Human-centric Computing and Information Sciences

RESEARCH Open Access

TSME: a trust-based security scheme for message exchange in vehicular Ad hoc networks

Ryma Abassi[1*], Aida Ben Chehida Douss[1] and Damien Sauveron[2]

*Correspondence: ryma.abassi@supcom.tn
[1] Digital Security Research Lab, Higher School of Communication of Tunis, SUP'Com, University of Carthage, Tunis, Tunisia
Full list of author information is available at the end of the article

Abstract

A Vehicular Ad hoc NETwork (VANET) is a self-organized network formed by connected vehicles, which allows the exchange of useful traffic information in a timely manner. In such a context, evaluating the reliability of transmissions is vital. Trust can be used to promote such healthy collaboration. In fact, trust enables collaborating vehicles to counter uncertainty and suspicion by establishing trustworthy relationships. The main contribution of this paper is the proposition of a trust-based security scheme for message exchange in a VANET called TSME. Because of VANET characteristics, including dynamicity and high speed, we first proposed a VANET Grouping Algorithm (VGA); a suitable clustering algorithm organizing the network into groups with elected Group-Heads. Second, built on the VGA, we defined our trust management scheme dealing with vehicles' reputations. Finally, we proposed a formal specification of the scheme using an inference system, and conducted a formal validation to assess its completeness and soundness rather than conducting simulations where some potentially rare conflicting or malfunctioning situations might not be detected. Soundness was proven by showing that there were no conflicts in our scheme, and completeness was established by assessing that all potential situations could be handled. The results obtained showed that our scheme for evaluating the veracity of exchanged messages is formally sound and complete.

Keywords: VANET, Security, Mobility, Embedded intelligence, IoT, Clustering

Introduction

A Vehicular Ad hoc NETwork (VANET) is a special case of a Mobile Ad hoc NETwork (MANET), where the nodes are vehicles equipped with On-Board Units (OBUs) [1]. These vehicles can directly inter-communicate, or communicate through routers called Road Side Units (RSUs). The first case is called Vehicle-to-Vehicle communication (V2V), while the second case is Vehicle-to-Infrastructure communication (V2I). In both cases, Trusted Authorities (TA) control the whole network. VANETs are mainly used to improve traffic security (such as traffic services, alarms and warning messaging) and efficiency. In this context, a security problem can have disastrous consequences since an attacker may have the ability to broadcast false alerts and/or messages for its own benefit.

Trust can be used to promote healthy collaboration by enabling collaborating vehicles to counter uncertainty and suspicion by establishing trustworthy relationships [2, 3]. Because of the importance of the challenge, trust is associated with an abstract system that helps decision making, referred to as Trust Management (TM) [4]. Hence our main proposal is a trust-based security scheme for message exchanges in VANETs. Because of VANET characteristics such as dynamicity and high speed, this scheme is built upon a new grouping algorithm called the VANET Grouping Algorithm (VGA), which organizes vehicles into groups characterized by a Group-Head (GH) and member nodes. When grouping vehicles into multiple groups, the system becomes scalable by having message relay done between GHs instead of between two neighboring peers. Grouping is generally deployed using two phases: setup and maintenance [5]. In the first phase, some nodes are chosen to act as coordinators (GHs) and each GH is associated with a number of member nodes, the whole making one group. Because the network topology changes over time, mainly due to displacement, failure, arrival, or departure of a node, a maintenance phase is required to update the group's organization. The next goal in the definition of a security architecture for a VANET is its validation. To obtain a comprehensive assessment, we decided to conduct a formal validation rather than simulations, where some potentially rare conflicting or malfunctioning situations might not be detected. Hence, we proposed an inference system for handling the VGA maintenance phase. Next, we formally validated the proposed inference system according to two main properties: (1) soundness, which ensures that the proposed model reacts correctly; and (2) completeness, which determines that the model is complete—i.e. no other situations can be found.

Contributions

This paper proposes TSME, a dedicated trust-based scheme to secure message exchange in vehicular ad hoc networks, which is formally verified.

The salient contributions of this paper are as follows:

1. Proposing a new grouping algorithm known as VGA to organize the VANET into scalable groups and deal with specific vehicular network characteristics, including dynamicity (i.e. vehicle arrival and departure), high speed and other salient characteristics.
2. Proposing a trust management scheme, built upon VGA, to handle message exchange into the VANET and deal with vehicles' reputations.
3. Formally validating the completeness and soundness of TSME; i.e., the whole proposal including VGA and the trust management scheme, with regard to the defined specification using an inference system.

Structure of the paper

Section 2 reviews some existing studies dealing with trust in VANETs. Section 3 introduces our clustering algorithm for VANETs, called the VANET Grouping Algorithm.

Section 4 provides a description of the proposed trust management scheme built upon the VGA. Section 5 details the formal specification of the scheme using an inference system and elaborates the formal validation procedure for assessing soundness and completeness. Section 6 concludes this paper.

Related work

Several reputation systems have been proposed for peer-to-peer networks [6, 7], ad hoc networks [8–10], wireless sensor networks [11, 12] and Internet of Vehicle [13–15]. However, these systems cannot be applied to VANETs in their existing forms they do not consider the main VANET characteristics: dynamicity and high speed.

In [16], the authors proposed a beacon-based trust management system, called BTM, which aims to prevent internal attackers from sending false messages in privacy-enhanced VANETs. A vehicle can use not only direct or indirect event messages, but also beacon messages to construct trust relationships in order to distinguish trustworthy event messages.

Al Falasi and Mohamed [17] proposed a "similarity-based trust management system for detecting fake safety messages in VANETs". Their scheme uses similarity-based trust relationships to detect false safety event messages from abnormal vehicles in VANETs. Moreover, it reacts to safety event claims made by a vehicle and predicts that the source vehicle will react to a truthful safety event report.

Zhang et al. [18] proposed a "trust-modeling framework for message propagation and evaluation in VANETs". In their model, a vehicle can decide whether to trust a message or not by evaluating others' opinions. However, such decentralized trust systems, relying on interactions with neighbors, are not practical in the highly dynamic environment of a VANET.

In [19], a trust-extended authentication mechanism (TEAM) was proposed. It is a decentralized, lightweight authentication scheme for highly dynamic VANETs, ensuring integrity and non-repudiation and thus increasing the vehicles' confidence in communications. However, TEAM does not deal with the reliability of the message data itself.

In [20], the authors proposed a trust-based relay selection scheme, called PTRS, which, based on Dirichlet distribution, differentiates the trust levels of the vehicles, while preserving robustness. The PTRS scheme is robust against some attacks, such as packets analysis attacks, reputation link attacks, packets dropping attacks [21], and fake reputation attacks.

Recently, Das et al. [22] proposed schemes for finding the trusted location of a vehicle. Firstly, the trust percentage of the information is computed using the responses received from vehicles. Based on this, the trust percentage of the information is calculated on the basis of the number of requests and the number of positive responses. Each vehicle giving a positive response about the information is rewarded with points for providing true information, thus enabling calculation of the trustworthiness of each node present in the network. In the second case, when the trust is below 50%, instead of going to an RSU or a TA, the vehicle will check the trustworthiness of each node in the network and accept the response of the most trustworthy node.

Mahmood et al. [23] proposed a hybrid trust management scheme to identify malicious vehicles and to prevent them from being elected as the GH. Their scheme encompasses a composite metric (i.e., trust values assigned to the vehicles coupled with their resource availability) for GH and proxy GH selection via intermittent elections. This approach helps to form trustworthy and resource-efficient vehicular networks.

In [24], Sugumar et al. proposed a trust-based authentication scheme for cluster-based VANETs. The vehicles are clustered, and the trust degree of each node is estimated. The trust degree is a combination of direct trust degree and indirect trust degree. Based on this estimated trust degree, cluster heads are selected. Then, each vehicle is monitored by a set of verifiers, and the messages are digitally signed by the sender's private key and encrypted using a public key (keys are distributed by a trusted authority and decrypted at the destination). This verifies the identity of the sender as well as the receiver, thus providing authentication to the scheme.

Hasrouny et al. [25] proposed a Trust Model for VANETs. It is a combination of centralized and distributed cooperation between vehicles and infrastructure to achieve the selection of the trustiest node as the GH. This proposed model is based on different metrics to analyze the behavior of the vehicles in the group while preserving the privacy of the participants and maintaining low network overheads.

Hao et al. [26] proposed the concepts of local trust and global trust to indicate the local and global trust relationships between vehicles. They adopted the PageRank algorithm [27], used to rank web pages to calculate the global trust of vehicles.

To summarize, the trust management models discussed above are not dynamic enough to cope with VANETs' characteristics. Some of them [16, 17] were proposed to deal with a specific type of message while others [3, 6, 19] used decentralized trust systems, relying on interactions with neighbors, which is not practical in the highly dynamic environment of a VANET. Several works [4, 10, 15, 23] tried to cope with a specific type of attack, which can be a limitation whereas several others were designed to handle a specific challenge such as authentication [24], privacy [20, 25] or localization [18, 22]. Therefore, unlike other solutions, we propose an adaptive trust-based security scheme for message exchange in VANETs based on VGA, our new clustering algorithm designed to handle the dynamicity such networks. It is described in Section 3.

VGA: a VANET grouping algorithm

Our overall proposal, TSME, a trust-based security scheme for message exchange in VANETs, depicted in Fig. 1, comprises two parts: (a) our VANET Grouping Algorithm (VGA), which is described below, to handle VANET characteristics, including dynamicity and high speed; and (b) a trust management scheme described in Section 4 built upon VGA.

VGA is composed of three phases: pre-processing, setup and maintenance. In the preprocessing phase, initial reputations are set, whereas in the setup, groups are formed. The maintenance phase is used to handle the mobility of the vehicles and update the formed groups.

The trust management scheme uses the formed groups as well as the veracity of exchanged event messages to increase or decrease vehicles' reputations via the reputation update module. The veracity of the event messages is computed based on three indicators: L_c the location closeness, F_c the number of forwarders and T_c the time closeness. More details about these indicators are provided later in this section.

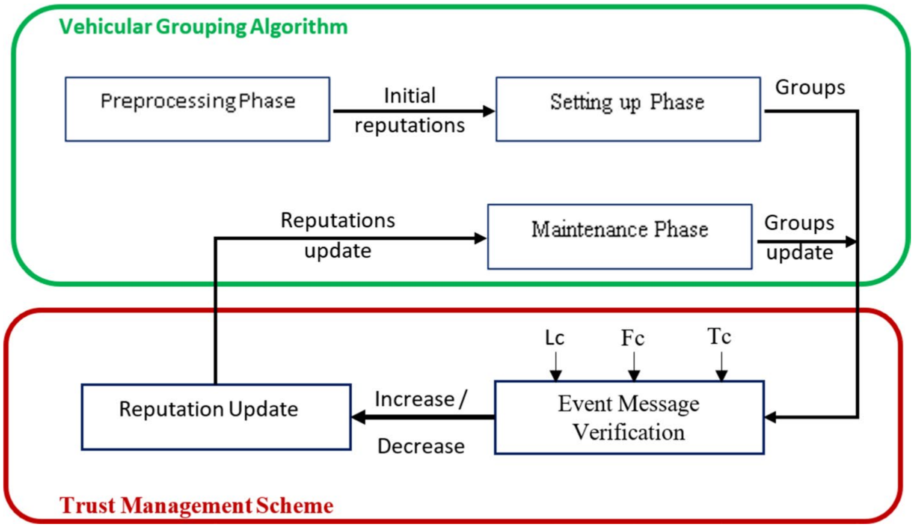

Fig. 1 The proposed Trust-based Security Scheme for Message Exchanges in VANETs

According to [28], the general procedural flow of a clustering algorithm is described in five steps: neighborhood discovery; cluster head selection; affiliation; announcement; and maintenance. Hence, VGA is based on similar steps with some additional ones needed to fit our specific need: trust management.

Assumptions

Our algorithm is based on the following assumptions:

- OBUs periodically broadcast single-hop beacon messages containing (at least) position, velocity and direction.
- Each vehicle generates its keys, sends a certificate signing request to the TA and retrieves a properly signed certificate as proof.
- All the vehicles' clocks are synchronized.

Notations used are depicted in Table 1.

Pre-processing phase

During this phase, initial reputations are assigned to vehicles based on some exchanged information (called credentials). A credential measures the level of trust that we can attribute to the node during the set-up phase and when the node is entering a new cluster or moving from one cluster to another. Hence, a classification of these credentials into three levels of sensitivity is presented and a set of negotiation policies to be used during the negotiation process is defined. Our negotiation approach allows the nodes to provide more information about themselves in order to increase their degree of trust. Here, we highlight that we use a classification similar to [2], which is used to define a XeNa negotiation framework [29]. When the node is able to provide more sensitive information about itself, its trust level is enhanced. Hence, the following resources will

Table 1 Used notations

Notations	Meaning
Pr_{v_i}	The private key of vehicle v_i
Pu_{v_i}	The public key of vehicle v_i
M_b	The beacon message
M_{ack}	The acknowledgment message
E	An encryption function
$sign_M$	The signature generated for a given message M. It is calculated by encrypting the message using the encryption function E and the private key Pr_{v_i}
(x_i, y_i)	The position of vehicle v_i
G_{id}	A group identifier
GH_{id}	The identifier of the Group-Head
ts	A timestamp
MP_{TA}	The public key of the trusted authority
L_c	Location closeness
T_c	Time closeness
F_c	Number of forwarders
Tab_N	The neighbors table
rep_{v_i}	The initial reputation of vehicle v_i
Id_{v_i}	The identifier of the vehicle v_i
dId_{v_i}	The identity of the driver of vehicle v_i
s_{v_i}	The reputation score of vehicle v_i representing its reputation during the setting up phase
VS	The veracity score representing the trustworthiness of a given message
d_{v_i}	The direction of vehicle v_i
c_{v_i}	The manufacturer of vehicle v_i
ph_{v_i}	The photo of vehicle v_i
p_{v_i}	The position of vehicle v_i
vel_{v_i}	The velocity of vehicle v_i
$s - flag$	A state flag set to 'B' or 'H' to indicate whether a vehicle is **B**enevolent or **H**ostile
$Verify(sign_M)$	A boolean function verifying a signature

be considered during the negotiation process: driver identity, direction, identifier, photo, position and manufacturer. These parameters are classified as follows:

- Level 1 (most sensitive): driver identity, direction.
- Level 2 (normally sensitive): identifier, photo, velocity.
- Level 3 (less sensitive): position, manufacturer.

Two different negotiation strategies are defined: (1) vehicles send to the GH the set of all their information in order to get the highest level of trust; (2) vehicles prefer to protect their private information and preserve their privacy preferences by sending only less sensitive information.

Table 2 presents the negotiation policies corresponding to the different resources. Each vehicle builds trust in other vehicles gradually by gathering information related to the other vehicles. For less sensitive resources no negotiation policies are defined. For other resources, a negotiation policy is considered based on the received information. For instance, to be able to get the driver identity of vehicle v_i, vehicle v_j must give its identifier Id_{v_j}, direction d_{v_j} and driver identity dId_{v_j}.

Table 2 Negotiation policies

Resources	Notation	Negotiation policies
Position	p_{v_i}	$P1(v_i): -$
Manufacturer	c_{v_i}	$P2(v_i): -$
Identifier	Id_{v_i}	$P3(v_i): p_{v_j} \wedge c_{v_j}$
Photo	ph_{v_i}	$P4(v_i): p_{v_j} \wedge c_{v_j} \wedge Id_{v_j}$
Velocity	vel_{v_i}	$P5(v_i): p_{v_j} \wedge c_{v_j} \wedge vel_{v_j}$
Driver identity	dId_{v_i}	$P6(v_i): Id_{v_j} \wedge d_{v_j} \wedge dId_{v_j}$
Direction	d_{v_i}	$P7(v_i): vel_{v_j} \wedge dId_{v_j} \wedge d_{v_j}$

Once the negotiation procedure between vehicles is completed, initial reputations are affected as depicted by Algorithm 1 as follows: if the trust level is 1, the reputation value is initialized to 3, if it is 2, reputation is initialized to 2 and when the trust level is 3, reputation is initialized to 1.

Algorithm 1 Pre-processing Phase: phase 1 of VGA

```
Data: a given vehicle $v_i$ and all vehicles $v_j$ in its coverage area
Result: $rep_{v_i}$: initial reputation of the vehicle
begin
    foreach $v_j$ do
        foreach $v_i$ (in coverage area of $v_j$) do
            result = negotiation-module($v_i$,$v_j$)
            if result == Level 1 then
                $rep_{v_i}$= 3
            else if result == Level 2 then
                $rep_{v_i}$= 2
            else
                $rep_{v_i}$= 1
            end
        end
    end
end
```

Setting up phase

Once the pre-processing phase is completed, the setting up phase is triggered in order to discover the neighborhood, select a GH, and form the groups. It is based on six steps: (1) the exchange of signed beacon messages; (2) the reception of these messages; (3) the sending of acknowledge messages; (4) the reception of acknowledgement messages; (5) the election of GHs; and (6) constitution of the group. These steps are detailed in the following and the whole process is described by Algorithm 2. Exchanged messages during this phase are depicted in Table 3.

Sending of beacon messages

The beacon message, M_b, is structured as follows:

$$M_b \equiv < Id_{v_i},\ (x_i, y_i),\ vel_{v_i},\ d_{v_i},\ ts >$$

Table 3 Exchanged messages

Message	Meaning
$M_b(Id_{v_i},\ (x_i, y_i),\ vel_{v_i},\ d,\ ts)$	This message is broadcast by the vehicle identified by Id_{v_i} to all its neighbors to notify them of its location (x_i, y_i), its velocity vel_{v_i} and its direction d ts is the timestamp associated with this message
$M_{ack}(Id_{v_j},\ s_{v_j},\ ts)$	The acknowledgment message is sent by a vehicle identified by Id_{v_j} to a vehicle v_i to notify it of its reputation score s_{v_j} ts is the timestamp associated with this message
$M_{GH}(GH_{id},\ G_{id},\ Members,\ ts)$	The GH message is sent by a GH vehicle to all its group members to inform them of its identity GH_{id}, the group identifier G_{id}, and the set of members *Members* belonging to this group. ts is the timestamp associated with this message
$M_{join}(Id_{v_i},\ GH_{id},\ ts)$	The join message is broadcast by a member vehicle identified by Id_{v_i} to inform other members that it is going to join the cluster whose GH's identifier is GH_{id}. ts is the timestamp associated with this message
$M_{Blacklist}(GH_{id},\ Id_{v_i},\ ts)$	A Blacklist message is sent by the GH GH_{id} in order to inform the RSU that a vehicle identified by Id_{v_i} is acting maliciously. ts is the timestamp associated with this message
$M_{warning}(Id_{v_{id}},\ s-flag,\ ts)$	The warning message is sent by the RSU to GHs in order to inform them about a misbehaving *s-flag* = *'B'* vehicle identified by Id_{v_i} The same message can be send by the RSU to GHs in order to inform them about a rehabilitated *s-flag* = *'H'* vehicle identified by Id_{v_i} ts is the timestamp associated with this message
$M_{event}(Id_{v_i},\ type,\ (x, y),\ l,\ t_r)$	An event message is sent by a vehicle identified by Id_{v_i} to its GH to report an observed event by indicating its type *type* ∈ {*accident, road liberation, traffic information*}, (x, y) the coordinates of the event's location, l the location of the vehicle when it generated the message, and t_r the reporting time

where Id_{v_i} is the vehicle identifier sending the message, (x_i, y_i) designates the location of vehicle v_i, vel_{v_i} corresponds to the velocity of the vehicle, d_{v_i} is the vehicle direction and ts is the time stamp associated with the message.

Initially, each vehicle v_i uses its private key Pr_{v_i} to generate $sign_{M_b}$ a signature for M_b, the beacon message, as follows: $sign_{M_b} = E(Pr_{v_i}, M_b)$.

Then the vehicle v_i broadcasts $(M_b||sign_{M_b})$.

Reception of beacon messages

Each vehicle v_j receiving the signed message from v_i, verifies it using the public key Pu_{v_i}, which is verified using MP_{TA}, the public key of the TA, as well as the vehicle's identifier Id_{v_i}. If the verification is successful, then a second verification is made: the ability of the vehicle to belong to the same group. Hence, vehicle v_j verifies the direction d_{v_i} of the sending vehicle v_i, then its proximity (L_c), as formalized by Eq. 1.

$$L_c = \begin{cases} \frac{1}{x_i + y_i} & if\ (x_i - x_j)^2 + (y_i - y_j)^2 < \Delta^2 \\ 0 & otherwise \end{cases} \tag{1}$$

v_j is used as the origin $(x_j = 0,\ y_j = 0)$. Any vehicle located at $(x,\ y)$ around the event position within a radius of Δ can be trusted with a level of confidence that decreases with increases in $(x_i + y_i)$. The value of Δ can be fixed initially.

If these checks are successful, vehicle v_j computes its reputation score s_{v_j} as shown in Eq. 2 and sends an acknowledgement message $(M_{ack}\|sign_{M_{ack}})$ to v_i. Otherwise, the score is set to a value of 0.

$$s_{v_j} = \begin{cases} vel_{v_j} \times rep_{v_j} & if\ L_c \neq 0\, and\, same\, direction \\ 0 & otherwise \end{cases} \tag{2}$$

This score depends on the reputation of the vehicle as well as its velocity. Due to the dynamicity of the network, a trustworthy vehicle must have a good reputation and a similar velocity to other vehicles in order to communicate with them. Otherwise the vehicle will be left behind or will overtake other vehicles and communication with it is not of interest.

Sending of acknowledgement messages

An acknowledgement message M_{ack} is structured as follows:

$$M_{ack} \equiv < Id_{v_j},\, s_{v_j},\, ts >$$

where Id_{v_j} is the identifier of the vehicle sending the message, s_{v_j} is its computed reputation score and ts is the time stamp associated with the message.

Reception of acknowledgement messages

Each vehicle v_i receiving an acknowledgement message (from v_j for instance), verifies the vehicle's existence by comparing its identifier with the one on the certificate, to avoid flooding attacks based on random identifiers. If confirmed, the message is added to its neighbors table Tab_N. This table is maintained by each node and contains, for each vehicle, its initial reputation rep_{v_j} as negotiated in the pre-processing phase; its reputation score s_{v_j}; its membership, which is a Boolean indicating whether the vehicle belongs to a current group; and a state flag $s-flag$ indicating whether the vehicle is **B**enevolent or **H**ostile.

Group-head election

The next step in this phase is the election of the Group-Head (GH). Each vehicle compares its reputation score with the received scores and the vehicle having the greatest score sends a GH Message M_{GH} to all its members to inform them of the creation of the cluster and the election of the GH as follows:

$$M_{GH} \equiv < GH_{id},\, G_{id},\, Members,\, ts >$$

where GH_{id} is the identifier of the GH, G_{id} is the identifier of the group given by the elected GH, $Members$ corresponds to the set of vehicles belonging to this group and ts is the time stamp associated with the message.

Group constitution

Each member that receives the GH message from a GH vehicle replies with a M_{join} message to confirm its role as a member node.

Each vehicle then creates a group list $GL = (G_{id},\, GH_{id},\, Members)$ containing the elected GH as well as the member vehicles belonging to the group identified by G_{id}.

$$M_{join} \equiv < Id_{v_i},\, GH_{id},\, ts >$$

```
Algorithm 2 Setting up Phase: phase 2 of VGA
Data: $v_i$: a vehicle; $M_b$: Beacon message; $M_{ack}$: ACK message; $TA$: Trusted Authority;
      $MP_{TA}$: TA's Public key; $Tab_N$: Neighbors Table
Result: Formed groups
begin
    $v_i$ : generate($Pr_{v_i}$, $Pu_{v_i}$)
    $v_i$ -> TA: Certificate signing Request
    $v_i$ <- TA: signed certificate
    $v_i$ : $sign_{M_b}$ = E($Pr_{v_i}$, $M_b$)
    $v_i$ -> *: <$M_b || sign_{M_b}$>
    $v_j$: <$M_b || sign_{M_b}$>
    $v_j$: $Pu_{v_i}$=E($MP_{TA}$, $Id_{v_i}$)
    $v_j$: if Verify($sign_{M_b}$)== True then
        direction = get direction($M_b$)
        if same direction then
            $L_c$=calculate location closeness
            if $L_c$ <> 0 then
                Calculate reputation score $s_{v_j}$
                  $M_{ack}$ = <$Id_{v_j} || s_{v_j} || ts$>
                  $sign_{M_{ack}}$ = E($Pr_{v_j}$, $M_{ack}$)
                  $v_j$ -> $v_i$: <$M_{ack} ||$ $sign_{M_{ack}}$>
            end
        else
            ignore $M_b$
        end
    end
    foreach $v_i$ receiving $M_{ack}$ do
        $v_i$: $Pu_{v_j}$=E($MP_{TA}$, $Id_{v_j}$)
         if Verify($sign_{M_{ack}}$) == True then
            add ($Id_{v_j}$, $Tab_N$)
        end
    end
    foreach $v_i$ receiving $s_{v_j}$ do
        compare $s_{v_i}$ and $s_{v_j}$
        if $s_{v_i}$ is the greatest then
            $v_i$ -> Members: $M_{GH}$ = <$GH_{id} || G_{id} ||$Members$|| ts$>
        end
    end
    foreach $v_i$ in Members do
        $v_i$ -> GH: $M_{join}$ = <$Id_{v_i} || GH_{id} || ts$>
    end
end
```

Maintenance phase

The maintenance phase reacts to all topology changes that may occur in the VANET, such as the departure of a vehicle or the arrival of a new vehicle. In this section, the procedures relative to these two kinds of topology change are presented.

Vehicle departure

Two cases are possible: (1) A member vehicle quits the group; and (2) The GH quits the group.

When the GH detects a member vehicle departure, it removes the vehicle from its Tab_N and sends a new M_{GH} informing the rest of the members of this change.

When the closest vehicle to the GH detects the departure of the GH, it informs other members, and the member vehicle having the highest reputation score s sends a new M_{GH} informing other members that a new GH has been elected.

Vehicle arrival

When the GH receives a beacon message from a new vehicle v_{new}, it simply:

1. Verifies its signature,
2. Negotiates its initial reputation $rep_{v_{new}}$,
3. Adds v_{new} to the Tab_N,
4. Sends a M_{GH} to all the group members in order to inform them that a new vehicle belongs to their group.

Trust management scheme

Our main objective of this section is to propose a trust management scheme for VANETs, as depicted in Fig. 1. Hence, the first stage was the design of a trust-suitable grouping algorithm for VANETs while the second stage is dedicated to the trust management scheme.

Our proposal is based on the following assumptions:

1. Vehicles are exchanging event messages.
2. Each vehicle has a reputation.
3. Reputations are between − 3 and 3. A negative reputation is a synonym for a malicious vehicle.
4. Reputation is maintained through direct observations, as well as reputation messages exchanged with other vehicles.
5. Only GHs maintain reputation tables.
6. Active GHs are safe; i.e. cannot behave maliciously.

When a vehicle needs to communicate with another vehicle or a group of vehicles in order to declare an incident and/or request road liberation, a V2V warning propagation is used. A vehicle observing an event sends an event message M_{event} to its GH as follows:

$$M_{event} \equiv < Id_{v_i},\ type,\ (x,y),\ l,\ t_r >$$

The GH, on receiving such an alert, verifies it by calculating a veracity score VS, on the basis of which the message can be forwarded or stopped. The veracity score, VS, is defined as follows:

$$VS = (Lc + Tc) * Fc \tag{3}$$

where Lc is the location closeness, as defined by Eq. 1, Tc is the time closeness representing the freshness of the reported event and formalized in Eq. 4, and Fc is the forwarding chain closeness, estimating the number of vehicles that have forwarded the reported event, and formalized in Eq. 5.

$$T_c = \begin{cases} 1 - \frac{1}{|t_r - t_e|} & if\ |t_r - t_e| < \delta_t \\ 0 & otherwise \end{cases} \tag{4}$$

Where the time of occurrence of the event t_e is used as the origin, t_r corresponds to the reporting time and the value of δ_t is fixed initially.

$$F_c = \begin{cases} \frac{1}{n}\ if\ n < \delta_n \\ 0\ \ otherwise \end{cases} \tag{5}$$

Where n is the number of forwarders and the value of δ_n is fixed initially.

More precisely, this score satisfies the following hypotheses:

- The closer the sender is to the event location, the higher is the veracity score.
- As time closeness decreases, the veracity score decreases.
- As the number of senders increases, the F_c decreases and consequently, the score decreases. In fact, the greater the number of vehicles that have forwarded the reported event, the higher is the probability of a modification of the event or the loss of it, e.g. due to a malevolent node.
- If $VS > 0$, the message is considered trustworthy, the reputation score is increased by $+0.2$, is added to the message, and is forwarded. Otherwise, it is seen as untrustworthy and is simply stopped.

Whenever the score VS is zero, the node reputation score is decreased by 1. It is worth noting that values +0.2 and -1 are derived from our previous work [30]. Once it reaches zero, the vehicle is blacklisted and the GH sends a blacklist message $M_{Blacklist}$ to the RSU as follows:

$$M_{Blacklist} \equiv < GH_{id}, Id_{v_i}, ts >$$

The RSU then informs other RSUs and the TA about this misbehaving vehicle. Each RSU receiving this message informs the GHs under its control by sending them a warning message as follows:

$$M_{warning} \equiv < Id_{v_i}, s - flag = `H', ts >$$

Each GH receiving this message notifies it in its neighbors table and each time a message is sent from this vehicle, it is simply ignored.

We also propose a rehabilitation mechanism enabling malicious nodes to change their behavior so they can rejoin the system. Each GH monitors the behavior of any member nodes detected as malicious. If the malicious node subsequently behaves well, its reputation score is incremented by $+0.1$ using the reputation module, until it reaches the neutral value of 0. Once reached, the rehabilitated node is removed from the blacklist.

Formal specification and validation

The VGA can malfunction due to conflicts between exchanges or lack of necessary functionality in messages or in the scheme phases presented in Section 3. Hence, it is necessary to validate it prior to implementation. The rest of this section is divided into two parts. First, the VGA maintenance phase is specified using a formal and automated method referred to as an inference system, based on the use of logical rules; i.e. a function that takes premises, analyses their applicability and returns a conclusion. Second, a validation task is performed using the proposed inference system. According to [31], validating a model can be done by showing that this model is free of conflict or lack of functionality in the proposed message exchange. Specifically, two main properties have to be considered as proposed in [31, 32]: (1) soundness, by checking that a topology change does not have any influence on clusters; and (2) completeness, by assessing whether the proposed inference system handles all possible situations. In the next subsection, we describe the inference systems for the proposed VGA maintenance algorithms.

Preliminaries

The proposed inference system is based on the following assumptions:

- The VGA set-up phase is already complete. The VANET is organized into groups with members and elected GHs. Each vehicle has a unique role (member or GH) and belongs to only one group.
- Vehicles perform the pre-processing phase periodically; i.e. each vehicle exchanges credentials with its neighbors and initial reputations are established for each one of them.
- The proposed inference system is triggered in three cases: (1) When a novel vehicle arrives in the VANET; (2) when a vehicle moves from its group to another group; and (3) when a member or a GH fails.
- The inference system stops when all new, moving or failing vehicles have been handled.

Formal specification

In this section, the proposed inference system handling the changes that could occur in a VANET topology is presented in Fig. 2. Table 4 summarizes the notations used.

The rules of the system, known as inference rules, apply to couples (N, GL) whose first component N is a set of couples (a, b), where a denotes a new, moving or failing vehicle in the VANET and b denotes the vehicle detecting the arrival or the failure of node x. The second component, GL, represents the initial set of groups generated after the VGA setting up phase. Groups belonging to the GL set are composed by a couple $(\{V_{GH}\}, Members)$, where V_{GH} is the elected vehicle head and $Members$ is the set of vehicles belonging to the same group as V_{GH}. Three inference rules are proposed. $VM_{Failure}$ and $VGH_{Failure}$ are concerned with existing members or GH failures. $V_{Arrival}$ addresses the case of the arrival of new vehicles in the VANET or the

$$init \qquad \frac{}{N, GL}$$

$$VM_{Failure} \qquad \frac{(\{V_m\}, \{V_{GH}\} \vee \{V_{m'}\}\} \sqcup N, (\{V_{GH}\}, Members) \cup GL}{N, (\{V_{GH}\}, Members') \cup GL} \quad Where \begin{cases} V_{m'} \in Members \\ Members' = Members \setminus \{V_m\} \\ Failure(\{V_m\}) \equiv True \end{cases}$$

$$VGH_{Failure} \qquad \frac{(\{V_{GH}\}, Members) \sqcup N, (\{V_{GH}\}, Members) \cup GL}{N, (\{V_m\}, Members') \cup GL} \quad Where \begin{cases} Members' = Members \setminus \{V_m\} \\ Failure(\{V_{GH}\}) \equiv True \\ S(\{V_m\}) > Max_{\{V_x\} \in Members}(S(\{V_x\})) \end{cases}$$

$$V_{Arrival} \qquad \frac{(\{V_n\}, \{V_{GH}\} \vee Members\} \cup N, (\{V_{GH}\}, Members) \cup GL}{N, (\{V_{GH}\}, V_n \cup Members) \cup GL}$$

$$stop \qquad \frac{\emptyset, GL}{stop}$$

Fig. 2 VGA's inference system

Table 4 Used notations

Notation	Meaning
N	A set of couples (a, b) where a is a new, moving or failing vehicle and b, a vehicle detecting the failure or the arrival of vehicle a.
GL	The "initial" set of groups generated after the VGA setting up phase
Members	The set of members belonging to a given group of GL
V_{GH}	A GH vehicle
V_m	A member vehicle belonging to the set of members *Members*
$S(\{V_n\})$	The reputation score of a vehicle n belonging to N
$Failure(\{V_n\})$	The status of a vehicle n i.e. true if it fails otherwise false
$\vert -^*$	The reflexive application of the proposed inference rules

displacement of existing members or GHs. The inference system stops when all vehicles (new, moving or failing vehicle) are dealt with.

Each of the proposed inference rules is detailed below.

1. $VM_{Failure}$ inference rule. $VM_{Failure}$ is triggered when a GH or a member vehicle detects the failure of a member ($Failure(\{V_m\} \equiv True)$). In this case, the $VM_{Failure}$ inference rule is applied to remove the member vehicle V_m for the members set *Members* in an existing group of GL.
2. $VGH_{Failure}$ inference rule. The $VGH_{Failure}$ inference rule is applied when a member vehicle detects the failure of its $GH(Failure(\{V_{GH}\} \equiv True))$. In this case, the $VGH_{Failure}$ inference rule is triggered in order to elect another vehicle member V_m, having the highest reputation score, to take over the old GH's role.
3. $V_{Arrival}$ inference rule. $V_{arrival}$ is triggered when a vehicle V_i (a GH or a member) detects a new vehicle V_j by receiving a Beacon message (V_j is detached from its group or has joined the VANET for the first time). In this case V_i verifies the direction and the closeness of V_j. If these checks are successful, V_i computes its reputation score and sends a M_{ack} message to V_j in order to integrate it into its group as member.

Validation

In this section, verification of the soundness and completeness of the proposed inference system is achieved. Soundness is proved by showing that groups remain safe even after a VANET's topology changes (due to the arrival of a new vehicle, or the displacement or failure of an existing vehicle). Completeness is proved by checking that all expected potential scenarios are handled by the proposed inference system. The groups' safety proof is built upon three formal properties: (1) independence: each vehicle belongs to only one group; (2) single role: each vehicle has a unique role i.e. GH or member; and (3) stability: each group has a unique GH.

Soundness Verification

In this section, we prove that the proposed inference system is sound by showing that groups remain safe even after a VANET topology change. Hence, three properties have to be considered: independence, single role and stability.

In the following, these properties are first defined and then proved using appropriate theorems.

Proposition 1 (Independence) *"Two groups G_i and G_j are independent iff $G_i \bigcap G_j = \emptyset$".*

Theorem 1 (Groups Independence) *Initially, all groups in a VANET are independent (by assumption). If $(N, GL) \mid -^{*}$ stop then the independence property is preserved.*

Proof *If $(N, GL) \mid -^{*}$ stop* then only one inference rule among $VM_{Failure}$, $VGH_{Failure}$ or $V_{Arrival}$ can be applied for each element in *N*. Hence, we must verify whether the application of each inference rule locally maintains this property.

- When a new vehicle V_n arrives in the VANET network, only the $V_{Arrival}$ inference rule is triggered in order to integrate V_n in a GL's group as a member node. Therefore, groups remain separate and the independence property is preserved.
- When the set *N* includes failing members, the $VM_{Failure}$ inference rule is applied to remove the failing V_n member from the Members set.
- Otherwise (if a *GH* vehicle fails), the $VGH_{Failure}$ inference rule is applied to remove the *GH* and to elect another member as *GH*.

□

In these three cases, modifications occur in a single group without altering the others. Hence, the independence property is preserved.

Proposition 2 (Single vehicle role) *"A vehicle has a unique role (GH or member): Given a group $GL(\{GH\}, Members)$, $\{GH\} \bigcap Members = \emptyset$".*

Theorem 2 (Vehicles' single role) *Assuming that initially, all vehicles in the VANET have a single role, if $(N, GL) \mid -^{*}$ stop then the single role property is preserved.*

Proof After the VGA setup phase, vehicles have a single role (GH or member). Hence, we have to check whether the application of each rule of the proposed inference system locally maintains this property. If $(N, GL) \mid -^{*}$ *stop* then only one inference rule of $VM_{Failure}$, $VGH_{Failure}$ or $V_{Arrival}$ applies for each element in N.

- When a new vehicle V_n arrives in the VANET, $V_{Arrival}$ the inference rule is applied by including V_n in the Members' set in GL's group. Therefore, $\{GH\}$ and *Members* remain disjoint.
- For a failing member vehicle V_n, only the $VM_{Failure}$ inference rule is triggered by removing V_n from the members set Members in a GL's group and all the other roles are maintained. Therefore, $\{GH\}$ and *Members* remain disjoint.
- For the case of GH failure, only the $VGH_{Failure}$ inference rule is applied by removing the failing *GH* and by electing another member vehicle from the Members set as the new GH. Its role as a member disappears. Hence, the single role property is preserved.

□

Proposition 3 *(Stability). "A group GL is stable if it has a unique GH".*

Theorem 3 (Groups' single GH) *Assuming that initially, all vehicles in a VANET have a single role, if* $(N, GL) \mid -^{*}$ *stop then the stability property is preserved.*

Proof By assuming that, after the VGA setup phase, groups are stable, we must check whether the application of each inference rule locally maintains this property. If $(N, GL) \mid -^{*}$ *stop* then only one inference rule among $VM_{Failure}$, $VGH_{Failure}$ or $V_{Arrival}$ applies for each element in N.

- When a new vehicle V_n arrives in the network, the $V_{Arrival}$ inference rule is applied to integrate V_n into the detected GL's group as a member. Hence, the unique $\{GH\}$ vehicle in *GL* is preserved.
- For a failing member V_m in a GL group, only the $VM_{Failure}$ inference rule is applied to remove it from its group in the Members' set. In this case, the stability property is preserved because the $\{GH\}$ vehicle in *GL* remains unique.
- $VGH_{Failure}$ is applied when a GH fails, by removing it, and electing another member V_m as the GH. The *GH* vehicle remains unique.

□

Corollary 1 (Safety) "*A group is safe if it is independent from any other groups, all its vehicles have a unique role, and it is stable".*

Proposition 4 (Soundness) *Assuming that initially, the VANET is safe, if* $(N, GL) \mid -^{*}$ *stop then the safety property is preserved.*

Proof Using Theorems 1, 2 and 3, if $(N, GL) \mid -^{*}$ *stop*, the independence, single role and stability properties are preserved. Hence, the VANET remains safe. □

Completeness Verification

Once the soundness of the proposed inference system has been established, we can proceed to the verification of its completeness. This is achieved by determining whether all potential situations are handled by the inference system.

Theorem 4 (Completeness) *If the VANET remains safe after the arrival, displacement or failure of vehicles then* $(N, GL) \mid -^{*}$ *stop.*

Proof Assume that a VANET remains safe after the arrival, displacement or failure of a set k of vehicles. The safety property implies that all groups are independent, include single node roles, and are stable. □

Two situations can be distinguished:

- When a vehicle V_n arrives or an existing vehicle moves, it integrates an existing group GL_n as a member node.
- When a vehicle V_n fails, its treatment depends on its role: if V_n is a GH, $VGH_{Failure}$ is applied; otherwise, $VM_{Failure}$ handles failing members.

In both cases, V_n is removed from the VANET. It follows that $(N, GL) \mid - (N1, GL_1) \mid - ...(\emptyset, GL_k) \mid -^{*}$ *stop.*

Conclusion

VANETs offering interesting opportunities in traffic safety and road network efficiency while raising several technical issues such as privacy, and the ability to prevent malicious agents from interfering with network operations (e.g. modification of exchanged data, or fraudulent generation of data). In this paper, we proposed TSME, a trust-based security scheme for VANETs to counter such uncertainty. Firstly, we proposed a new grouping algorithm named VGA, associating vehicles into groups and selecting a Group-Head to mediate between the group and the rest of the network. Our grouping algorithm was designed to be highly dynamic and scalable since it can cope with the main situations that a vehicle may face in such a network, including vehicle arrival and departure. Secondly, we introduced a trust management process handling the message exchange into the VANET and built upon the grouping algorithm. Our proposal measured the veracity of a given alert message using a score calculation based on location closeness, time closeness and freshness of the message, as well as the sender's reputation. Reputations were computed and updated based on the closeness of the witness vehicle sending the message to the event, the delay between the event and its associated message, and the number of forwarders. Thirdly, we formally validated the whole proposal, TSME, using an inference system to establish its soundness and its completeness.

In future work, now we have shown the completeness and soundness of TSME, we intend to deal with some real case studies using simple numerical examples as well as large performance simulations to benchmark its efficiency.

Acknowledgements
None

Authors' contributions
RA: main contributor of this article, ABCD: main contributor of this article, DS: writing, proofreading, has achieved lot of revisions related to scientific content and its organization to finalize the manuscript. All authors read and approved the final manuscript.

Funding
This work was partially supported by the ex-Région Limousin, under a grant for the project "IoTSec"; by the MIRES research federation under grants for projects "SPOCK2" and "CANIoT"; by the Région Nouvelle-Aquitaine under the grant for project "SVP-IoT"; and by the ID-Fix project, an ANR-funded project (ANR-16-CE39-0004).

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Digital Security Research Lab, Higher School of Communication of Tunis, SUP'Com, University of Carthage, Tunis, Tunisia.
[2] MathIS, XLIM (UMR CNRS 7252/Université de Limoges), Limoges, France.

Received: 24 April 2020 Accepted: 21 September 2020
Published online: 17 October 2020

References

1. Raw RS, Kumar M, Singh N (2013) Security challenges, issues and their solutions for vanet. Int J Netw Secur Appl 5(5):95–105
2. Mohammadi V, Rahmani AM, Darwesh AM, Sahafi A (2019) Trust-based recommendation systems in internet of things: a systematic literature review. Hum Centric Comput Inf Sci 9(1):21
3. Sharma PK, Moon SY, Park JH (2017) Block-vn: a distributed blockchain based vehicular network architecture in smart city. J Inf Process Syst 13(1):184–195
4. Abassi R, El Fatmi S, Guemara A (2015) Countering the collusion attack in a trust-based manet. In Konstantinos Lambrinoudakis, Vincenzo Morabito, and Marinos Themistocleous, editors. In: Proceedings of the European, Mediterranean & Middle Eastern Conference on Information Systems. p. 575–585.
5. Chehida AAB, Abassi R, El Fatmi R, Guemara S (2013) A reputation-based clustering mechanism for manet routing security. In: 2013 international conference on availability, reliability and security. IEEE. p 310–315.
6. Kamvar SD, Schlosser MT, Garcia-Molina H (2003) The eigentrust algorithm for reputation management in p2p networks. In: Proceedings of the 12th international conference on World Wide Web. Association for Computing Machinery. p 640–651
7. Awasthi SK, Singh Y (2020) Absolutetrust: algorithm for aggregation of trust in peer-to-peer networks. IEEE transactions on dependable and secure computing.
8. Junqi D, Dong Y, Haoqing Z, Sidong Z, Jing Z (2014) Tsrf: a trust-aware secure routing framework in wireless sensor networks. Int J Distrib Sensor Netw 10:1
9. Lupia A, De Rango F (2014) Performance evaluation of secure aodv with trust management under an energy aware perspective. In: International symposium on performance evaluation of computer and telecommunication systems (SPECTS 2014). IEEE. p. 599–606
10. Naveena S, Senthilkumar C, Manikandan T (2020) Analysis and countermeasures of black-hole attack in manet by employing trust-based routing. In: 2020 6th international conference on advanced computing and communication systems (ICACCS). p. 1222–1227
11. Li Y, Hongyun X, Cao Q, Li Z, Shen S (2015) Evolutionary game-based trust strategy adjustment among nodes in wireless sensor networks. Int J Distrib Sensor Netw 11:2
12. Saidi A, benahmed PK (2020) Secure cluster head election algorithm and misbehavior detection approach based on trust management technique for clustered wireless sensor networks. Ad Hoc Networks. p. 102215.
13. Cheng JJ, Yuan GY, Zhou MC, GaoSC, Huang ZH, Liu C (2020) A connectivity prediction-based dynamic clustering model for vanet in an urban scene. In: IEEE Internet of Things Journal.
14. Yong-hao W (2020) A trust management model for internet of vehicles. In: Proceedings of the 2020 4th international conference on cryptography, security and privacy, New York, NY, USA, 2020. Association for Computing Machinery. p. 136–140.
15. Ahmad F, Kurugollu F, Adnane A, Hussain R, Hussain F (2020) Marine: Man-in-the-middle attack resistant trust model in connected vehicles. IEEE Internet Things J 7(4):3310–3322
16. Chen YM, Wei YC (2013) A beacon-based trust management system for enhancing user centric location privacy in vanets. J Commun Netw 15(2):153–163

17. Al Falasi H, Mohamed N (2015) Similarity-based trust management system for detecting fake safety messages in vanets. In: International conference on internet of vehicles. Springer. p. 273–284.
18. Zhang J, Chen C, Cohen R (2013) Trust modeling for message relay control and local action decision making in vanets. Security Commun Netw 6(1):1–14
19. Chuang M-C, Lee J-F (2013) Team: trust-extended authentication mechanism for vehicular ad hoc networks. IEEE Syst J 8(3):749–758
20. Hao H, Rongxing L, Huang C, Zhang Z (2017) Ptrs: a privacy-preserving trust-based relay selection scheme in vanets. Peer-to-Peer Netw Appl 10(5):1204–1218
21. Terence JS, Purushothaman G (2019) A Novel Technique to Detect Malicious Packet Dropping Attacks in Wireless Sensor Networks. J Inf Process Syst 15(1):203-216. https://doi.org/10.3745/JIPS.03.0110
22. Das S, Das I, Singh RP, Johri P, Kumar A (2019) Trust-based scheme for location finding in vanets using trustworthiness of node. In: Data and communication networks. Springer. p. 43–55.
23. Mahmood A, Butler B, Zhang WE, Sheng QZ, Siddiqui SA (2019) A hybrid trust management heuristic for vanets. In: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) IEEE. p. 748–752.
24. Sugumar R, Rengarajan A, Jayakumar C (2018) Trust based authentication technique for cluster based vehicular ad hoc networks (vanet). Wirel Netw 24(2):373–382
25. Hasrouny H, Samhat AE, Bassil C, Laouiti A (2018) Trust model for group leader selection in vanet. Int J Digital Inf Wirel Commun 8(2):139–144
26. Xiao Y, Liu Y (2019) Bayestrust and vehiclerank: constructing an implicit web of trust in vanet. IEEE Transac Vehic Technol 68(3):2850–2864
27. Page L, Brin S, Motwani R, Winograd T (1999) The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, November
28. Cooper C, Franklin D, Ros M, Safaei F, Abolhasan M (2016) A comparative survey of vanet clustering techniques. IEEE Commun Surv Tutor 19(1):657–681
29. Haidar DA, Cuppens-Boulahia N, Cuppens F, Debar H (2009) Xena: an access negotiation framework using xacml. Ann Telecommun 64(1–2):155–169
30. Abassi R, Ben Chehida A, Guemara El Fatmi S (2016) A trust-based security environment in manet: definition and performance evaluation. Ann Telecommun 12:14
31. Douss Aida BC, Abassi R, Youssef NB, El Fatmi SG (2015) A formal environment for manet organization and security. In: International conference on cryptology and network security. Springer. p. 144–159.
32. Souri A, Rahmani AM, Navimipour NJ, Rezaei R (2019) A symbolic model checking approach in formal verification of distributed systems. Hum Cent Comput Inf Sci 9:4. https://doi.org/10.1186/s13673-019-0165-x