounter periods among nodes are short. As such, it is difficult to build a long experience history between peers. A trust model in VANETs should be able to handle this challenge.
- *Real Time data processing*: the time to react to a given situation in a vehicular environment is very critical and a vehicle should be able to accurately check the trust of the received information as well as the transmitter in real time.
- *Fake event detection*: trust computing frameworks should detect and eventually filter out to filter out non authentic messages [8].

Table 4 presents a qualitative comparison between DTCF and the three relevant approaches [6, 7, 13] using the above criteria along with the use of a recommendation system and the necessity of Road Side Units (RSU).

Table 4. Qualitative comparison between DTCF and existing proposals

	DTCF	R.Shaikh et al. [13]	TBSE [6]	C.Liao et al. [7]
Distribution	x	x	x	
Scalability	x	x		
Accuracy	x			
Time scarcity	x	x		
Real-time data processing	x	x	x	x
Fake event detection	x			
Use of Recommendations	-	-	x	x
Implication of RSU	-	-	-	x

DTCF fulfills all the above criteria despite the fact that it neither necessitates the use of RSUs nor being assisted by a recommendation system. DTCF is fully distributed by design, scalable as it keeps the same performances independently of the mobility scenario and the number of vehicles, and accurate as it detects both malicious vehicles and fake events. R. Shaikh et al.'s proposal [13] is a distributed protocol and verifies the scalability and the real-time data processing criteria, however it stands short in accurately computing the trust metrics as it may yield a high false positive rate. This happens when the signal emanating from the originating vehicle is obstructed by obstructing entities within the network. Recall that they require the alerting message to come directly from its originator. Moreover, their proposal does not allow the detection of fake events. TBSE [6] is a distributed protocol where data is processed in real time. It is also assisted by a built in recommendation system. However, it stands short in accurately computing the trust metrics especially for malicious vehicles. Simulation results portrayed on **Fig. 18** showed an average value of the trust metrics of malicious nodes higher than 0.5. Furthermore, TBSE does not allow the detection of non-authentic (fake) events.

5. Conclusion

In this paper, we proposed a novel dynamic and self-organized trust computing framework to establish accurate trust metrics among vehicles, declare malicious nodes, and filter out fraudulent data. Each vehicle monitors the behavior of its neighbors by inspecting the authenticity of their transmitted messages, and updates their trust metrics accordingly. A tier-based broadcasting scheme is used to disseminate control messages and information about occurring events.

Extensive simulations were conducted to evaluate the performance of our proposed DTCF in terms of the overall average of detected trusted vehicles, the overall average of declared malicious vehicles, the average latency required to detect both trusted and malicious vehicles and the consistency of the computed trust metrics. Despite the high dynamics of VANETs, our proposed DTCF establishes accurate and consistent trust values within very limited delays to detect and declare malicious vehicles as well as to detect trusted vehicles. DTCF allows declaring virtually all malicious nodes and more than 85% of trusted vehicles for all simulated scenarios. The performance of our proposed DTCF depends naturally on the different parameters used such as the number of event generators, the cooperativeness of the different vehicles, the probability of malicious nodes to tamper message contents and the collecting period τ_2 used in layer 1. Further investigations are underway to tune these different parameters and ascertain their impact on the efficiency of the proposed framework.

References

[1] Wex P, Breuer J, Held A, Leinmuller T, Delgrossi L., "Trust Issues for Vehicular Ad Hoc Networks," in *Proc. of the 67th IEEE Vehicular Technology Conference*, VTC'08 Spring, 2800–2804, 2008. Article (CrossRef Link)

[2] Alriyami Q, Adnane A, Smith A.K., "Evaluation criteria for trust management in vehicular ad-hoc networks (VANETs)," *ICCVE'14*, 118–123, 2014. Article (CrossRef Link)

[3] Soleymani S.A., Abdullah A.H., Hassan W.H, Anisi M.H, Goudarzi S, Rezazadeh Baee M.A, Mandala S., "Trust management in vehicular ad hoc network: a systematic review," *EURASIP Journal on Wireless Comm. and Networking 2015*,146, 2015. Article (CrossRef Link)

[4] K.Govindan, P.Mohapatra, "Trust Computations and Trust Dynamics in Mobile Ad hoc Networks: A Survey," *IEEE Communications Surveys and Tutorials*, volume 14, Issue 2, pp.279-298, 2012. Article (CrossRef Link)

[5] Yan Z, Zhang P, Vasilakos AV, "A survey on trust management for internet of things. Journal of Network and Computer Applications," 42(0):120 – 134, 2014. Article (CrossRef Link)
[6] Wei Z, Yu FR, Boukerche A., "Trust based security enhancements for vehicular ad hoc networks," in *Proc. of the Fourth ACM International Symposium DIVANet '14*, ACM: New York, NY, USA, 103–109, 2014. Article (CrossRef Link)
[7] Liao C, Chang J, Lee I, Venkatasubramanian K., "A trust model for vehicular network-based incident reports," *IEEE Wireless Vehicular Communications (WiVeC'13)*, 1–5, 2013. Article (CrossRef Link)
[8] Mármol FG, Pérez GM. TRIP, "A Trust and Reputation Infrastructure-based Proposal for Vehicular Ad Hoc Networks," *Journal of Network and Computer Applications*, 35(3):934–941, 2012. Article (CrossRef Link)
[9] Lassoued I., Bonnin J.M., Belghith A., "Towards an architecture for mobility management and resource control," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC'08)*, 2846-2851, Las Vegas, USA, 2008. Article (CrossRef Link)
[10] Gazdar T, Benslimane A, Belghith A., "Secure Clustering Scheme Based Keys Management in VANETs," in *Proc. of The 73rd IEEE Vehicular Technology Conference*, VTC Spring 2011, Budapest, Hungary, 2011. Article (CrossRef Link)
[11] Gazdar T., Benslimane A., Belghith A., Rachedi A., "A secure cluster-based architecture for certificates management in vehicular networks," *Wiley, Security and communication networks*, 7(3):665–683, 2014. Article (CrossRef Link)
[12] Raya M, Papadimitratos P, Gligor V, Hubaux JP., "On Data-Centric Trust Establishment in Ephemeral Ad hoc Networks," in *Proc. of the 28th IEEE INFOCOM'08*, Phoenix, AZ, USA, 1238 – 1246, 2008. Article (CrossRef Link)
[13] Shaikh RA, Alzahrani AS, "Intrusion-aware trust model for vehicular ad hoc networks," *Security and Communication Networks*, 7(11):1652–1669, 2014. Article (CrossRef Link)
[14] Sharma K, Chaurasia B., "Trust based location finding mechanism in vanet using DST," *Communication Systems and Network Technologies (CSNT'15)*, April 2015. Article (CrossRef Link)
[15] Zadeh LA., "A simple view of the dempster-shafer theory of evidence and its implication for the rule of combination," *AI Magazine*, 7(2):85–90, 1986. Article (CrossRef Link)
[16] Dixit K, Joshi KK, Joshi N., "A novel approach of trust based routing to select trusted location in AODV based vanet: A survey," *Int. Journal of Hybrid Information Technology*, 8(7):335–344, 2015. Article (CrossRef Link)
[17] Patel N.J, Jhaveri R.H., "Trust based approaches for secure routing in vanet: A survey," *Procedia Computer Science*, 45:592 – 601, 2015. Article (CrossRef Link)
[18] Abdelaziz Kerrache C, Lagraa N, Calafate C, Lakas A., "Trouve: A trusted routing protocol for urban vehicular environments," *Wireless and Mobile Computing, Networking and Communications (WiMob'15)*, 260–267, 2015. Article (CrossRef Link)
[19] Sichitiu M, Kihl M., "Inter-vehicle communication systems: a survey," *IEEE Communications Surveys Tutorials*, 10(2):88–105, 2008. Article (CrossRef Link)
[20] Son LH., "Dealing with the new user cold-start problem in recommender systems: A comparative review," *Elsevier Information Systems*, Volume 58, Pages 87-104, 2016. Article (CrossRef Link)
[21] Varga A. The OMNeT++ Discrete Event Simulation System. Proceedings of the European Simulation Multiconference (ESM'2001) 2001.
[22] Behrisch M, Bieker L, Erdmann J, Krajzewicz D. Sumo - simulation of urban mobility: An overview. SIMUL 2011, Barcelona, 63–68, Spain, 2011.

Dr. Tahani GAZDAR received her PhD, Master of Science degree and Engineering degree in Computer Science all from the National School of Computer Sciences (ENSI), University of Manouba, Tunisia, respectively in 2015, 2010 and 2009. She is currently an Assistant Professor at the Faculty of Computing and Information Technology, University of King Abdul-Aziz, KSA. She is also a researcher at the HANA Lab., University of Manouba, Tunisia. Her current research interests include clustering, keys management and trust computing systems in vehicular ad hoc networks.

Dr. Abdelfettah Belghith received his Master of Science and his PhD degrees in computer science from the University of California at Los Angeles (UCLA) respectively in 1982 and 1987. He is since 1992 a full Professor at the National School of Computer Sciences (ENSI), University of Manouba, Tunisia. He is currently on a sabbatical leave at King Saud University, Saudi Arabia. His research interests include computer networks, wireless networks, multimedia Internet, mobile computing, distributed algorithms, systems and information security, simulation and performance evaluation. He runs several research projects in cooperation with other universities, research laboratories and research institutions. He is the Past chair of the IEEE Tunisia section, the chair of the IEEE ComSoc and VTS Tunisia Chapters, and the Director of the HANA Research Laboratory (www.hanalab.org) at the National School of Computer Sciences. He published more than 300 research papers in international journals and conference proceedings

Dr. Ahmad S. AlMogren obtained his PhD degree in computer sciences from Southern Methodist University, Dallas, Texas, USA in 2002. Previously, he worked as an assistant professor of computer science and the head of the scientific council at Riyadh College of Technology. He also served as the dean of computer college and the head of the council of academic accreditation at Al Yamamah University. Presently, he works as an associate professor and the vice dean for development and quality at King Saud University. His research areas of interest include networking, security, mobile computing and data consistency.