An Efficient and Lightweight Intrusion Detection Mechanism for Service-Oriented Vehicular Networks

Hichem Sedjelmaci, Sidi Mohammed Senouci, Mosa Ali Abu-Rgheff

An Efficient and Lightweight Intrusion Detection Mechanism for Service-Oriented Vehicular Networks

Hichem Sedjelmaci, Sidi Mohammed Senouci
and Mosa Ali Abu-Rgheff

***Abstract*—Vehicular *ad hoc* networks (VANETs) are wireless networks that provide high-rate data communication among moving vehicles and between the vehicles and the road-side units. VANETs are considered as the main wireless communication platforms for the intelligent transportation systems (ITS). Service-oriented vehicular networks are special categories for VANETs that support diverse infrastructure-based commercial infotainment services including, for instance, Internet access, real-time traffic monitoring and management, video streaming. Security is a fundamental issue for these service networks due to the relevant business information handled in these networks. In this paper, we design and implement an efficient and light-weight intrusion detection mechanism, called efficient and light-weight intrusion detection mechanism for vehicular network (ELIDV) that aims to protect the network against three kinds of attacks: denial of service (DoS), integrity target, and false alert's generation. ELIDV is based on a set of rules that detects malicious vehicles promptly and with high accuracy. We present the performance analysis of our detection mechanism using NS-3 simulator. Our simulation results show that ELIDV exhibits a high-level security in terms of highly accurate detection rate (detection rate more than 97%), low false positive rate (close to 1%), and exhibits a lower overhead compared to contemporary frameworks.**

***Index Terms*—Intrusion detection, service attacks, service-oriented vehicular networks.**

I. Introduction

WITH THE development and advancement of wireless communication technology, researchers conceptualized the idea of vehicular communication networks, also known as vehicular *ad hoc* networks (VANETs). These networks aim to turn cars into intelligent machines that communicate with each other (V2V) or with an infrastructure (V2I) in order to improve traffic safety and comfort of driving [1]. VANETs applications can be categorized into two classes: road-traffic safety and service-oriented applications. With service-oriented applications, road-side units (RSUs) are deployed along the roads for users to request any location-based service (finding restaurants, downloading a map, locating a gas station or a parking space, etc.), internet-based services (multimedia, instant messenger), or real-time traffic concerns. It is expected that service-oriented vehicular networks attract a great deal of investment in large-scale deployment of wireless infrastructures [1], [2].

The success of such service-oriented vehicular networks depends mainly on the underlying communication system, and particularly, the information security since these networks are exposed to attacks generated and handled by these networks [3]. The Intrusion Detection System's (IDSs) techniques show that they are very effective in protecting the network against both internal and external attacks [4]–[8]. Therefore, in this paper, we design and develop an efficient and light-weight intrusion detection mechanism for vehicular networks (ELIDVs) that aim to protect the network against malicious vehicles. In this research work, we focus to detect three kinds of attacks: 1) denial of service (DoS) that aims to disturb the network operation; 2) integrity target that alters the message that is exchanged between legitimate vehicles or provides false information (FI) such as false locations; and 3) false alert's generation that broadcasts a false alert message. ELIDV relies on a set of detection rules related to each attack to model a normal (and anomaly) behavior of a vehicle. Furthermore, with the help of the proposed detection mechanism, we developed a vehicle's behavior evaluation (VBE) protocol that evaluates the trustworthiness level of a vehicle according to its behavior and the available information it provides. We have designed a light-weight detection framework with no need to any additional hardware resource (e.g., firewall) to achieve a high level of security. In addition, ELIDV is extensible for new functionalities that would allow detecting more complex attacks.

We note that, to the best of our knowledge, we are the first dealing with the intrusion issue on *service-oriented* VANETs, and the detection of the most dangerous attacks that could occur in such networks. In fact, most of the works such as [1], [2], [9] apply cryptography techniques to prevent external attackers penetrating the network.

This paper is organized as follows. In Section II, we describe the categories of attacks that can take place on *service-oriented* vehicular networks and highlight some intrusion detection schemas for VANETs proposed in the current literature. In addition, we introduce the network model that we attempt to secure. Section III presents details about our intrusion detection

H. Sedjelmaci and S. M. Senouci are with the DRIVE Laboratory, University of Burgandy, Dijon 58000, France (e-mail: sid-ahmed-hichem.sedjelmaci@u-bourgogne.fr).

M. A. Abu-Rgheff is with the Centre for Security, Communications and Network Research, University of Plymouth, Plymouth PL4 8AA, U.K.

mechanism ELIDV followed by the VBE protocol. The latter is capable of evaluating the trustworthiness level of each vehicle in the network. Section IV provides NS3 simulation results. Finally, in Section V, we conclude the paper and give some perspectives that we envisage to carry out in the near future.

II. Background and Related Work

In this section, we describe main attacks that target service-oriented networks and attempt to detect these attacks using our proposed framework, namely, *Integrity target, DoS*, and *False alert's generation attacks.* Then, we summarize some relevant intrusion detection schemes presented in the literature and discuss their main shortcomings. Finally, we describe the network architecture that we attempt to secure.

A. Common Attacks on Services-Oriented Vehicular Networks

We present three common categories of attacks that may target *services-oriented* vehicular networks and explain their characteristics.

1) Integrity Target Attack: Such an attack aims to alter the message that is exchanged between legitimate vehicles or provide FI such as location. We can cite Sybil attack that we described as follows.

Sybil attack: In vehicular networks, the vehicles usually discover their neighbors by periodically broadcasting cooperative awareness messages (CAMs), in which they claim their identities and positions [10]; thereby, a Sybil node aims to create multiple identities with providing false locations. Furthermore, such threat could also launch a variety of attacks such as DoS or simulate a fake crash, congestion, etc.

2) DoS Attack: The malicious vehicle that launches such an attack aims to disturb the network operation or the used routing protocol. Among the main dangerous DoS attacks, we cite black hole attack, which is briefly described below.

Black hole attack: The vehicle that carries out such an attack aims to drop all the received packets from legitimate vehicles.

3) False Alert's Generation Attack: In this case, the malicious vehicle sends an *alert message* to its k-hop neighbors (RSUs, vehicles) to urge them to take some evasive actions. The idea of the attacker is to send a false alert message in order, for instance, to clear the road for itself or create a traffic jam in the road. We summarize, in the following table, some alert messages.

B. Related Work

IDSs have proved their efficiency to detect intruders with high accuracy compared to cryptography mechanisms [4]–[6]. Recently, several security schemes have used such a system to address security issues in vehicular networks. In [6], the authors propose a security mechanism to detect and then evict the malicious vehicles from the network. To address the intrusion detection issue, they have developed a malicious node-detection system that relies on anomaly-based detection. The latter uses an entropy approach to model a normal behavior (NB) of a monitored vehicle and any deviation from this model is detected as an intrusion. This mechanism aims to detect two attacks: DoS and an attack that collects and disseminates FI (i.e., integrity target attack). According to their simulation results, these attacks were detected with a high accuracy. However, the performance of their scheme decreases when the number of attacks is high. In addition, the collisions rate increases when the density increases. In [8], the authors propose a new detection framework called Trust-aware Collaborative Learning Automata-based Intrusion Detection System (T-CLAIDS) that uses a learning automat technique for anomaly detection. This technique has the ability to model a normal and abnormal pattern with a high accuracy. In this work, the authors aim to detect intruders that collect packets from a legitimate vehicle and disseminate false copies of such information (i.e., integrity target attack). In their simulations, the authors prove that their scheme is able to detect up to around 90%–95% of malicious packets. Nevertheless, embedding such learning algorithm within a vehicular network is computationally expensive and could generate a significant delay, which makes it inappropriate in such networks specifically for safety applications.

In [7], the authors introduce a data-centric intrusion detection scheme to detect the malicious vehicle that generates a false alert, e.g., a false crash. In their approach, a set of predefined rules is used to model the behavior of the vehicle after it sends an alert. Then, the expected behavior and the action that performs by the vehicle after sending alert are compared. When there is no correspondence, this vehicle is identified as malicious that generates a false alert. In their simulation, the authors prove that their approach generates a low overhead compared to other security scheme proposed in the literature. However, the authors did not evaluate the detection accuracy of their scheme when such attacks occur.

C. Network Architecture of Service-Oriented VANETS

The architecture of the *service-oriented VANET* (as illustrated in [2] and [11]) is composed of two layers. The first layer comprises the onboard vehicles and the RSUs wireless communication devices, for communications either between vehicle and road-side infrastructure (V2I) or between vehicles themselves (V2V) using a dedicated short-range communications (DSRC) standard [12]. The second layer is composed of service provider (SP) server and a central authority (CA). Examples for the SP server are traffic control analysis centre [11], multimedia content service, and location-based service [2]. RSUs are connected to the SP server and CA through a wired communication and uses *transport layer security* (TLS) protocol [13].

Communication devices, onboard vehicles, and the RSUs exchange data using multihop routing protocols. In this paper, we use a *greedy forwarding scheme* based on unicast approach, where the forwarder node is selected according to its capability to provide a higher progress toward the final destination (i.e., RSU) [14]. This scheme also uses a store-and-forward (SNF) mechanism, which has the ability to store the information and

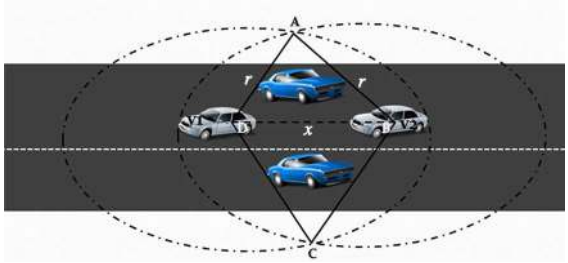

Fig. 1. Guard vehicle neighbors of a link $v_1 - v_2$.

forwards it periodically until it finds the next neighbor or the final destination. The choice of the forwarding period is crucial as it impacts the bandwidth resource and the end-to-end delay. Therefore, the SNF period should take into account the application' requirements (i.e., low bandwidth consumption and low end-to-end delay).

III. ELIDV

IDSs are the most reliable mechanisms to detect internal attacks that are aware of the network's cryptography key. In this paper, we propose an efficient and light-weight intrusion detection mechanism, where IDS agents require a low overhead to detect and prevent, with a high accuracy, three attack categories that are intended to disrupt the networks communication. We embed ELIDV in flat topology that uses a unicast greedy and forwarding protocol.

We first estimate the number of intrusion detection agents that are located within the link's radio range. We then present our detection policies based on rules related to each attacks cited above. Finally, we describe our VBE protocol that has the ability to assign a *malicious level (ML)* to malicious vehicles. Based on the ML, the vehicle is categorized into one of the following classes: *trustworthy, uncertain, or untrustworthy node*.

A. Estimation of the Number of Intrusion Detection Agents

In our proposed detection scheme, each vehicle has the ability to activate an intrusion detection agent in order to monitor its neighbors on a promiscuous mode and apply detection rules related to each attack. The promiscuous mode means that since vehicle v_1 is within the radio range of vehicle v_2, it can overhear communications to and from v_2 as illustrated in Fig. 1.

In this section, we analyze the minimum number of vehicles that have the ability to play the role of an intrusion detection agent, i.e., could monitor a link ($v_1 - v_2$). We call this agent a *guard vehicle*.

Using the greedy forwarding protocol, the farthest vehicle from a communicating vehicle (v_1) forwards the transmitted packets. Therefore, if vehicle v_2 is a forwarder node, it should be located mostly on vehicle v_1's radio-range boundary as illustrated in Fig. 1.

Let x be the distance between v_1 and v_2, and we assume that all the vehicles have the same radio range (r). For any distance x, the radio coverage area where *guard vehicles* are located is the area of the rhombus ABCD subtracted from the area of the sectors ABC and ADC which is calculated as follows [15]:

$$A(x) = 2r^2\cos^{-1}\left(\frac{x}{2r}\right) - 2x\sqrt{r^2 - \frac{x^2}{4}}. \quad (1)$$

The average number of *guard vehicles* for each link is equal to $E[A(x)] * d$, where d is the network vehicles density and $E[A(x)]$ is the expectation of a function $A(x)$, which is computed as follows:

$$E\left[A(x)\right] = \int_0^r A(x)\mathrm{dG}(x). \quad (2)$$

$G(x)$ is the probability distribution of x, which is given by

$$G(x) = \begin{cases} 0, & x \leq 0 \\ \dfrac{x^2}{r^2}, & x > 0 \end{cases}$$

$$E\left[A(x)\right] = \int_0^r \frac{2x}{r^2}\left(2r^2\cos^{-1}\left(\frac{x}{2r}\right) - 2x\sqrt{r^2 - \frac{x^2}{4}}\right)\mathrm{d}x$$

$$\Leftrightarrow E\left[A(x)\right] \approx 0.29\pi r^2.$$

The average number of *guard vehicles*, which depends on the network vehicles density d is given by

$$0.29\pi r^2 d. \quad (3)$$

As a result, we conclude that there is at least one *guard vehicle* at each link to monitor the vehicles that are within its radio range, and the number of these monitoring vehicles depends on the network vehicles density.

B. Rules-Based Intrusion Detection Technique

1) Detection of DoS Attacks

Black hole attack's detection: Black hole attack drops all received packets from legitimate vehicles. Therefore, to detect such attacks, each guard vehicle (G) monitors the behaviors of its neighbors (e.g., G and G' monitor the sender v_1 and a relay node v_2). Furthermore, the greedy forwarding selects a capable forwarder to provide a higher progress toward the destination [14]. Thereby, the sender v_1 knows the position of the relay vehicle v_2 and can use *promiscuous mode* to monitor whether the relay forwards the packets it received or not. In case v_1 could not determine whether v_2 forwards the packets or not, due to the incertitude caused by a high mobility and collisions, it broadcasts a *monitoring alert* to guard vehicles to monitor the link between v_1 and v_2. The guard vehicle (G) captures and stores packets going out of vehicle v_1 and monitors whether v_2 forwards these packets. Therefore, when G finds that v_2 does not forward any packets from v_1, it concludes the following possibilities may have occurred: 1) the forwarded packets could not be observed due to packets collision; 2) vehicle v_2 does not find any neighbors to forward packets and hence a SNF mechanism is launched; or 3) vehicle v_2 carried out a black hole

attack. To determine whether collision or a malicious behavior (MB) is taking place, a cooperative detection is launched. In this detection, the *guard vehicle* cooperates with its guard neighbors (located within radio range of link $v_1 - v_2$) to determine whether v_2 forwards the packets sent by v_1. When *SNF period* is elapsed and all the *guard vehicles* claim that a vehicle v_2 does not forward any packets from v_1, v_2 will be labeled as a black hole attacker. The *SNF period* depends on the greedy forwarding protocol that is used (see Table II).

2) Detection of Integrity Target Attacks

Sybil attack's detection: According to the works in [10], [16], and [17], detection of Sybil attacks usually relies on one of these three detection approaches*: radio resource testing*, *identity registration*, and *position verification*. Our detection mechanism applies a position verification approach to prevent the occurrence of such attack. To estimate the position of a suspected vehicle and hence detect such attack, we propose two detection algorithms that rely on *signal strength intensity (SSI)* and packet's *round trip time (RTT)* to determine accurately the distance (or location) of a vehicle.

The *guard vehicle* monitors the SSI generated by its neighbors (SSIn) and computes the distributed SSI model (SSId) using a shadowing radio propagation model [10] as follows:

$$\frac{P_r(L)}{P_r(L_0)} = -10\beta \log\left(\frac{L}{L_0}\right) + X_{dB}. \tag{4}$$

X_{dB} is a Gaussian random variable with zero mean and a standard deviation σ_{dB}, and β is called the path-loss exponent [10]. L_0 is a reference position.

L is the distance from the guard vehicle G with a position (x_{vG}, y_{vG}) to suspected vehicle v_1 with a position (x_{v_1}, y_{v_1}), computed as follows:

$$L = \sqrt{(x_{vG-}x_{v_1})^2 + (y_{vG-}y_{v_1})^2}. \tag{5}$$

The guard vehicle G obtains the suspected node's coordinates (x_{v_1}, x_{v_1}) through the periodic exchanges of CAM messages. The distributed model SSId at a distance L from the suspected vehicle is computed as follows:

$$\begin{aligned} \text{SSId}(L) &= E\left(\frac{P_r(L)}{P_r(L_0)}\right): \\ &= -10\beta \log\left(\frac{L}{L_0}\right) \end{aligned} \tag{6}$$

where E is the statistical expectation function.

To check whether the suspected vehicle-claimed position is correct, the result of (7) should follow a Gaussian distribution, i.e., it should lie within $\text{Mean}(S) - 3 * \sigma(S_i)$ and $\text{Mean}(S) + 3 * \sigma(S_i)$ [10], [18], where σ is the standard deviation

$$\begin{aligned} S_i &= |\text{SSId}_{ti} - \text{SSIn}_{ti}|, \qquad i = \{1, \ldots, n\} \\ \text{Mean}(S) &= \sum_{i=1}^{n} \frac{S_i}{n} \\ \sigma(S_i) &= S_i - \text{Mean}(S). \end{aligned} \tag{7}$$

TABLE I
ALERTS AND EXPECTED BEHAVIOR [7]

Alerts	Expected behavior
Emergency electronic brake lights (EEBLs)	Vehicle must slow down
Post crash notification (PCN)	Decrease speed until vehicle stops and changes lane
Road hazard condition notification (RHCN)	Decrease speed until vehicle stops and changes route
Road feature notification (RFN)	Decrease speed
Stopped/slow vehicle advisor (SVA)	Change lane and decreas e speed
Cooperative collision warning (CCW)	Slow down
Cooperative violation warning (CVW)	Slow down
Congested road notification (CRN)	lane change
Change of lanes (CLs)	Change lane and slow down

Both SSId and SSIn are computed at each time period ti, and n is the number of observations. When S_i does not follow a Gaussian distribution, the vehicle v_1 is suspected to provide a false location, hence could carry out a Sybil attack. In order to confirm such attack, the packet RTT is computed as in [19], according to the following steps.

The guard vehicle G with a position (x'_{vG}, y'_{vG}) sends to a monitored vehicle v_1 a *request packet*. This vehicle should immediately reply with its position (x'_{v_1}, y'_{v_1}), as it receives this packet. Then, G computes the RTT between G and v_1, RTT_{G-v_1}. G checks whether RTT_{G-v_1} satisfies (8)

$$\text{RTT}_{G-v_1} = \frac{2 * L}{C} + \Delta t \tag{8}$$

where L is the distance from G to v_1 given by (5) and C is the speed of light. Δt is the time delay incurred by vehicle v_1 due to collusions, and processing the incoming packet (Δp) [19]. The guard vehicle estimates the probability of collision (PC), which depends on the vehicle density and speed. A model to estimate this probability is proposed in [20]. When $\text{PC} = 0$ and $\Delta t = \Delta p$, otherwise $\Delta t = \Delta p + \text{back-off time}$. In the *reply packet* sent by vehicle v_1, it adds, with its position, the Δp and back-off time values.

If (8) does not hold, v_1 is detected as a node that provides a false location. As a result, it will be identified as a Sybil attacker.

3) False Alert's Generation Attacks Detection: The detection of such attacks is fundamental for road safety. Here, we focus on alerts that are generated for safety application, which are described in Table I. In the following, we describe our detection policy to identify the malicious node. The vehicle v_1 sends CAM messages periodically to a vehicle v_2 that includes its coordinate $(x_{\text{cam}\,v_1}, y_{\text{cam}\,v_1})$ and the time when a particular message is generated (t_{cam}).

However, when an event occurs, such as crash or road congestion, vehicle v_1 sends *alert messages* to v_2 that include the alert type and the same information as in a CAM message [i.e., time (t_{alarm}) and $v'_1 s$ coordinates $(x_{\text{alarm}\,v_1}, y_{\text{alarm}\,v_1})$]. Node v_2 computes the following parameters t_1, d_1, s_1 before an alert is issued and then computes these parameters again as t_2, d_2, s_2

after the alert generation in order to monitor the behavior of v_1 and detect whether the generated alert is valid. These parameters are defined as follows.

a) t_1 is the period that v_1 spends during the generation of a CAM and subsequent alert message, d_1 is a distance traveled during this period, and s_1 is the average speed. These parameters are computed as follows:

$$t_1 = t_{\text{alarm}} - t_{\text{cam}}$$
$$d_1 = \sqrt{(x_{\text{alarm}v_1} - x_{\text{cam}v_1})^2 + (y_{\text{alarm}v_1} - y_{\text{cam}v_1})^2}$$
$$s_1 = \frac{d_1}{t_1}.$$

b) t_2 is the period that v_1 spends during a generation of an alert message and subsequent CAM, d_2 is a distance traveled during this period, and s_2 is the average speed. These features are computed as follows:

$$t_2 = t'_{\text{cam}} - t_{\text{alarm}}$$
$$d_2 = \sqrt{(x'_{\text{cam}v_1} - x_{\text{alarm}v_1})^2 + (y'_{\text{cam}v_1} - y_{\text{alarm}v_1})^2}$$
$$s_2 = \frac{d_2}{t_2}.$$

Vehicle v_2 applies the values of the computed parameters to compare the behavior of vehicle v_1 after an alert is issued with the expected action provided by the authors in [7] and illustrated in Table I. For instance, when PCN, SVA, or RHCN alerts are raised, v_2 checks v_1's speed: s_1 (before crash detection) and s_2 (after crash detection), and whether v_1 changes a lane (or route). In case the speed s_2 is not decreased and/or it traverses the same lane (or route), v_2 ignores such alert and v_1 will be identified as a malicious vehicle. The expected behaviors when the remaining alerts are raised are given in Table I.

It's important to note that v_2 could provide wrong coordinates information (i.e., false position), such behavior may lead to a Sybil attack. The detection of the Sybil is explained above.

C. Vehicle's Behavior Evaluation Protocol

During the network lifetime, a malicious vehicle could switch to and operate as a legitimate [7]; hence, the behavior of the node oscillates between a legitimate and malicious patterns. Thereby, it is not necessary to disconnect a malicious vehicle from VANET immediately when it exhibits a MB. Furthermore, we assume that a malicious vehicle could carry out several attacks discussed above.

The VBE protocol assigns a ML to a node that carries out malicious activities through processes explained in the following.

1) Vote Process: Malicious vehicles could cooperate between each other and claim that a legitimate vehicle exhibits an attack or vice versa. Therefore, in order to overcome this issue, a vote mechanism is applied. Thereby, when a *guard vehicle* v_i suspects a node as malicious, it forwards *a ballot vote* to nearby RSU encrypted with a session key $\text{KS}_{\text{RSU}-v_i}$. This ballot vote contains *the identity of* v_i, *the identity of the suspected vehicle* v_j, *type of detected attack*, and the *time* when the attack was detected. When the RSU is not within *guard vehicle's* radio range, a store and forward mechanism is launched as explained above. The RSU carries out a vote mechanism to check the detection' reliability claimed by a *guard vehicle* since this latter could provide a false detection, i.e., claim a legitimate vehicle as malicious and hence leading to increase the false positive rate.

Furthermore, during the passage through RSU's radio range, the monitored node can oscillate between a legitimate and malicious modes, so the RSU collects the feedbacks from *guard vehicles* related to this vehicle and computes the attack probability of each *detection period* Δtime as follows:

$$P_{\text{attack}}(\Delta\text{time}) = \frac{\text{detection rate}}{\text{nb\_total}} \tag{9}$$

where detection rate is the number of *guard vehicles* that detect vehicle v_j as malicious during a period Δtime, and nb_total is the number of v_j's *guard vehicles neighbors* during this period. When $P_{\text{attack}}(\Delta\text{time}) > 0.5$, the suspected vehicle v_j is declared as malicious, otherwise *guard vehicle* v_i is designated as a node that provides a false detection. In the following, we explain how to compute a reputation related to each vehicle and their related ML.

2) Reputation and Monitored Vehicles' Categorization: Due to the high mobility of vehicles and the large scale of VANETs, traditional reputation mechanisms applied in MANET cannot be used. Therefore, we propose a reputation mechanism adapted to vehicular networks. In this mechanism, each RSU computes the reputations of the vehicles that are located within its radio range, and the CA aggregates the reputations of all vehicles within the network and assigns a ML to each node that exhibits intrusion attack.

When the RSU confirms the attack that a suspected vehicle exhibits (i.e., $P_{\text{attack}}(\Delta\text{time}) > 0.5$), both guard's good reputation and suspected vehicle's bad reputation are increased. Otherwise, the guard's bad reputation is increased. Furthermore, when a vehicle exhibits a legitimate behavior during its passage through the RSU's range, its good reputation is increased.

According to [7], it is more important to detect FI (e.g., Sybil attacks, false alerts, and false detections) than to detect a MB (e.g., black hole attacks) since FI can cause chaos in the network, i.e., increase the false positives and create a fake congestion. As a result, the vehicle that provides FI, its bad reputation should be increased rapidly compared to the one who exhibits a MB. Furthermore, the *guard vehicle* that provides a correct detection (CD), its good reputation should be increased rapidly compared to the one who exhibits a legitimate behavior. Based on these arguments, exponential and linear functions are used to represent a vehicle that provides FI (or CD) and exhibits a MB (or NB), respectively. Consequently, the reputation of vehicle v_j at RSU level is computed using (10), where the

good and bad reputations are defined as GRep_{v_j} and BRep_{v_j}, respectively,

$$\begin{aligned}
\mathrm{BRep}_{v_j}1 &= \alpha_1 \exp^{FI} + \beta_1 \quad \mathrm{BRep}_{v_j}2 = \alpha_2 \mathrm{MB} + \beta_2 \\
\mathrm{BRep}_{v_j} &= \mathrm{BRep}_{v_j}1 + \mathrm{BRep}_{v_j}2 \\
\mathrm{GRep}_{v_j}1 &= \alpha_1 \exp^{\mathrm{CD}} + \beta \quad \mathrm{GRep}_{v_j}2 = \alpha_2 \mathrm{NB} + \beta_2 \\
\mathrm{GRep}_{v_j} &= \mathrm{GRep}_{v_j}1 + \mathrm{GRep}_{v_j}2 \\
\mathrm{Rep}_{v_j} &= (\mathrm{BRep}_{v_j} - \mathrm{GRep}_{v_j}) \qquad (10)
\end{aligned}$$

here $\alpha_1, \beta_1, \alpha_2, \beta_2 \in [0, 1]$.

The RSU stores the reputation Rep of each vehicle located within its radio range in the Reputation_database as (vehicle v_j, Rep_{v_j}) and periodically forwards this list to the CA. This latter aggregates the reputation of each vehicle in the network and computes the ML using (11) and (12), respectively. Afterward, the ML is compared with a trust formula in (13), proposed in [21]. We note that ML $\in [0, 1]$

$$\mathrm{Rep\_total}_{v_j} = \frac{\sum_{k=1}^{n} \mathrm{Rep}_{vj_k}}{n} \qquad (11)$$

where n is the number of RSUs that compute the reputation of vehicle v_j

$$\begin{cases} \text{if } \mathrm{Rep\_total}_{v_j} < 0, & v_j \text{ is a trustworthy node} \\ \text{Otherwise,} & \mathrm{ML}_{v_j} = E[\mathrm{Rep\_total}_{v_j}]. \end{cases} \qquad (12)$$

Here, E is the statistical expectation of the reputation function $\mathrm{Rep\_total}$

$$\begin{cases} \mathrm{ML}_{vj} \in [0.71] & \text{The vehicle } v_j \text{ is categorized as untrustworthy node} \\ \mathrm{ML}_{vj} \in [0.30.7] & \text{The vehicle } v_j \text{ is categorized as uncertain node} \\ \mathrm{ML}_{vj} \in [0, 0.3] & \text{The vehicle } v_j \text{ is categorized as trustworthy node.} \end{cases} \qquad (13)$$

IV. Performance Evaluation

We evaluated the performance of ELIDV via simulations performed using NS-3.17 [22]. In this section, we discuss the experiment's methodology, highlight our simulation environment setup, and present our main results.

A. Experiment's Methodology and Metrics

ELIDV is compared with the intrusion detection frameworks proposed in [6]–[8]. Specifically, we computed the detection rate, false positive rate, and the overhead. These metrics together with the simulation results from NS-3.17 are used to evaluate the performance of ELIDV. We varied the number of malicious vehicles from 5% to 50% of overall vehicles.

The metrics that we simulated are as follows.

1) *Detection rate* measures the percentage of correctly identified malicious vehicles and their categorization in appropriate lists.
2) *False positive rate* measures the ratio of the number of legitimate vehicles that are incorrectly classified as malicious over the total number of legitimate vehicles
3) *Overhead* computes the cost for securing the communications and detecting intruders. This metric measures the amount of information generated by the vehicle, i.e., communication overhead.

TABLE II
Simulation Parameters

Simulation area	9 km²
Simulation time	180 s
802.11p maximum range	300 m
Vehicles number	From 50 to 300
Vehicles velocity	90 to 145 km/h, step 18
Propagation model	Tow-ray ground
SNF period	~10 s
Detection period (Δtime)	7 s
Mobility generator	SUMO

Main simulation parameters are summarized in Table II and were chosen to be as realistic as possible. Our results are based on averaging the simulation readings obtained from 15 simulation runs.

B. Results Analysis

In this section, we compared ELIDV's performances with local revocation protocol by voting evaluators (LEAVE) [6], data-centric misbehavior detection (DCMD) [7], and T-CLAIDS [8]. In our simulations, we injected separately the attacks cited above and investigated the effects of each attack in isolation by varying the number of intruders from 5% to 50% of overall nodes. We summarized, hereafter, the most important results.

1) Detection Rate: Fig. 2 shows the proposed intrusion detection mechanism ELIDV exhibits a high detection rate when the attacks considered in our paper occurred. Furthermore, ELIDV outperforms other detection frameworks [6]–[8] in terms of attacks detection. In the worst case (i.e., the number of attackers is equal to 50% of overall nodes), the detection rate of ELIDV is equal to 98.4%, 98.66%, and 97.33% when DoS, integrity target, and false alert's generation attacks occurred, respectively. These results are achieved thanks to the cooperative detection between the *guard vehicles* on one hand and the detection rules related to each attack for modeling the legitimate behavior of the nodes.

2) False Positive Rate: Fig. 3 shows when the number of intruders increases, the false positive rate increases. Furthermore, the VBE influences the ELIDV performances. This is evident when considering the performance of framework that did not use VBE protocol (see Fig. 3) where the false positives increase rapidly, specifically when the number of intruders approaching 50% of overall nodes. As a result, we can claim that by using VBE protocol, the number of false positives is increasing slowly even when the number of intruders increased. According to Fig. 3, we can see that ELIDV outperforms the detection frameworks [6]–[8] in terms of low false positive. When the number of intruders is 50% of overall nodes, the false positive rate of ELIDV is 1.33%, 0.6%, and 1.33% for DoS, integrity target, and false alert attacks, respectively. This result is achieved thanks to VBE protocol, which distinguishes with a high accuracy, the malicious vehicle

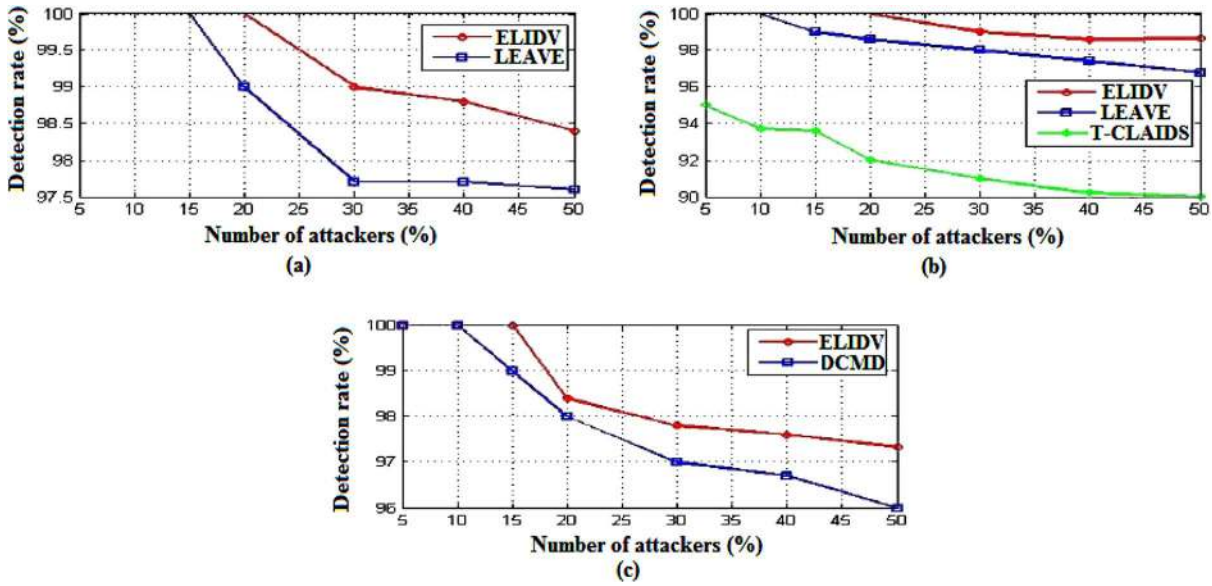

Fig. 2. ELIDV's detection rate under (a) DoS; (b) integrity target; and (c) false alert's generation attacks.

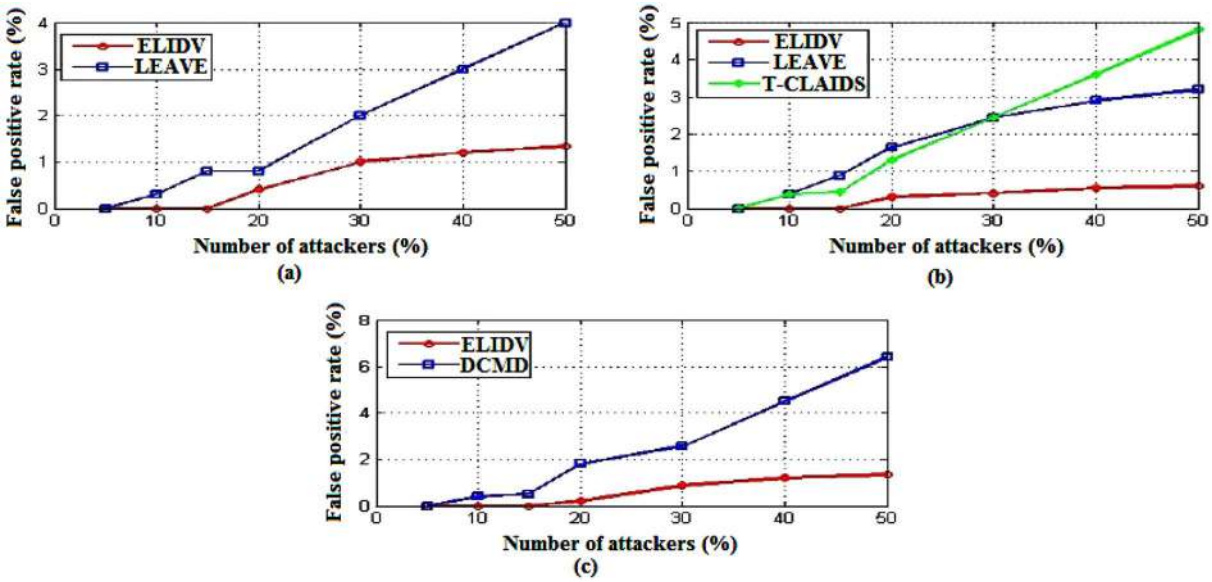

Fig. 3. ELIDV's false positive rate under (a) DoS; (b) integrity target; and (c) false alert's generation attacks.

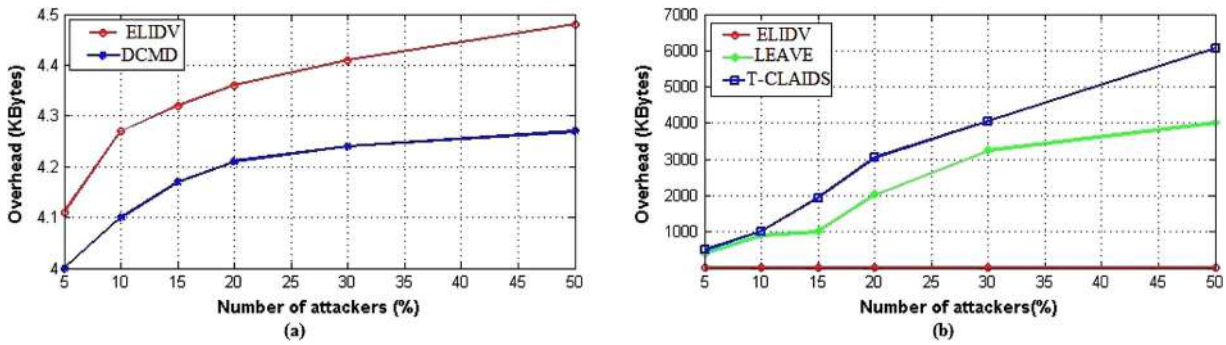

Fig. 4. Comparison of overhead generated by the detection frameworks.

3) Overhead Analysis: Our intrusion detection mechanism has the ability to detect an important number of attacks compared to the detection frameworks mentioned above. On one hand, we can see in Fig. 4(a) that ELIDV and DCMD generate approximately the same overhead. However, DCMD framework detects only one type of attack, which is a false alert

attack. On the other hand, as illustrated in Fig. 4(b), LEAVE and T-CLAIDS frameworks require a high overhead to identify the attackers since they use an anomaly-based detection technique to monitor the behavior of the target vehicle, which is greedy in terms of computing and communication process. As a result, we can conclude that ELIDV requires a low overhead (between 4.1 and to 4.5 kB) compared to LEAVE and T-CLAIDS frameworks. This result is achieved thanks to the following reason: *guard vehicles* rely on a policy that minimizes the amount of exchanged information between each other and transferred to the RSU. In fact, only the suspected vehicle's *id*, *guard vehicle' id*, and the type of detected attack are exchanged.

V. Conclusion

The security in *service-oriented* VANETs is a challenging issue. In this paper, we proposed and implemented ELIDV, a new intrusion detection mechanism, that has the ability to detect both internal and external attacks and differ from the contemporary detection schemes and that only use cryptography algorithms to enhance the privacy and protection for the networks from external attacks. ELIDV detects three kinds of family attacks: DoS, Integrity target, and false alert's generation. Our detection mechanism relies on a light-weight detection technique that uses a set of rules, compared to contemporary detection mechanisms [6], [8] that use a heavy algorithm such as anomaly-detection to model a legitimate vehicle's behavior. Furthermore, using a newly proposed VBE reduces the false positive rate that could incur in the network. This protocol has the ability to evaluate the behavior of the monitored vehicles during their passage through the network and assign a ML to each vehicle according to the attacks they generate. According to our simulation results, the ELIDV mechanism exhibits a high accuracy of attack detection rate (more than 97%) and low false positive rate (close to 1%). In addition, ELIDV generates a low overhead. These results are achieved when the number of vehicles and intruders are equal to 300 and 50% of overall nodes, respectively.

Now, our goal in the CarCoDe project [23] is to embed ELIDV in real-time vehicular network test bed and compare the simulation and experimental results.

References

[1] K. Mershad and H. Artail, "A framework for secure and efficient data acquisition in vehicular ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 62, no. 2, pp. 536–551, Feb. 2013.

[2] H. Zhu, R. Lu, X. Shen, and X. Lin, "Security in service-oriented vehicular networks," *IEEE Wireless Commun.*, vol. 16, no. 4, pp. 16–22, Oct. 2009.

[3] T. Zhang, H. Antunes, and S. Aggarwal, "Defending connected vehicles against malware: Challenges and a solution framework," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 10–21, Feb. 2014.

[4] H. Sedjelmaci, S. M. Senouci, and M. Feham, "An efficient intrusion detection framework in cluster-based wireless sensor networks," *Secur. Commun. Netw.*, vol. 6, no. 10, pp. 1211–1224, 2013.

[5] A. Daeinabi, A. G. P. Rahbar, and A. Khademzadeh, "VWCA: An efficient clustering algorithm in vehicular ad hoc networks," *J. Netw. Comput. Appl.*, vol. 34, no. 1, pp. 207–222, 2011.

[6] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1557–1568, Oct. 2007.

[7] S. Ruj, M. A. Cavenaghi, Z. Huang, A. Nayak, and I. Stojmenovic, "On data-centric misbehavior detection in VANETs," in *Proc. IEEE Veh. Technol. Conf. (VTC Fall)*, San Francisco, CA, USA, 2011, pp. 1–5.

[8] N. Kumar and N. Chilamkurti, "Collaborative trust aware intelligent intrusion detection in VANETs," *Comput. Elect. Eng.*, vol. 40, no. 6, pp. 1981–1996, 2014.

[9] E. Coronado and S. Cherkaoui, "Service discovery, and service access in wireless vehicular networks," in *Proc. SUPE Workshop IEEE Globecom*, New Orlean, LA, USA, 2008, pp. 1–6.

[10] B. Yu, C. Z. Xua, and B. Xiao, "Detecting Sybil attacks in VANETs," *J. Parallel Distrib. Comput.*, vol. 73, pp. 746–756, 2013.

[11] C. Zhang, R. Lu, X. Lin, P. H. Ho, and X. ShenAn, "Efficient identity-based batch verification scheme for vehicular sensor networks," in *Proc. IEEE INFOCOM*, Phoenix, AZ, USA, 2008, pp. 816–824.

[12] *ETSI-World Class Standard Dedicated Short Range Communication (DSRC)*, 2004 [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/index.html

[13] Z. Md. Fadlullah, T. Taleb, A. V. Vasilakos, M. Guizani, and N. Kato, "DTRAB: combating against attacks on encrypted protocols through traffic-feature analysis," *IEEE/ACM Trans. Netw.*, vol. 18, no. 4, pp. 1234–1247, Aug. 2010.

[14] M. Rondinone and J. Gozalvez, "Contention-based forwarding with multi-hop connectivity awareness in vehicular ad-hoc networks," *Comput. Netw.*, vol. 57, no. 8, pp. 1821–1837, 2013.

[15] I. Khalil, S. Bagchi, and N. B. Shroff, "LITEWORP: A lightweight countermeasure for the wormhole attack in multi hop wireless networks," in *Proc. IEEE Int. Conf. Depend. Syst. Netw.*, Yokohama, Japan, 2005, pp. 612–621.

[16] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proc. ACM Int. Workshop Veh. Ad Hoc Netw.*, Philadelphia, PA, USA, 2004, pp. 29–37.

[17] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: Analysis & defenses," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw.*, Berkley, CA, USA, 2004, pp. 259–268.

[18] H. Sedjelmaci and S. M. Senouci, "Efficient and lightweight intrusion detection based on nodes' behaviors in wireless sensor networks," in *Proc. IEEE Global Inf. Infrastruct. Symp.*, Trento, Italy, 2013, pp. 1–6.

[19] N. Sastry, U. Shankar, and D. Wagner, "Secure verification of location claims," in *Proc. ACM Workshop Wireless Secur. (WiSe)*, San Diego, CA, USA, 2003, pp. 1–10.

[20] J. An, X Guo, and Y Yang, "Analysis of collision probability in vehicular ad hoc networks," in *Proc. ACM Genet. Evol. Comput. (GEC'09)*, Shanghai, China, 2009, pp. 791–794.

[21] M. Naseri and A. Simone, "Evaluating workflow trust using hidden Markov modeling and provenance data," in *Data Provenance and Data Management in eScience Studies in Computational Intelligence*. New York, NY, USA: Springer, 2013, vol. 426, pp. 35–58.

[22] *Network Simulator (NS-3)*, 2010 [Online]. Available: http://www.nsnam.org

[23] *European Project ITEA 2 CarCoDe Project (2013–2015)* [Online]. Available: http://www.itea2-carcode.org/