An accurate and efficient collaborative intrusion detection framework to secure vehicular networks

Hichem Sedjelmaci, Sidi Mohammed Senouci

An accurate and efficient collaborative intrusion detection framework to secure vehicular networks

Hichem Sedjelmaci *, Sidi Mohammed Senouci

University of Burgundy, DRIVE Lab, 49 Rue Mademoiselle Bourgeois, 58000 Nevers, France

The advancement of wireless communication leads researchers to develop and conceive the idea of vehicular networks, also known as vehicular ad hoc networks (VANETs). Security in such network is mandatory due to a vital information that are managed by the vehicle. Therefore, in this paper we design and implement an accurate and lightweight intrusion detection framework, called AECFV, that aims to protect the network against the most dangerous attacks that could occur on such network. AECFV is suitable for VANET's characteristics such as high node's mobility and rapid topology change. This is achieved with a help of the proposed secured clustering algorithm that considers both node's mobility and network vulnerability during cluster formation. Clusters are constructed with a high stability and good connectivity. Cluster-Heads (CHs) are elected based on both node's mobility and the vehicle's trust-level. The simulation performed using NS-3 simulator shows, AECFV exhibits a high detection rate, low false positive rate, faster attack detection, and lower communication overhead compared to current detection frameworks.

1. Introduction

Vehicle ad hoc networks (VANETs) are attracting much attention from both academia and industry. They are considered as the main system for the deployment of Intelligent Transportation Systems (ITS) based applications. These networks rely on a various types of data collected and/or disseminated from/to vehicles to provide multiple services, which can be sorted in three classes: (i) Road traffic management application such as driver assistance, management of traffic signals, providing information about road and traffic conditions, and route planning. (ii) Traffic safety applications such as self-driving (or autonomous car), prevention and warning of accidents, and emergency management (e-Call for Emergency-Call). (iii) Mobility and comfort application such as point of interest services for vehicles, eco-driving services, management of vehicle fleets services and machine to machine (M2M) services. Securing these networks is an important challenge, especially when traffic-safety applications are deployed. In fact, with these applications, vehicles manage vital and sensitive information that are attractive for attackers. A security mechanism is mandatory to protect VANETs against attacks.

The intrusion detection systems (IDSs) have shown their efficiency to detect internal and external attacks with a high accuracy [1–5]. These systems use special agent nodes to monitor the behavior of a target node and trigger an alarm when a malicious behavior is detected. This paper describes the design and implementation of an accurate and lightweight

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Associate Editor Dr. Ayaz Isazadeh.

* Corresponding author.

E-mail addresses: Sid-Ahmed-Hichem.Sedjelmaci@u-bourgogne.fr (H. Sedjelmaci), Sidi-Mohammed.Senouci@u-bourgogne.fr (S.M. Senouci).

intrusion detection framework for vehicular networks (AECFV) that takes into account the VANET's characteristics such as high node's mobility and rapid topology change. To handle these characteristics an effective secured clustering[1] algorithm is proposed. This algorithm considers node's mobility during cluster formation, produces clusters with high stability, assures more connectivity between cluster members and elects Cluster-Heads (CHs) based on the vehicles' trust-level. Choosing a cluster-based topology lies in the fact that it is the most appropriate structure for large-scale networks since it allows reducing the broadcast storm and hence decrease the communication overhead [2,6,7].

AECFV aims to secure traffic-safety applications, where the focus is to detect and prevent dangerous attacks that could occur in this application, such as: selective forwarding, black hole, packet duplication, resource exhaustion, wormhole and Sybil attacks. It uses three intrusion detection agents: Local Intrusion Detection System (LIDS) running at cluster member level, Global Intrusion Detection System (GIDS) running at CH level and Global Decision System (GDS) running at Road Side Unit (RSU) level. To detect the malicious vehicle with a high accuracy (i.e. high detection and low false positive rates), the LIDS uses a rules-based detection and GIDS uses a hybrid detection technique (i.e. rules-based detection and anomaly detection based on Support Vector Machine – SVM). The rules based detection relies on a certain rules related to each attack to model a normal behavior and anomaly detection is based on a learning algorithm to model a normal behavior. The combination of these both techniques (i.e. hybrid detection) allow a high detection and low false positive rates [1]. Furthermore, with a help of the proposed detection techniques, a new reputation mechanism is developed that evaluates the trustworthiness level of vehicles according to their behaviors and the information they provide. AECFV suits the following requirements: fast in terms of attacks detection, lightweight in terms of communication overhead, and scalable.

The rest of the paper is organized as follows: In Section 2, we underline previous related work and describe their main shortcomings. In Section 3, we describe our secured vehicular clustering algorithm. Section 4 presents details about our intrusion detection framework AECFV and Section 5 provides NS3 simulation results. In Section 6, we analyze various security aspects of the AECFV. Finally, we conclude the paper and give some perspectives that we envisage to carry out in Section 7.

2. Related work

The intrusion detection system is the most reliable technique to protect vehicular networks against the malicious nodes since it has the ability to detect internal and external attacks with a high accuracy (i.e. high detection and low false positive rates), unlike cryptography mechanisms that prevent only from external attackers to penetrate the network [1,8]. Furthermore, the proposed detection system should take into account the node's high- mobility and frequent network topology change.

Recently, some intrusion detection frameworks have been proposed to address security issues in vehicular networks [2,3,9–13]. In [9], the authors aim to identify the vehicle that provides a false location by applying a set of detection rules. In this scheme, a cooperative detection is applied between intrusion detection agents to identify the malicious vehicle with a high accuracy. To check the claimed position of a monitored vehicle, a packet's Time-of-Flight (ToF) technique proposed by the authors in [10] is used. ToF is defined as the time taken by the packet to arrive at the destination and return back. In their simulations results, their scheme exhibits a high detection rate, high delivery ratio and low packet loss when the number of malicious nodes is large (60 intruders). However, the authors did not take into account collusion issues that can occur in such wireless networks. In fact, when collusions occurred, ToF value computed by the intrusion detection agents will be incorrect. In [11], the authors propose a detection framework to identify the malicious vehicle that provides a false position coordinates. The detection policy proposed by the authors rely on comparing the claimed position of a monitored node and the expected position computed by this IDS, which is based on plausibility model [14]. This latter relies on the vehicle position and movement verification. In this research work, the authors aim to detect two attacks, which are fake congestion and denial of congestion. According to the simulation results, their framework exhibits a high detection rate. In [3], an intrusion detection schema against selective forwarding and false information dissemination attacks is proposed. The detection policy used by this schema relies on *anomaly detection* technique based on a entropy method, which aims to model a normal behavior of a monitored vehicle and any deviation from this model is detected as an attack. According to their simulation results, these attacks were detected with a high accuracy. Furthermore, when the number of attackers increases the performance of the detection frameworks [11,3] degrade significantly (i.e. high false positive rate and low detection rates). In [12], the authors propose a detection framework for VANET called T-CLAIDS that uses an *anomaly based detection technique* to identify the malicious vehicle. This technique uses a learning automat and Markov Chain Model (MCM) approaches to model a normal behavior of node. Combining between these two approaches, the authors prove in the simulation that their approach allows detecting the attacks with a high accuracy. Nevertheless, embedding these both algorithms in a vehicular network could generate a high computation and communication overheads, specifically when the number of vehicles increase. Furthermore, the authors did not define the kind of attacks that were detected.

Recently in [2,7,13], the authors develop a secure cluster-based vehicular network (i.e. the most trusted node at each cluster is elected as a Cluster-Head, CH). Specifically, in [2] the authors propose a detection framework called VWCA to secure a

[1] Clustering architecture aims to group vehicles into a set of clusters, where cluster members communicate with a special node called Cluster-Head (CH).

cluster-based vehicular network. The cluster's formation and cluster-head's election are based on vehicle's trust-level, transmission range, and direction. When clusters are formed, each vehicle monitors it neighbors and assign a trust-level to each vehicle they monitor. When a suspected node is detected the monitoring vehicle forwards the identity of this node to it CH (elected as a trusted node) to take a final decision, i.e. whether this suspected node is malicious or not. *The rules based detection technique* is used to model a normal behavior of vehicle and detect the malicious vehicle. It is noted that this technique use a set of rules related to each attack that they attempt to detect. According to the simulation results, this schema exhibits a high detection rate even when the number of vehicles increases. However, this schema exhibits a high false positive rate specifically when the number of vehicles increases as proved in [13]. In our recent work [13], an efficient intrusion detection framework called IDFV is embedded in a cluster-based topology to detect and eject the attacks like selective forwarding and wormhole attacks. IDFV applies a *hybrid detection technique* i.e. *rules based* and *anomaly detection* techniques, to model a normal behavior of vehicles and detect the malicious nodes. According to simulation results, this framework exhibits a high detection and low false positive rates even when the number of malicious vehicles is high. However, its drawback lies in the fact that it is heavy in terms of communication and computation overheads specifically when the number of vehicles increases.

In this paper, we develop an accurate and lightweight intrusion detection framework for the cluster-based vehicular network that handles the weaknesses of the intrusion detection schemas proposed in the current literature. The proposed approach applies a certain number of detection agents that run at three levels i.e. cluster member, cluster-head and RSU to detect with a high accuracy and short time the selective forwarding, black hole, wormhole, packet duplication, resource exhaustion and Sybil attacks.

3. Secured clustering algorithm

The purpose of a cluster-based algorithm is to group nodes into a set of clusters and assign a Cluster-Head (CH) for each cluster that has the ability to manage the information forwarded by its cluster members. It creates a virtual structure that simplifies the network management task and eases the deployment of services. This architecture takes advantage of node properties to issue this global structure that is sufficiently autonomous and dynamic to deal with any local change. Such algorithm is very interesting in highly-dynamic networks like VANETs [2,7]. Therefore, a new one-hop clustering algorithm is proposed inspired by the authors in [2,7] that aims to address three main issues: (i) Stability, since a high frequency of CH election will increase the overhead in the network. Thereby, reducing the CH election process is important to keep the network stability [2], (ii) Connectivity, it assures that a vehicle is reachable from any other neighbor's nodes and (iii) Security, it is important to elect at each cluster the most trusted vehicle as a CH. Hence, cluster's formation and CH election are based on two main parameters to maintain the cluster stability, connectivity and assuring security:

3.1. Normal distribution of vehicle's velocities

The nodes within a same cluster are grouped according to their velocities distributions, i.e. the nodes with velocities following a normal distribution are grouped in the same cluster. Each vehicle in the network broadcasts periodically a *cluster_formation* message, which includes its identifier id, *position*, velocity, and the set of identifiers id_s of one-hop neighbors with their *Trust Values (TVs)*. Upon the receipt of the *cluster_formation* message, all vehicles that have their velocities following a normal distribution are grouped into the same cluster. It has to be noted that in a normal distribution concept, the mean and standard deviation of data are computed. Velocities of the vehicles follow a normal distribution if the velocities lie within three standard deviations of the mean [15].

3.2. Social behavior

To assure a more connectivity within a cluster and elect the more stable and trusted node as a CH, a *social behavior* parameter is introduced. This parameter aims to elect a CH based on both *vehicle's Trust Value (TV)* and *Boundary Distance (BD)*. Concerning the TV, before the cluster formation, all vehicles have the ability to play an IDS role i.e. each vehicle monitors its neighbors nodes. This monitoring uses a detection policy-based on rules (explained in Section 4.1). In case when a monitored neighbor exhibits an attack, its TV is set to 0 otherwise it is equal to 1. It is important to mention that, at the beginning all vehicles have the ability to monitor their neighbors node and compute their trust values. However, when a vehicle exhibits an attack (i.e. it trust value equal to 0) it has not the ability to play an IDS agent role and hence all the messages that are broadcasted by this malicious vehicle is ignored by the other vehicles. Furthermore, after cluster's formation, the TV is computed based on reputation protocol, see Section 4.2.2. As mentioned above, the trust values of monitored vehicles with their id_s are broadcasted periodically within a network using the *cluster_formation* message. Concerning the vehicle's boundary distance BD, to maintain more connectivity within a cluster, the leaf node must not be close to the radio range's boundary of the father node as it will likely leave the cluster (when father node is elected as CH). Therefore, we fix a boundary distance that a vehicle must respect to prolong its connectivity, which is lower than $\frac{R}{2}$, where R is the maximum radio range of vehicle as shown in Fig. 1. It is noted that, in our experiments we carry out several simulations by varying the boundary distance from R, $R/2$ and $R/3$. However according to our simulation result we found that $R/2$ is more

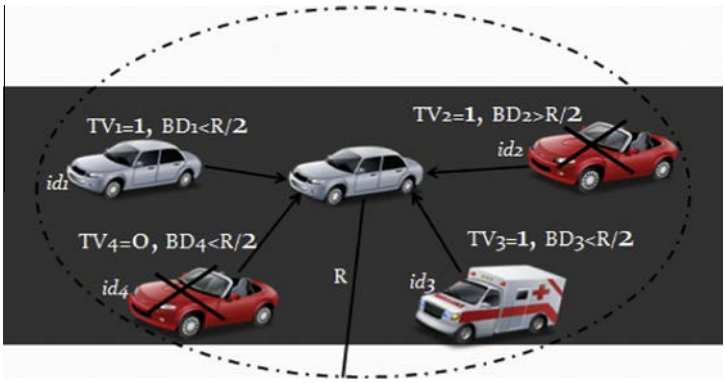

Fig. 1. The building of Social Behavior List (SBL).

Table 1
Social Behavior List (SBL).

id	TV	BD
id_1	$TV_1 = 1$	BD1 < $R/2$
id_3	$TV_3 = 1$	BD3 < $R/2$

appropriate boundary distance that allow us to obtain a more stable cluster (i.e. reducing the CH election) and prolong the network connectivity between the cluster member and it cluster head.

Each vehicle creates a *Social Behavior List* (*SBL*) for CH election purpose based on BDs and TVs of the monitored nodes. When monitored node's TV is equal to 0 or/and its BD $\geqslant R/2$, the node is removed from the SBL as illustrated in Fig. 1 and shown in Table 1.

Finally, after obtaining enough information about the network, each vehicle v_i computes the number of vehicles stored in the SBL (NS_{v_i}) and broadcasts *SBL_status* message. This latter contains the vehicle v_i's id and the NS_{v_i}'s value. Upon the receipt of the *SBL_status* message, each vehicle computes a utility function related to each vehicle v_i as shown in Eq. (1).

$$U_{v_i} = \alpha_1 \cdot TV_{v_i} + \alpha_2 \cdot NS_{v_i} \tag{1}$$

where $\alpha_1, \alpha_2 \in [0, 1]$ and TV_{v_i} is a trust-level of vehicle v_i.

Each vehicle broadcasts a *CH_election* message within its cluster. This message includes the utility function's values related to each neighbor. Afterward, the node with the high utility function is elected as cluster-head. CH is an attractive target for attackers due to the relevant information it manages. However, when the CH exhibits a malicious behavior (detected by AECFV), the election process of new CH is launched as explained above. In this case, the computation of trust value is determined according to the reputation mechanism (see Section 4.2.2).

In this paper the aim is to secure the VANET based-safety applications. In this service when a crash occurs, the nearby cluster members broadcast an alert message such as *Post Crash Notification* [4] to their CH. Afterward, this CH aggregates these information, inform the incoming vehicles to take some evasive actions and forwards the aggregated information to the neighboring CH. This process continues until the RSU receives this information. In this study, we assume that the RSU is powerful and cannot go down. However, when a CH does not function correctly, another cluster head is elected. Furthermore, we assume also that the network work well since when there is an anomaly in the network, the message alert that are disseminated by the nearby vehicles of crash areas does not arrive at the incoming node and hence a traffic jam could occur in the network.

4. An accurate and lightwight intrusion detection framework for vehicular networks (AECFV) description

In this paper a new intrusion detection framework AECFV is proposed, that takes into account the specific characteristics of vehicular networks thanks to the secured clustering algorithm. This algorithm considers both node's mobility and network vulnerability during clusters formation. The proposed framework applies a set of techniques to detect and prevent the occurrences of the most dangerous attacks that can appear in a vehicular network. Intrusion detection systems use special agent nodes to monitor the behavior of a target node and trigger an alarm when a malicious behavior is detected. As illustrated in Figs. 2 and 3, AECFV is equipped with two main detection systems and a decision system: Local Intrusion Detection System (LIDS) running at cluster member level that monitors the behaviors of it neighboring vehicles and the cluster-head, Global Intrusion Detection System (GIDS) running at CH level that monitors the behaviors of it cluster members and evaluates the trustworthiness of monitored vehicles, and lastly Global Decision System (GDS) running at RSU level that computes the

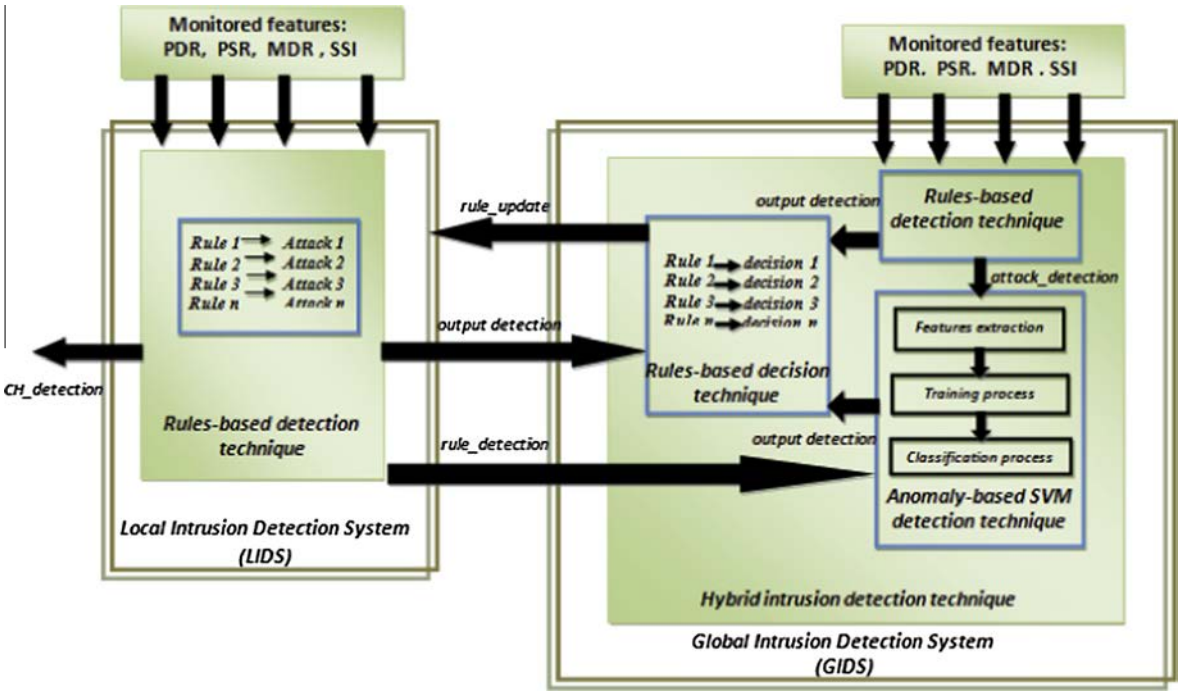

Fig. 2. Intrusion detection process between LIDS and GIDS.

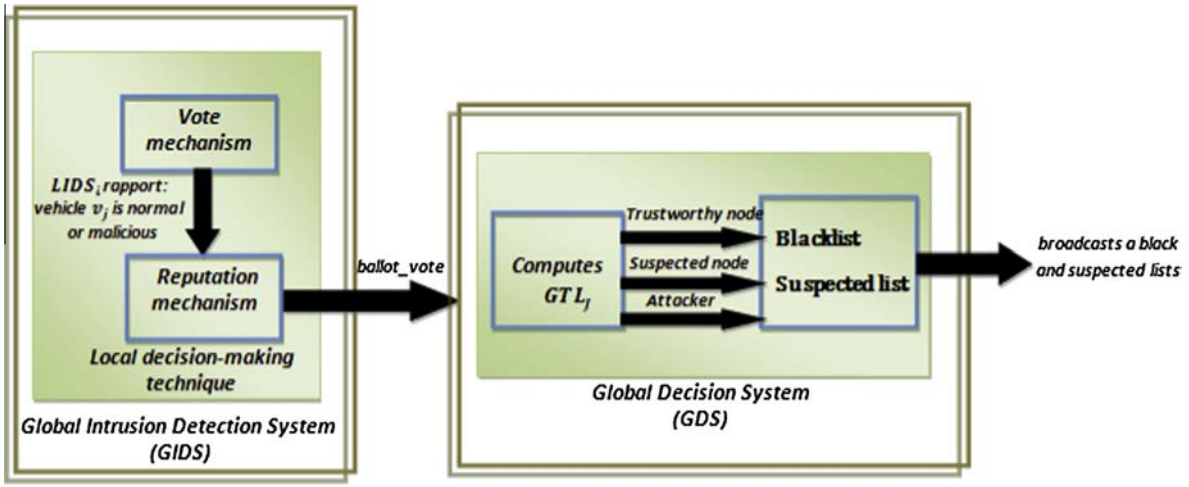

Fig. 3. Intrusion detection and decision process between GIDS and GDS.

Trust-level (TL) related to each vehicle and categorize them into an appropriate list according to their TL. These systems correspond to a network IDS (NIDS) since they monitor the behaviors of nodes located within their radio range. It is noted that, before the cluster formation, all nodes have the ability to play the LIDS role. However, after cluster formation and in order to decrease the overhead, only an optimal number of cluster member nodes activate their LIDSs. Furthermore, to assure the communication's privacy and ensure source authentication, an Elliptic Curve Cryptography (ECC) provided by the current VANET standard IEEE 1609.2 [16] is used.

In the rest of this section, the main techniques of these systems, the process of attack detection and the reputation mechanism are described.

4.1. Local Intrusion Detection System (LIDS)

This system runs at cluster member level so that they can monitor their CH's behavior. This is vital as CHs are attractive targets for attackers due to the relevant information they manage. Furthermore, each LIDS monitors its neighbors located

within its radio range. However, to decrease the resulting overhead, only an optimal number of cluster members can activate their LIDSs function. In fact, when a high number of nodes activate their LIDS function, the overhead highly increases and as a consequence the network performance is degraded. Therefore, to achieve a high level of security (i.e. high detection and low false positive rates) and generates a low overhead, we use our IDS activation strategy [17]. This later is based on a game theory concept to predict the future behavior of a malicious vehicle. In work [17], we model a game between the RSU and vehicles located within it radio range as *Bayesian game*. Furthermore, when the RSU reaches an optimal solution defined as *Nash equilibrium* it activates its monitoring process to detect and categorize the monitored node. It is noted that, the RSU activate its monitoring process before the attacker launches his attack. In this paper, we are inspired from this game method and model a game between each cluster member and vehicles located within its radio range, and as in [17] when cluster member reaches a Nash *equilibrium* it activates its monitoring process (i.e. LIDS).

The LIDS monitors the neighboring vehicles including the CH by storing their id_s and computes the following features related to each one of them: Packets Drop Ratio (PDR), Packets Sent Ratio (PSR), Message Duplication Ratio (MDR) and Signal Strength Intensity (SSI). After that, it applies the rules-based detection technique, to detect any malicious behavior that could occur. This technique has the ability to use a set of rules, detailed in the following paragraph, to model a normal behavior of a vehicle and detect an attack when it occurs:

4.1.1. Detection of selective forwarding and black hole attacks

Such attacks could occur at CH level. As illustrated in Fig. 4(a), when the CH performs a selective forwarding, it stops the transmission of certain aggregated packets and starts dropping them. However, the black hole drops all the received packets form the cluster members (refer Fig. 4(a)). In order to identify such attacks, the CH's Packet Drop Rate (PDR) is computed. In this case, the LIDS monitors the number of packets sent from cluster members to the CH and CH's PDR. Therefore, when the CH drops all received packets or the CH's PDR is above a certain threshold TH_{sf} it will be suspected as a node that carries out a black hole or selective forwarding attacks, respectively. The threshold TH_{sf} is updated with a help of learning algorithm as described in Section 4.2. The detection rules corresponding to the selective forwarding and black hole attacks are illustrated in Fig. 5(a).

4.1.2. Detection of packet duplication and resource exhaustion attack

In such attacks, the attacker sends a considerable number of unwanted packets in order to waste bandwidth [18]. The attacker could be one of the cluster members as illustrated in Fig. 4(b). When a malicious vehicle carries out a packet duplication or resource exhaustion attack, its Message Duplication Ratio (MDR) or Packets Send Rate (PSR), respectively will be higher comported to legitimate neighbors. Therefore, the LIDS (and/or GIDS) monitors the number of packets that their

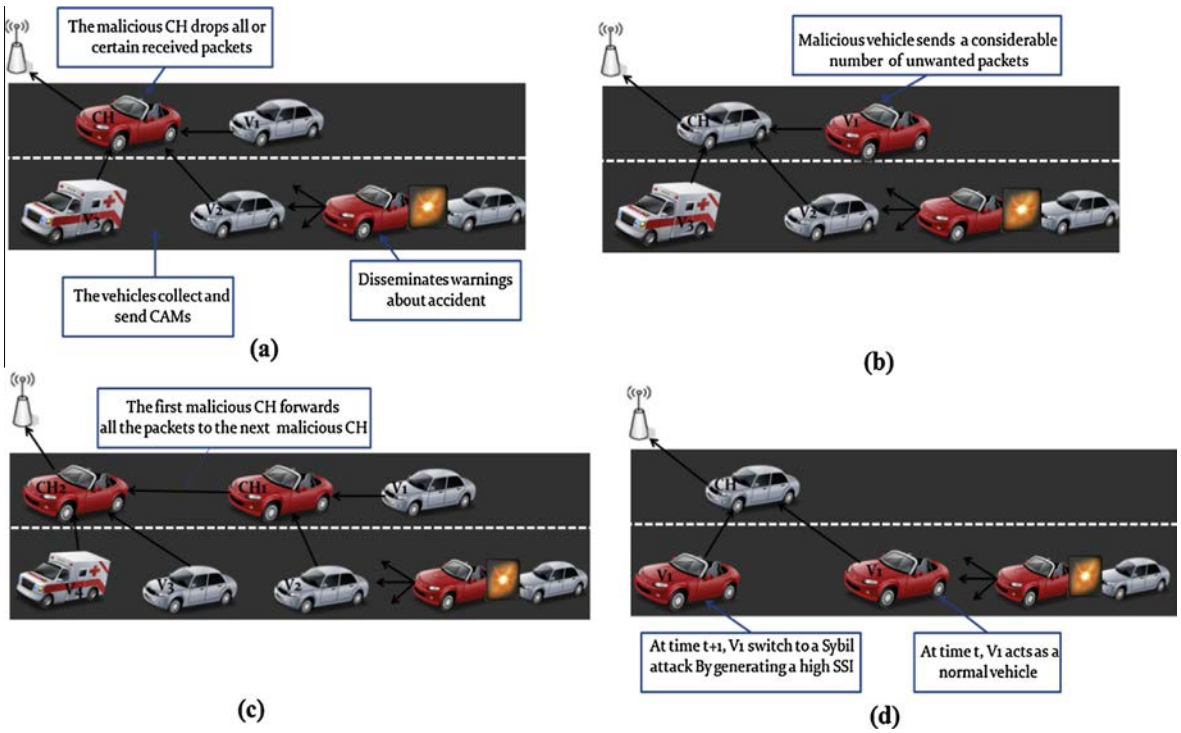

Fig. 4. Attacks' scenario: (a) Selective forwarding and black hole, (b) packet duplication and resource exhaustion, (c) wormhole and (d) Sybil attacks.

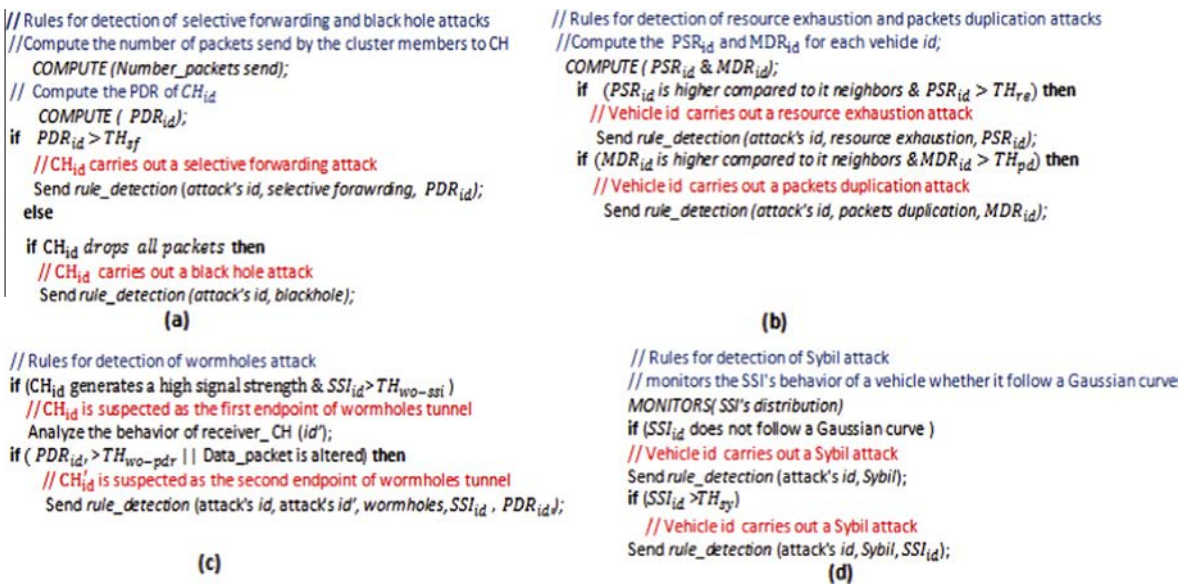

Fig. 5. Attacks detection rules. (At the beginning of our monitoring process, we fix the thresholds values: TH_{SF}, TH_{RE}, TH_{PD}, TH_{wo-ssi}, TH_{wo-pdr} and TH_{sy} as shown in Table 1. Afterward, these thresholds vary over time with the help of the training algorithm as described in Section 4.2).

neighbors send. In case when a monitored vehicle generates a high number of packets compared to its neighbors, it will be suspected as a node that carries out packet duplication or resource exhaustion attacks. For instance, vehicles situated in a crash area will send an alert message during a predefined period. The malicious vehicle will continue to send unwanted packets to its CH in order to waste its resources. As result, we define two thresholds TH_{pd} and TH_{re} related to packet duplication and resource exhaustion attacks, respectively. The thresholds TH_{pd} and TH_{re} are updated with a help of learning algorithm as described in Section 4.2. Rules related to the detection of these attacks are described in Fig. 5(b).

4.1.3. Detection of wormhole attack

In this attack, two malicious vehicles in a network can cooperate and transfer packets from a private tunnel that they have built. In addition, the first malicious node generates high Signal Strength Intensity (SSI) to convince a legitimate node that is close to destination (or CH) [13,19]. Hence, all received packets will be forwarded only to the next malicious node, which will in turn drops or modify certain packets and forward them to legitimate nodes (or RSU), as illustrated in Fig. 4(c). Such attack could occur between two malicious CHs. As a result, to identify such attack, the SSI and PDR (or altered data) of the targeted nodes are monitored, where SSI and PDR of the malicious nodes will be higher compared to their neighbors. We define two thresholds $TH_{wo-rssi}$ and TH_{wo-pdr} related to PDR and SSI that vehicles should respect. These thresholds are updated with a help of learning algorithm as described in Section 4.2. The rule of wormhole attack's detection is illustrated in Fig. 5(c).

4.1.4. Detection of Sybil attack

Such attack aims to create more than one identity on a single physical device [20], in order to weaken the detection process when it launches other attacks such as black-hole. When the Sybil attacker creates new identity, the SSI of that identity will be high enough to be distinguished from the neighboring nodes [20] as shown in Fig. 4(d).

The LIDS (or GIDS) collects SSIs from its neighbors and verifies whether distribution of SSIs follow a normal distribution (i.e. *Gaussian curve*). It should be noted that, in a normal distribution concept, data (e.g. SSIs) follow a normal distribution over time if the SSIs values are within (Mean $-3 * SD$) and (Mean $+3 * SD$) [15,21], where SD is a standard deviation (see Eq. (2)).

$$\begin{aligned} \text{Mean(SSI)} &= \sum_{i=1}^{n} \frac{\text{SSI}_i}{n} \\ \text{SD}(\text{SSI}_i) &= \text{SSI}_i - \text{Mean(SSI)} \end{aligned} \tag{2}$$

$i = \{1,\ldots,n\}$, where n is number of neighboring vehicles of LIDS (or GIDS).

In case when SSI_i of a monitored vehicle does not fall within *Gaussian curve* and its value is above a maximum threshold TH_{sy}, it will be detected as a node that carries out Sybil attack. The threshold TH_{sy} is updated with a help of learning algorithm as described in Section 4.2. Such attack could occur in CH or/and at the cluster member nodes. The Sybil attack's rule detection is illustrated in Fig. 5(d).

In case when a vehicle is suspected as an attacker, a *rule_detection* message is forwarded to GIDS (i.e. CH) as illustrated in Fig. 2 to confirm a malicious behavior of a suspected vehicle by launching a learning algorithm based on *Support Vector Machine (SVM)*. The SVM has the ability to model a normal and anomaly behaviors with a high accuracy (i.e. high detection and low false positive rates), and update all the thresholds cited above as explained in Section 4.2.

However, in case when a CH is suspected as an attacker, a cooperative detection between LIDSs is launched. In this detection, the LIDS broadcasts a *CH_detection* to all LIDSs within the same cluster as shown in Fig. 2. In case when more than half of LIDSs detect a CH as an attacker, it will be removed from the cluster and a new CH is elected (as explained in Section 3). Furthermore, the new CH collects *rule_detection* message from LIDSs and carry out a training and classification process to confirm the malicious behavior of the oldest CH and updates the thresholds as explained in Section 4.2.

4.2. Global Intrusion Detection System (GIDS)

This system is activated at each CH. It has the ability to monitor its cluster members and take a decision about the suspected vehicle detected by LIDS (normal or an attacker). The system is equipped with *hybrid intrusion detection* and *local decision-making techniques*, which are detailed as follows:

4.2.1. Hybrid intrusion detection technique

This technique combines between *rules-based detection* and *anomaly detection* techniques as shown in Fig. 2. The *anomaly detection* relies on learning algorithm based on SVM to model a normal behavior of a vehicle with a high accuracy. The GIDS monitors its cluster members by using a *rules-based detection* (as explained in Section 4.1). When a suspected vehicle is detected, it forwards an *attack_detection* message to *anomaly-based SVM detection* technique to confirm this malicious behavior as illustrated in Fig. 2. In addition, as mentioned above the anomaly based detection is used to confirm the attack detected by the LIDS. In the following, the training and classification processes of the SVM learning algorithm and the rules for *decision-making* are described. The choice of SVM lies on the fact that, it is more suitable for vehicular networks as it provides very good results with less training time compared to other learning algorithms such as neural networks [22].

The main components of the *anomaly-based SVM detection technique* (see Fig. 2) are, (1) *Features extraction:* this component receives the *rule_detection* and *attack_detection* messages from LIDS and GIDS, respectively. Such messages contain the type of detected attack, the LIDSs identifies id_s (for the *rule_detection* message) and the related attack features (PDR, PSR, MDR, SSI). For instance, in case when a wormhole attack is detected, the PDR and SSI of wormhole nodes represent the attack's features, which are applied as input vectors of the *classification process* for the attack confirmation purpose. (2) *Training process:* Each CH trains the SVM locally; then compute a set of vectors called *support vectors* that allow separation of data into two planes, normal and anomalous (binary classification). Ref. [22] describes more details about support vectors computation. In the first training process, the input vectors (PDR, PSR, MDR, SSI) are assumed to be collected from legitimate vehicles in order to simplify the modeling of the normal patterns. To determine with a high accuracy of a vehicle's behavior (normal or anomaly), the training process is carried out periodically since these features (i.e. input vectors) will vary over time due to several parameters such as collision of packets. (3) *Classification process*: This component classifies new incoming data (i.e. attack's features) delivered from feature *extraction* component according to the anomaly and the normal pattern, determined during the training process. Afterward the output of this process (the node is malicious or normal) is delivered to a *rules-based decision* as illustrated in Fig. 2.

4.2.1.1. Rules-based decision technique. It gathers the outputs from *rules-based detection* of LIDS or/and GIDS and *classification process* as illustrated in Fig. 2. Afterward, it involves the GIDS to take a decision by applying the following rules: (i) In case when a learning based detection confirms the attacks detected by LIDS, a vote mechanism is launched (see Section 4.2.2). (ii) Furthermore, when a learning based detection does not confirm the attack detected by LIDS, the GIDS launches a vote mechanism to verify the reliability of detection provided by LIDS, afterward it broadcasts a *rule_update* message to all LIDSs within a cluster to update their rules as illustrated in Fig. 2. In this case, the thresholds applied by a *rules-based detection* (i.e. TH_{sf}, TH_{pd}, TH_{re}, $TH_{wo-rssi}$, TH_{wo-pdr} and TH_{sy}) are replaced by the current features (i.e. input vector) provided by the SVM training algorithm. (iii) When the learning technique does not confirm the attack detected by a *rules-based detection* of GIDS, the thresholds mentioned above are updated as in the second rules.

4.2.2. Local decision-making technique

It has the ability to evaluate the trustworthiness of monitored vehicles and check the reliability of the decision provided by LIDSs. It computes the reputations of its cluster member and forwards this information to RSU for a *final decision-making.* This component relies on *vote and reputation-based mechanisms,* which are explained as follows: When a vehicle v_j is suspected as an attacker by LIDS, the *vote* mechanism checks the trustworthiness of this information by applying a majority vote process. The CH computes the number of LIDS agents that suspect this vehicle as malicious. Thereby, three scenarios are possible: (i) If more than half of LIDSs neighbors of a vehicle v_j identifies it as an attacker, the good reputation (Grep) and bad reputation (Brep) of $LIDS_j$ (that provide a correct detection) and a vehicle v_j are increased, respectively. These reputations are computed as follows: ($Grep_{i,j} = \sum_{i=1}^{n} \alpha_1 \cdot i + \beta_1$ and $Brep_{i,j} = \sum_{i=1}^{n} \alpha_2 \cdot i + \beta_2$. Here n is the number of LIDS agents that detect a vehicle v_j as an attacker, α and $\beta \in [0,1]$. (ii) When less than half of LIDS agents claim a malicious behavior of a vehicle v_j, the

bad reputations of $LIDS_j$ (that provide a false detection) are increased. This reputation is computed as follows: $Brep_{i,j} = \sum_{i=1}^{m} \alpha_2 \cdot i + \beta_2$. Here m is the number of false detections provided by the $LIDS_j$. (iii) During its passage through the cluster, the monitored vehicle v_j that exhibits a normal behavior, its good reputation is increased, which is computed as follows: $Grep_{i,j} = \sum_{i=1}^{k} \alpha_1 \cdot i + \beta_1$. Here, k is the number of LIDS agents that detect the vehicle v_j as a normal node. It is noted that, more than half of LIDS agents' confirmation is selected to decide whether a suspect vehicle is an attacker or not since it is an optimal number that satisfies our requirements, i.e. high detection and low false positive rates as proved in our simulation results. Finally, when the GIDS computes the bad and good reputations of a vehicle v_j (or $LIDS_j$), a *ballot_vote* message will be forwarded to RSU for a *final decision-making* as illustrated in Fig. 3. Such message includes the suspect vehicle v_j's id (or suspect $LIDS_j$'s id), the id_s of LIDS agents that detect v_j as an attacker and the good (and bad) reputations values of v_j (or $LIDS_j$). In case, when no RSUs is within the radio range of CH, a *Store-and-Forward (SNF)* mechanism is launched. This mechanism has the ability to store a *ballot_vote* message and forward it periodically until an RSU is found. The choice of the forwarding period is crucial as it impacts the bandwidth resource and end-to-end delay [23]. Therefore, in this study, a forwarding period proposed by the authors in [23] are applied, that is appropriate for the traffic safety applications (see Table 2).

4.3. Global Decision System (GDS)

This system is embedded at each RSU, which has the ability to aggregate the reputations of each vehicle v_j forwarded by the CHs. Afterward, it computes the Trust-level (TL) related to each vehicle v_j and categorize them into a selected list according to their TL.

The aggregated reputation of vehicle v_j ($Arep_{i,j}$) is computed as shown in Eq. (3).

$$Arep_{i,j} = \left(\frac{\sum_{i=1}^{n'} Grep_{i,j}}{n'} - \frac{\sum_{i=1}^{m'} Brep_{i,j}}{m'} \right) \tag{3}$$

where n' and m' are respectively the number of good and bad reputations values delivered by GIDSs.Each RSU computes the TL (see Eq. (4)) and exchanges between each other the value of TL_j related to a vehicle v_j. Afterward, each RSU computes the Global Trust-level (GTL_j), which is equal to $\frac{\sum_{i=1}^{k'} TL_{i,j}}{k'}$, where k' is the number of RSUs that a v_j crossed. Finally as in [24], it categorizes the monitored vehicle according to GTL into three classes (see Eq. (4)). It is noted that, the RSU is assumed to be a trust node and all RSUs are connected through a wired communication by using a TLS (Transport Layer Security) protocol [8].

$$\begin{cases} TL_j = E[Arep_{i,j}] \\ GTL_j \leqslant 0.3, \text{ the vehicle } v_j \text{ is an attacker} \\ 0.3 < GTL_j \leqslant 0.7, \text{ the vehicle } v_j \text{ is a suspected node} \\ 0.7 < GTL \leqslant 1 \text{ the vehicle } v_j \text{ is trustworthy} \end{cases} \tag{4}$$

The vehicle v_j with a GTL_j below 0.3 is assigned as an *attacker* and will be stored in a *Blacklist.* Therefore, the RSU broadcasts such list in order to prevent legitimate vehicles to communicate with them as illustrated in Fig. 3. Furthermore, the vehicles that are classified as *suspected nodes* have limited role in the network, i.e. no possibility to have a CH and LIDS roles. In addition, alert messages like *Post Crash Notification* [4] delivered by *suspected nodes* are ignored by the vehicles. These kind of vehicles are stored in *Suspected list* and forwarded to all vehicles located within RSU's range as illustrated in Fig. 3. Finally, the *trustworthy vehicle* has the ability to play the CH and LIDS roles and all alerts messages that it broadcasts will be taken into account by vehicles. Broadcasting the *Blacklist* and *Suspected list* to all vehicles located within RSU's range could incur a high overhead, specifically if the number of attackers and *suspected nodes* increase. To overcome this issue, the RSUs filters these lists and broadcasts a fraction of *attackers* and *suspected nodes* that have a probability to pass within RSU's range.

Table 2
Simulation setup.

Simulation area	9 km^2
Simulation time	180 s
802.11p maximum range	300 m
Vehicles number	From 50 to 300
Attackers number	45% of overall nodes
Vehicles velocity	90–145 km/h, step 18
Detection period	7 s
TH_{SF}	54% (at $t = 0$) of packets being dropped. This value is updated over the time
TH_{RE}	10% (at $t = 0$) of sent packets compared to neighboring nodes. This value is updated over the time
TH_{PD}	8% (at $t = 0$) of sent duplicated messages. This value is updated over the time
TH_{wo-ssi}	(at $t = 0$) −40 dBm. This value is updated over the time
TH_{wo-pdr}	54% (at $t = 0$) of packets being dropped. This value is updated over the time
TH_{sy}	(at $t = 0$) −38 dBm. This value is updated over the time
SNF period	∼10 s

This reduces the size of lists since they are built based on mobility-perdition of the targeted vehicle as proposed by the authors in [25]. Furthermore, we apply a learning algorithm to predict the direction of targeted vehicles as in [25]. The input vectors of the SVM learning algorithm are: the vehicle's directions, the last RSU where there were attached, and the neighboring RSU's distribution [25]. More details can be found in [25] to predict vehicle's direction.

5. AECFV experimental evaluation

We implemented AECFV in NS-3.17 simulator [26] and compared it with some VANET intrusion detection frameworks: VWCA [2], IDFV [13] and T-CLAIDS [12]. The maximum number of attackers is fixed to 45%. When number of attackers exceeds this value, AECFV performances degrade significantly. The performance metrics are:

- Detection Rate (DR): the number of attacks detected over the total number of attacks.
- False Positive Rate (FPR): the ratio of the number of normal vehicles that are incorrectly identified as attackers over the total number of normal vehicles.
- Detection Time (DT): the time required for intrusion detection agents (LIDSs and/or GIDSs) to detect malicious vehicles. It is computed as follows:

$$\mathrm{DT} = \sum_{i=1}^{n} \frac{D_i - T_i}{\text{Sampling frequency} * n} \tag{5}$$

  where D_i is the detection time of the attacker, T_i is the moment time when the attack starts and n is the number of attackers. This metric is important, specifically in real-time applications since it allows us to evaluate the performances of our framework in terms of fast attack detection.
- Communication Overhead (CO): the amount of information generated by the vehicles to achieve a high level of security.

5.1. Mobility model and simulation setup

The mobility model that is used to generate the traffic has a great impact on the accuracy of the obtained simulation results in vehicular networks [23]. So, in order to simulate realistic vehicle network, we used the mobility model defined in [27]. This generates a trace file that can be used by NS3, including collision free movement, lane changes, and maintaining distance between vehicles [28]. In our simulation we used a Manhattan map provided by SUMO [27]. The simulation area covers $3000 * 3000\ \mathrm{m}^2$, two parallel highways (2×3 lanes) and urban scenario. Main simulation parameters were chosen to be as realistic and are summarized in Table 2. The results are based on averaging the simulation readings obtained from 15 simulation runs.

5.2. Performance comparison

In this subsection, the performance of AECFV with VWCA [2], IDFV [13] and T-CLAIDS [12] are compared by computing the main metrics mentioned above. It should be recalled that the detection frameworks VWCA, IDFV and T-CLAIDS are also embed in NS3 simulator, where the detection techniques applied by such frameworks are explained in the related work section. In the simulations to compute the detection rate, false positive rate and detection time, the attacks mentioned above were introduced separately and the effect of each attack are investigated in isolation. Furthermore, to analyze the communication overhead generated by the detection frameworks all the attacks are introduced simultaneously. It is noted that, the number of selective forwarding, black hole, packet duplication, resource exhaustion, wormhole and Sybil attacks were kept same, where the number of malicious vehicles equal to 45% of overall nodes. This study investigates the effect of scaling mode also by varying the number of vehicles from 50 to 300. The most important results are summarized below.

5.2.1. Detection rate

As shown in Fig. 6, when the number of vehicles increases, the detection rate of AECFV, VWCA, IDFV and T-CLAIDS frameworks decrease. This decrease in the AECFV is much less in comparison to other frameworks. In case when selective forwarding, black hole, packet duplication, resource exhaustion and wormhole attacks occur AECFV and IDFV exhibits a high detection rate, close to 99%. Furthermore, in case when Sybil attack occur, AECFV and T-CLAID have the ability to detect almost of Sybil attacks, where the detection rate of AECFV in the worst case (i.e. the number of vehicles is equal to 300) is close 96%. As a result, AECFV exhibits a high detection rate when the attacks mentioned above occur compared to the current detection frameworks. This result is achieved even when the number of malicious nodes and vehicles are high. The detection's improvement provided by AECFV detection framework is achieved due to the following facts:

(i) *Hybrid detection technique (i.e. rules-based and anomaly detection techniques):* AECFV uses this technique that has the ability to detect with a high accuracy (i.e. high detection and low false positive rates) the malicious node as proved by several research work [1,29]. (ii) *Mutual monitoring*: as explained above, each LIDS (located at the cluster member level) monitors it neighbors vehicle and GIDS (located at CH level), where the GIDS in turn monitors also the LIDSs (located within its cluster). Therefore, this monitoring approach helps to identify any malicious node that occurs within the network.

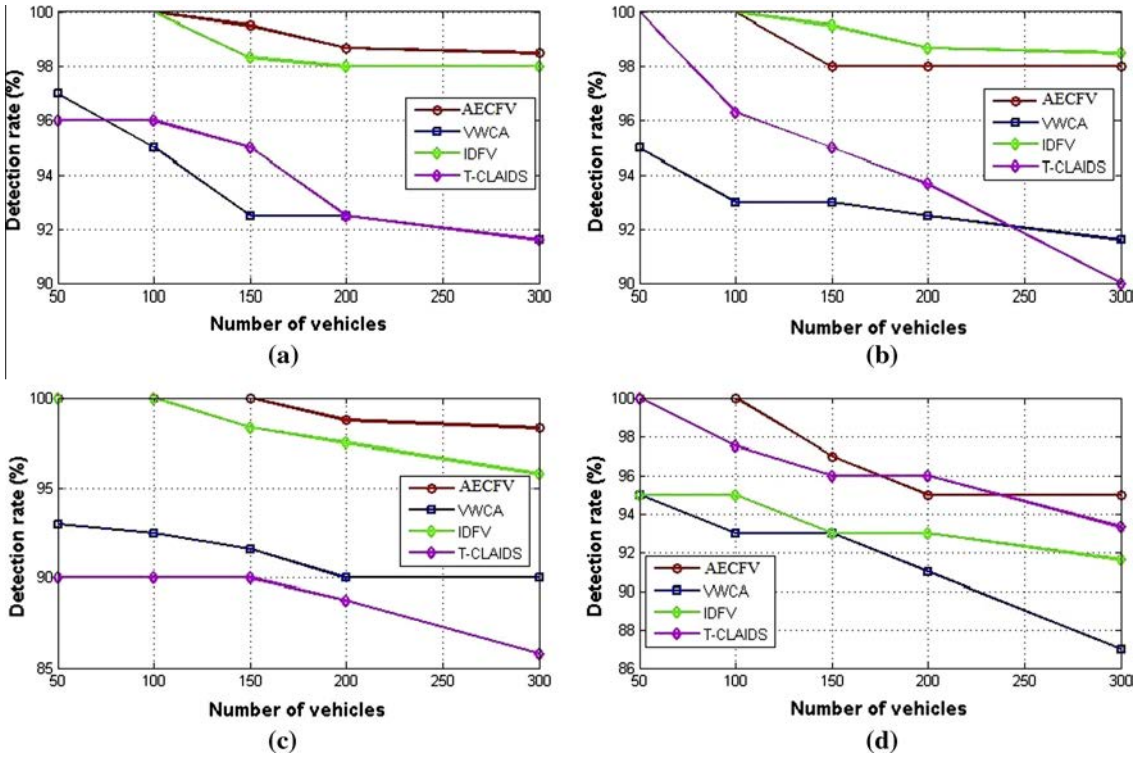

Fig. 6. Detection rate under (a) selective forwarding and black hole, (b) packet duplication and resource exhaustion, (c) wormhole and (d) Sybil attacks.

5.2.2. False positive rate

Fig. 7 shows when the number of vehicles increases, the false positive rate increases. Furthermore, the number of false positive is increasing slowly in AECFV and IDFV compared to VWCA and T-CLAIDS, since when the attacks mentioned above occur their false positive rates are less than 2%. This result is achieved in a worst case (i.e. the number of vehicles is equal to 300). The low number of false positive that AECFV generates is due to the following reasons: (i) *Cooperative detection*: in case when LIDS agent detects an attack it forwards an alarm message to GIDS in order to check the reliability of information forwarded by LIDS, since such agent could be malicious and wrongly accuses the normal node as malicious. Therefore, to decrease the false positive rate, the GIDS applies a vote mechanism and computes the number of LIDSs that detect the node as malicious (see Section 4.2.2, for more details). (ii) *Trust-level:* the monitored node is categorized into three lists (trustworthy, suspected node and attacker) according to its trust-level. This categorization is done by the reputation protocol, which has the ability to evaluate the node's trustworthy-level according to the action taken and information sent. Therefore, embedding the reputation protocol in AECFV decreases the false positive rate. (iii) *Hybrid detection technique:* as mentioned above this technique has the ability to detect the malicious vehicle with a high accuracy (i.e. high detection and low false positive rates).

5.2.3. Detection time

Fig. 8 illustrates the detection time of each security frameworks when all attacks mentioned above occur. It is observed that, when the number of vehicles increases, the required time of intrusion detection agents to detect all malicious vehicles for each security framework increases. Furthermore, according to Fig. 8, the VWCA and T-CLAIDS require a considerable amount of time to detect the attacker compared to AECFV and IDFV, specifically when the number of vehicles increases. In the worst case, when the number of vehicles is equal to 300, AECFV's detection time is 64, 56, 60 and 67 milliseconds (ms) for (selective forwarding and black hole), (packet duplication and resource exhaustion), wormhole and Sybil attacks respectively. This result has been achieved due to two main reason: (i) *Cooperative detection:* as mentioned above there is a cooperation detection between the IDS that is located in the cluster member level (LIDS) and IDS located in the CH level (GIDS) to detect suspected vehicles in a short duration, unlike VWCA and T-CLAIDS where there is no cooperation between IDS nodes, which leads to considerable amount of time spent to confirm the malicious behavior of a monitored vehicle. (ii) *The optimal placements*: an optimal number of IDS agents (i.e. LIDS, GIDS) are activated within a network to identify any suspected node in a short time and reports it promptly to GDS (located at the RSU) in order to take a final decision (normal or malicious).

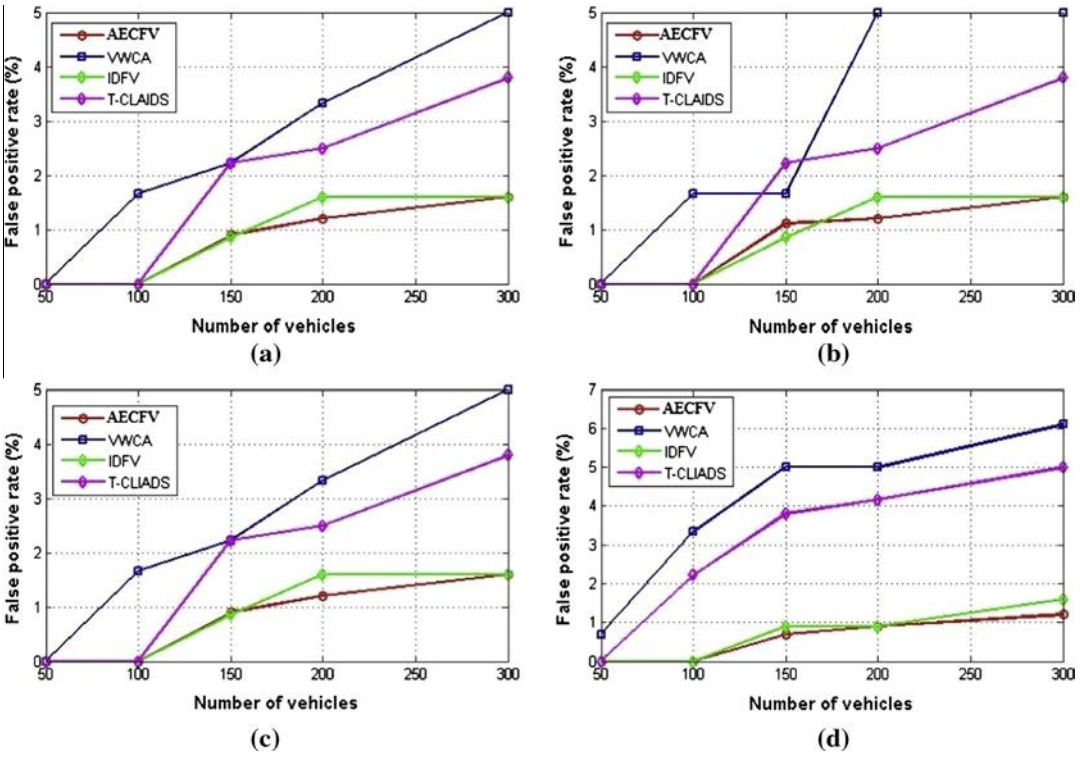

Fig. 7. False positive rate under (a) selective forwarding and black hole, (b) packet duplication and resource exhaustion, (c) wormhole and (d) Sybil attacks.

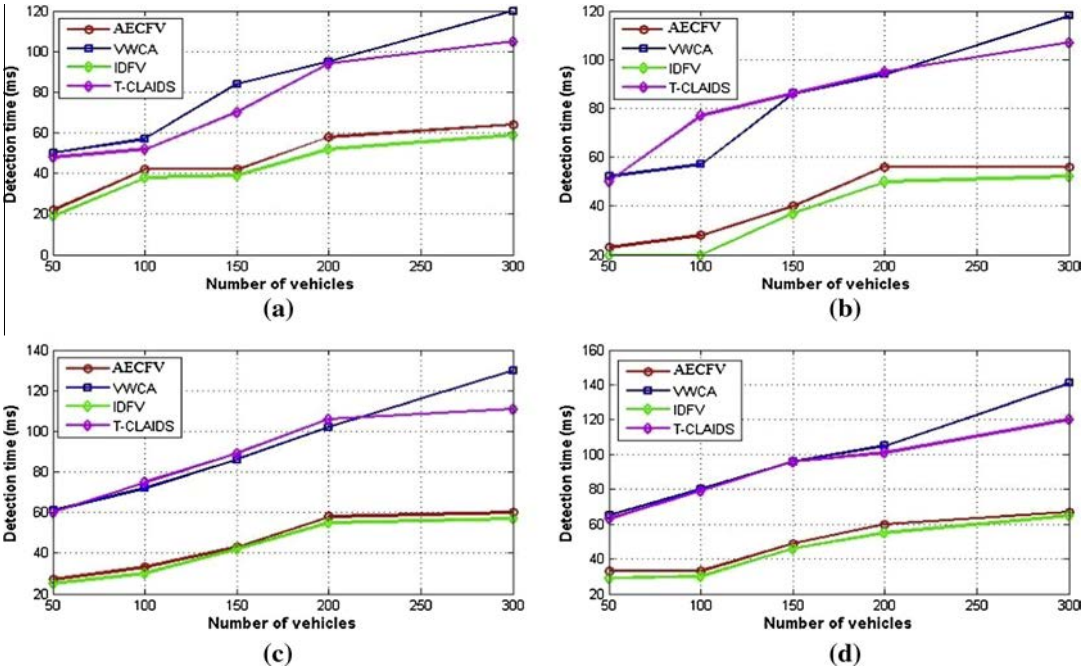

Fig. 8. Detection time under (a) selective forwarding and black hole, (b) packet duplication and resource exhaustion, (c) wormhole and (d) Sybil attacks.

5.2.4. Communication overhead

As illustrated in Fig. 9, AECFV requires a low communication overhead to achieve a high level of security compared to IDFV, VWCA and T-CLAIDS. As shown above, the performances of AECFV and IDFV in terms of detection rate, false positive

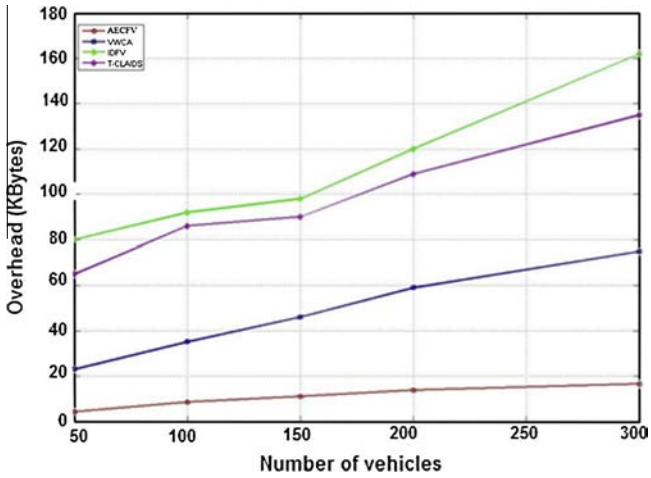

Fig. 9. Comparison of the communication overhead generated by the intrusion detection frameworks.

and detection time are close similar (except when Sybil attack occurs). However, to achieve this high level of security IDFV requires a high communication overhead, specifically when the number of vehicles is high. AECFV framework generates a low communication overhead to achieve a high level of security even in scaling mode and when the number of attackers is higher (i.e. 45% of overall nodes). This result is achieved by the following reasons: (i) AECFV relies on a policy that minimizes the amount of exchanged information between (LIDS and GIDS) and (GIDS and GDS). In fact only the information related to the suspected vehicles is forwarded, e.g. the identity, the reputation values and the attack's features. (ii) An optimal number of IDS agents such as LIDS and GIDS are selected to launch the monitoring process and analyze the behavior of their neighboring nodes, unlike other detection frameworks, where all vehicles within the network launch simultaneously the monitoring process.

6. Security analysis

In this section, we analyze various security aspects of the AECFV, listed as follow:

6.1. Node authentication

To ensure source authentication and authorize only the legitimate vehicles to participate in the network, Elliptic Curve Cryptography (ECC) provided by the current VANET standard IEEE 1609.2 [16] is used, where elliptic curve digital signature algorithm (ECDSA) [30] is used to authenticate the vehicles. ECDSA is recommended by current VANET standards since it generates a fast signature verification [30]. Each vehicle that wants to communicate with its neighbors sign a message using its private key to generate an ECDSA signature, where the receiver verifies the signature by using the sender's public key.

6.2. Communication privacy

The intrusion detection system does not have the ability to ensure the communication privacy and hence it cannot prevent passive attacks. Such attacks aim to overhear the entire message that pass within its radio range. Therefore, to overcome this issue we use elliptic curve integrated encryption scheme (ECIES) provided by IEEE 1609.2 [16] to encrypt the message for data confidentiality. The vehicle encrypts a message with the public keys of its neighbors and decrypts the message using its private key. Furthermore, to assure the confidentiality of subsequent communications, a session key is generated between the vehicles (or between the vehicle and RSU), by using the elliptic curve Diffie–Hellman (ECDH).

6.3. Secure localization

In vehicular networks, vehicles usually discover their neighbors by periodically broadcasting cooperative awareness messages (CAM), in which they claim their identities and location [31]. Furthermore, the malicious vehicle could provide a false location in order to launch a variety of attacks such as black hole or simulate a fake crash and congestion. The Sybil is one of the attacks that create multiple locations. AECFV aims to detect such misbehavior by checking the claimed position of the vehicle (by analyzing the SSI) and ensures the correct location information exchanged with neighboring nodes.

6.4. Security in a large-scale environment

As shown in Figs. 6–9, when the number of vehicles is high (equal to 300), AECFV performs good in terms of attack detection (i.e. high detection rate, low false positive rate and low detection time) and requires a low communication overhead to achieve a high level of security. This result is achieved by the following reasons: (i) *Security performance:* an optimal number of IDS agents are activated as per hierarchy (i.e. at cluster member, CH and RSU levels) to identify any suspected behavior in a short duration. These agents combine the advantage of anomaly and rules based detection techniques which are high detection and low false positive rates respectively [1]. (ii) *Network performance*: the cluster based-architecture reduces the broadcast storms since only the CH has the ability to aggregate and broadcast the information received from its cluster members. In addition, AECFV relies on the detection policy that aims to reduce the information exchanged between IDS agents, since only the information related to the suspected vehicle is exchanged between these agents.

6.5. Vehicle's traceability

Each RSU has the ability to trace the path of a vehicle and predict its direction by applying an SVM-based learning algorithm. According to the simulation results, mobility-perdition algorithm exhibits a high accuracy rate close to 97%. It is proved that each RSU has the ability to be aware with a high accuracy about the next RSU that *attackers* and *suspected nodes* travel. Therefore, the RSU broadcasts only a fraction of *attackers* and *suspected nodes*, which leads to a decrease on the communication overhead.

6.6. Adaptability and extensibility

AECFV intrusion detection framework could be incorporated in all cluster vehicular network architecture. It also allows the detection policies and the cryptography primitives provided by other detection schemas to be added in this framework to prevent other type of attacks that can occur within a network. Furthermore, in our recent work [32] we provide a detection policy to identify the attack that broadcasts a false alerts. Therefore, this detection policy could be embedded in AECFV.

7. Conclusion

AECFV detection framework has the ability to detect the most dangerous attacks that could occur in VANETs such as: selective forwarding, black hole, packet duplication, resource exhaustion, wormhole and Sybil attacks. To the best of our knowledge, we are the first that propose an intrusion detection framework for vehicular networks that uses a set of detection and categorization (reputation mechanism) techniques against the most dangerous attacks. The process of intrusion detection and decision making are carried out at the cluster-members, cluster-heads and RSUs to eliminate any security threat that may disrupt the network in a short duration. According to the simulation results, it is proved that AECFV outperforms existing frameworks in terms of attack detection accuracy (i.e. detection and false positive rates), required time to detect an attack and low communication overhead. These results are achieved even in worst case. The future scope in the CarCoDe project is to embed AECFV in real vehicles and compare simulation and experimental results.

Acknowledgment

This work has been funded by the European Project ITEA2 CarCoDe [33].

References

[1] Sedjelmaci H, Senouci SM, Feham M. An efficient intrusion detection framework in cluster-based wireless sensor networks. Security Commun Networks 2013;6(10):1211–24.
[2] Daeinabi A, Rahbar AGP, Khademzadeh A. VWCA: an efficient clustering algorithm in vehicular ad hoc networks. J Network Comput Appl 2011;34(1):207–22.
[3] Raya M, Papadimitratos P, Aad I, Jungels D, Hubaux J-P. Eviction of misbehaving and faulty nodes in vehicular networks. IEEE J Selected Areas Commun 2007;25(8):1557–68.
[4] Ruj S, Cavenaghi MA, Huang Z, Nayak A, Stojmenovic I. On data-centric misbehavior detection in VANETs. In: IEEE vehicular technology conference (VTC Fall), San Francisco, USA; 2011. p. 1–5.
[5] Kumar N, Chilamkurti N. Collaborative trust aware intelligent intrusion detection in VANETs. Comput Elect Eng 2014;40(6):1981–96.
[6] Hassanabadi B, Shea C, Zhang L, Valaee S. Clustering in vehicular ad hoc networks using affinity propagation. Ad Hoc Networks 2014;13:535–48.
[7] Gazdar T, Benslimane A, Belghith A, Rachedi A. A secure cluster-based architecture for certificates management in vehicular networks. Security Commun Networks 2013;7(3):665–83.
[8] Fadlullah ZMd, Taleb T, Vasilakos AV, Guizani M, Kato N. DTRAB: combating against attacks on encrypted protocols through traffic-feature analysis. IEEE/ACM Trans Netw 2010;18(4):1234–47.
[9] Zhang P. Cooperative location verification for vehicular ad-hoc networks. In: IEEE ICC, Ottawa, Canada; 2012. p.37–41.
[10] Sastry N, Shankar U, Wagner D. Secure verification of location claims. In: Proc. of ACM workshop on wireless security (WiSe), San Diego, California; 2003.
[11] Bißmeyer N, Stresing C, Bayarou K. Intrusion detection in vanets through verification of vehicle movement data. In: IEEE vehicular networking conference (VNC), Jersey City, New York, USA; 2010. p. 166–73.
[12] Kumar N, Chilamkurti N. Collaborative trust aware intelligent intrusion detection in VANETs. Comput Elect Eng 2014;40(6):1981–96.

[13] Sedjelmaci H, Senouci SM. A new intrusion detection framework for vehicular networks. In: IEEE ICC, Sydney, Australia; 10–14 June 2014.
[14] Connell L, Keane MT. A model of plausibility. Cognitive Sci 2006;30:95–120.
[15] Wisitpongphan N, Bai F, Mudalige P, Sadekar V, Tonguz O. Routing in sparse vehicular ad hoc wireless networks. IEEE Trans Veh Technol 2007;25(8):1538–56.
[16] Dedicated Short Range Communication (DSRC). <http://grouper.ieee.org/groups/scc32/dsrc/index.html>.
[17] Sedjelmaci H, Bouali T, Senouci SM. Detection and prevention from misbehaving intruders in vehicular networks. In: IEEE Globecom, Austin, USA; 8–12 December 2014.
[18] Abrougui K, Boukerche A. Efficient group-based authentication protocol for location-based service discovery in intelligent transportation systems. Security Commun Networks 2013;6(4):473–84.
[19] Safi SM, Movaghar A, Mohammadizadeh M. A novel approach for avoiding wormhole attacks in VANET. In: Second international workshop on computer science and engineering, Qingdao, China; 2009.
[20] Abbas S, Merabti M, Jones DL, Kifayat K. Lightweight Sybil attack detection in MANETs. IEEE Syst J 2012;7(2):236–48.
[21] Sedjelmaci H, Senouci SM. Efficient and lightweight intrusion detection based on nodes' behaviors in wireless sensor networks. In: IEEE Global Information Infrastructure Symposium, Trento, Italy; 2013. p. 1–6.
[22] Scholkopf B, Smola AJ. Learning with kernels: support vector machines, regularization, optimization, and beyond. The MIT Press; 2006.
[23] Cherif MO, Senouci SM, Ducourthial B. Efficient data dissemination in cooperative vehicular networks. Wireless Commun Mobile Comput 2013;13(12):1150–60.
[24] Naseri M, Simone A. Evaluating workflow trust using hidden markov modeling and provenance data. Data provenance and data management in eScience studies in computational intelligence, vol. 426. Springer; 2013. p. 35–58.
[25] Zhu H, Lu R, Shen X, Lin X. Security in service-oriented vehicular networks. IEEE Wireless Commun 2009;16(4):16–22.
[26] Network Simulator (NS-3) <http://www.nsnam.org>.
[27] Simulation of Urban Mobility (Sumo) <http://sumo-sim.org/>.
[28] Abumansoor O, Boukerche A. A secure cooperative approach for nonline-of-sight location verification in VANET. IEEE Trans Veh Technol 2012;61(1).
[29] Hai TH, Huh EN, Jo M. A lightweight intrusion detection framework for wireless sensor networks. Wireless Commun Mobile Comput 2010;10(4):559–72.
[30] Huang JL, Yeh LY, Chien HY. ABAKA: an anonymous batch authenticated and key agreement scheme for value-added services in vehicular Ad Hoc networks. IEEE Trans Veh Technol 2011;60(1):248–62.
[31] Rondinone M, Gozalvez J. Contention-based forwarding with multi-hop connectivity awareness in vehicular ad-hoc networks. Comput Network 2013;57(8):1821–37.
[32] Sedjelmaci H, Senouci SM, Abu-Rgheff MA. An Efficient and Lightweight Intrusion Detection Mechanism for Service-Oriented Vehicular Networks. IEEE Internet Things J 2014;1(6):570–7.
[33] ITEA 2 CarCoDe project (2013–2015) <http://www.itea2-carcode.org/>.

Sedjelmaci Hichem is a postdoctoral researcher at DRIVE Lab, university of burgundy, France. He received his Ph.D. degree in telecommunication systems from Tlemecen university, Algeria in 2013. His research interests include Vehicular Networks, Wireless Sensor Network, Unmanned Aerial Vehicle and security issues. He published his work in major IEEE conferences and renowned journals. He is a Member of IEEE.

Sidi-Mohammed Senouci obtained his Ph.D. degree in October 2003 in the Computer Science at the University of Paris 6. From September 2010, he is full professor at University of Bourgogne. His current research interests include Vehicular Communications, Ad hoc and Sensor Networks. He published his work in major IEEE conferences and renowned journals. He is a Member of IEEE.