A Macroscopic Traffic Model-based Approach for Sybil Attack Detection in VANETs

Marwane Ayaida, Nadhir Messai, Sameh Najeh, Kouamé Boris Ndjore

A Macroscopic Traffic Model-based Approach for Sybil Attack Detection in VANETs

Marwane AYAIDA[a,*], Nadhir MESSAI[a], Sameh NAJEH[b], Kouamé Boris NDJORE[b]

[a] *CReSTIC, University of Reims Champagne Ardenne Reims, France*
[b] *University of Carthage, Higher School of Communications of Tunis, COSIM Research Lab. Tunisia*

Abstract

Vehicular ad hoc networks (VANETs) are expected to play an important role in our lives. They will improve traffic safety and bring a revolution on the driving experience. However, these benefits are counterbalanced by possible attacks that threaten not only the vehicle's security, but also passengers lives. One of the most common ones is the Sybil attack, which is more dangerous than others since it could be the starting point of many other attacks in VANETs. This paper proposes a distributed approach allowing the detection of Sybil attacks using the traffic flow theory. The key idea here is that each vehicle will monitor its neighbourhood in order to detect an eventual Sybil attack. This is achieved by comparing between the real accurate speed of the vehicle and the one estimated using the V2V communications with vehicles in the vicinity. This estimated speed is obtained using the traffic flow fundamental diagram of the road's portion where the vehicles are moving.

A mathematical model that evaluates the rate of Sybil attack detection according to the traffic density is proposed. Then, this model is validated through some extensive simulations conducted using the well-known NS3 network simulator together with SUMO traffic simulator.

Keywords: ITS, VANETs, Traffic Model, Sybil attack, CAM, Ad Hoc

*Corresponding author
Email address: marwane.ayaida@univ-reims.fr (Marwane AYAIDA)

Network.

1. Introduction

The new mobility challenges of vehicles in Smart Cities need the enhancement of Intelligent Transportation Systems (ITS) that helps to reduce congestion, accidents, fuel consumption, etc. Thus, Vehicular Ad Hoc Networks (VANETs), which are a major component of ITS, have been a subject of some intensive research and experimental applications in these last two decades. In such networks, vehicles on the road will communicate with each others to exchange information about their directions, their speeds, their positions, the state of road, etc. Currently, the automotive industry is working to equip new vehicles with Wireless Access Vehicular Environment (WAVE) devices [1]. WAVE protocols are based on the IEEE 802.11p standard and provide the basic radio standard for dedicated short-range communication (DSRC).

Since a successful attack could have dramatic consequences, security of Vehicular Ad Hoc Networks becomes an important issue. A well known attack is the Sybil one. This attack is considered as one of the most dangerous and the basis of many other attacks [2]. In Sybil attack, malicious node may assume multiple identities. The least harmful objective of such attack is to create an illusion of traffic congestion in order to reroute other vehicles from the road that the attacker will take. At the other end, the attacker could push a specific vehicle to take a particular route in order to trap it or, even, guide it straight to crash in an accident. Therefore, detecting such attacks is very sensitive for several safety, privacy and security reasons.

This paper presents a new technique allowing the detection of Sybil attack in VANETs networks. This approach exploits some traffic flow theory phenomena in order to generate a residual corresponding to the difference between the measured speed of the vehicle and the estimated speed in a distributed way by using the information of its surrounding. A significant deviation of this residual from a predefined threshold is considered as an indicator of a potential

attack. Once a vehicle detects an attack, it notifies its neighbours allowing the detection of this attack thanks to a collaboration between the honest vehicles. Finally, some realistic simulation using NS3 network simulator combined with a mobility simulator SUMO are provided to demonstrate the efficiency of our proposed technique.

This paper makes the following contributions:

1. In contrast with the existing works that are generally based on secure key, resource testing, reputation or position verification techniques, this paper proposes a new approach based on the macroscopic traffic flow theory to detect Sybil attacks.

2. It proposes a mathematical model allowing to evaluate the closed form of the Sybil attack probability according to the proposed approach.

3. It presents a new technique which is easy to be implemented and does not need neither a central node nor additional hardware in the vehicle.

4. It provides realistic simulations using well-known tools to show the efficiency of our approach.

The reminder of this paper is as follow. The section 2 describes some related works about Sybil attacks detection. Section 3 introduces the context of our study by presenting the targeted scenario and the used traffic flow model. Section 4 details how our detection algorithm works. Section 5 presents a mathematical model that evaluates the detection rate of the proposed Sybil attack algorithm. Section 6 validates this algorithm using a realistic network simulation. Finally, section 7 concludes this paper and gives some perspectives to this work.

2. RELATED WORKS

Several mechanisms aiming to detect Sybil attacks have been proposed in the literature. Among them, we can mention those based on resource testing

[3] (i.e. computing ability, storage ability, communication bandwidth, etc.). In that case, each vehicle broadcasts a request to all its neighbours, this needs some physical resources to be computed. Thus, since attackers have to reply simultaneously for them and for the created fake nodes, they will not be able to reply in the given time interval and only honest vehicles will be trusted. However, this approach wastes a lot of computing resources and bandwidth for these tests. Moreover, attackers equipped with powerful computing devices can bypass these tests.

Authors in [4] proposed a RSSI-based localization technique denoted INTERLOC, which uses the mobile nodes as a support to localize accurately a neighbour node. It is used mainly to cancel the GPS signal interference effects, while it can also be used in detecting Sybil attacks. Mobile nodes assist a node on finding its accurate position using the received RSSI. This mechanism is based on the signal strength and its arrival angle. This can be used to detect that the received signal could not be the received one from a declared position in case of a Sybil attack. Therefore, this technique needs to use many neighbours, and then a high vehicle density, in order to accurately localize a node. Besides, it does not work properly in a highway scenario were the arrival angles are around 0 or 180 degree.

[5] proposed a new way to detect Sybil attacks, based on electro-acoustic positioning. Simulations in [5] showed that the electro-acoustic positioning outperforms RSSI-based positioning. Despite its efficiency, electro-acoustic positioning suffers from a major drawback, which is that every vehicle has to be equipped with an acoustic ultrasound beeper. This assumption is very strict in reality, since few vehicles are equipped with such devices.

[6] presented an encryption protocol to avoid Sybil attacks in VANETs. This protocol uses the link between a vehicle and a Road Side Unit (RSU) to exchange encrypted messages in order to obtain the network key, that allows it to communicate with other vehicles. Since this key is managed by the RSU, a node with a unique ID could obtain only one network key, and then it is not able to launch Sybil attacks. This protocol needs that all messages to be encrypted,

which is in opposition to the paradigm of C-ITS where security data have to be exchanged in an open way. Moreover, if a hacker succeeds in infiltrating the server where the vehicles' secret IDs are stored, he could use them to generate as much identities as needed to orchestrate a Sybil attack.

Authors in [7], proposed another totally distributed secure positioning algorithm based on an allocating number verification and on the mutual guarantee relying on neighbours for Wireless Sensor Networks (WSN). This algorithm uses a lot of messages in order to guarantee the correct location of nodes, which increases the cost of such algorithm.

[8] defined a protocol allowing the detection and the prevention from Sybil attacks in Mobile Ad-hoc Networks (MANETs). This protocol is based principally on clustering and path similarity. It uses the packet authentication by exploiting the electronic signature combined with private keys. The management of such keys is very challenging especially in a mobile networks.

Since vehicles are exchanging information about their status (position, speed, direction, etc.), these elements can be used to detect Sybil attacks like in the works [9] and [10]. In [9], authors proposed "Footprint", a detecting Sybil attack approach in urban networks. Footprint uses social relationship among trajectories to detect and eliminate sybil nodes. However, it supposes that roads are covered with Road Side Units (RSUs), which is very expensive for road operators. Besides, it supposes that all RSUs are considered as honest ones, which is not necessarily true. The work in [10] studied the driving pattern similarity in the Sybil attack detection. It classifies neighbours using the k-Nearest Neighbours algorithm in order to find malicious nodes. This algorithm seems to be efficient as demonstrated by the simulations, although its complexity is too high to be used in real time in attack detection within vehicles.

For more Sybil attack detection related works, readers are encouraged to read the survey presented in [11].

Another common used solutions for defending against Sybil attacks are based on Public Key Infrastructure (PKI). Since the vehicle can be authenticated with its public key, which is supposed to be unique, and managed by the Root

Authority (RA), an attacker can be detected at any time.

PKI-based approaches are complex and expensive to be implemented in terms of needed equipments. For example, we have to deploy a Root Authority (RA), a Long-Term Certificate Authority (LTCA) and a Pseudonym Certificate Authority (PCA), knowing that the PCA has to be reached by the vehicles in order to download new Pseudonym Certificates (PCs). Therefore, vehicles have to access to the PCA through the Road Side Units (RSUs). The deployment of these RSUs is estimated to end by 226 with a cost of 660 M €[12]. Another alternative is to take advantage of the existing cellular networks to download certificates. However, drivers have to pay this access. Moreover, vehicles will overload the cellular network if they use this media since it was not initially sized to manage this task. Moreover, even if a vehicle with a valid Long-Term Certificate (LTC) is corrupted, but not yet identified as it is, it can continue to download PCs as needed. Therefore, the PKI protection stills available for new vehicles but not really for already involved corrupted vehicles. Since all the nodes are perceived as honest by each others, this makes the detection of Sybil attacks very difficult and subsequently more difficult the defense against them [13].

To tackle some limitations of the overviewed approaches, we propose in this paper to design an original Sybil attack detection mechanism, which takes benefit from the traffic flow model already provided to the vehicle in order to detect Sybil attacks. The proposed mechanism is easy to implemented and very powerful, as it is demonstrated through a mathematical model and some realistic simulations.

3. Background

The traffic flow study is generally based on some models that can be microscopic or macroscopic. Microscopic models focus on the individual behaviour of drivers, while the macroscopic models consider the flow of vehicles. Microscopic models can deal with heterogeneous vehicles and stochastic aspects to provide

some accurate information about the individual speeds, the space-time diagram, etc. However, since these models require a large number of variables and a high calculation time, they are not suitable for the development of a real-time attack detection algorithm. On the other hand, macroscopic models require a relatively low computation time and can be more adapted for our study.

Macroscopic models [14, 15, 16, 17, 18] consider traffic flow as fluid that can be described using the hydrodynamic theory and some aggregated quantities, which are the speed (v), the flow (q) and the density (d). Moreover, all macroscopic models define a flow-density relationship often called the *Fundamental Diagram* (*FD*), which is obtained either from the calibration of a particular microscopic model or from the experimental data. Thus, without loss of generality, the remainder of this paper considers that the FD has a triangular form described by the following equations:

$$\begin{cases} \text{if } d < d_c, \text{ then } q = ad \\ \text{if } d_c \leq d < d_{max}, \text{ then } q = bd + c \\ \text{if } d \geq d_{max}, \text{ then } q = 0. \end{cases} \tag{1}$$

where, a, b, c, d_c, d_{max} *and* q_{max} are some given parameters that depends on the road's section as depicted by figure 1.

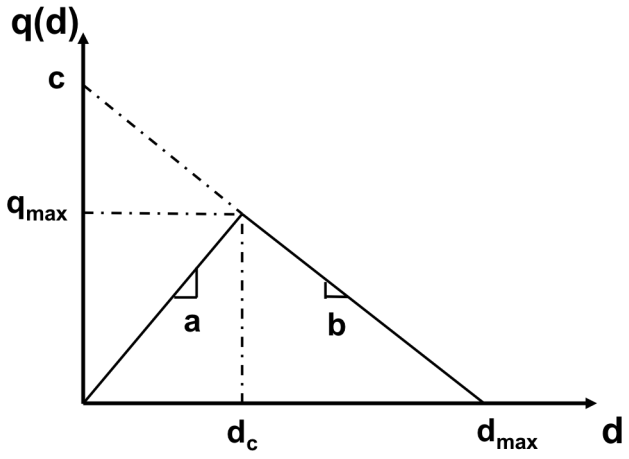


Figure 1: Fundamental diagram example

In order to exploit this traffic model in the context of VANETs, this study considers each vehicle, as a Cooperative-Intelligent Transportation System Station (C-ITSS), exchanging cooperative messages, denoted as Cooperative Awareness Message (CAM). Every vehicle sends standard CAM messages at a frequency varying between 1 and 10 Hz. This frequency depends principally on the vehicles speed. These messages aims at notifying the neighbors of their presence. The fields present in the CAM messages are presented in the Figure 2.

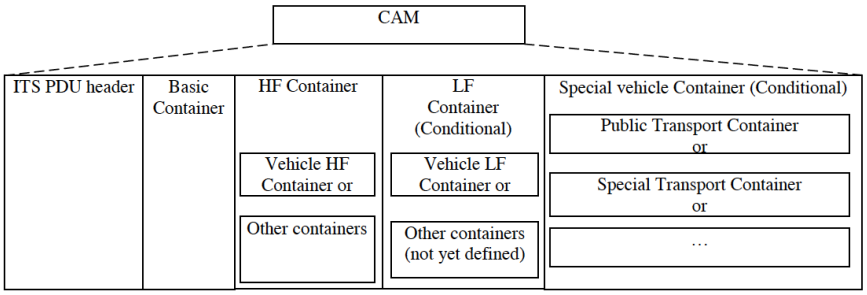


Figure 2: Structure of a CAM packet

As it is shown from figure 2, the fields which are included in the CAM message are principally:

- ITS PDU header: it includes principally the C-ITS station ID.
- Basic Container: it indicates the station type (pedestrian, cyclist, passenger car, bus, etc.) and the station position (latitude, longitude, altitude).
- High Frequency Container : it notifies for the data changing frequently (speed, orientation, curvature, yaw rate, lane position, steering wheel angle, etc).
- Low Frequency Container : it includes the role of the vehicle (public transport, special transport (2), dangerous goods (3), road work (4), emergency, etc.), exterior light status and the path history (history of last stored positions)

- Special Vehicle Container : it depends on the declared role of the vehicle, every role has its own special container, which indicates for example if the light bar and the siren are used or not.

4. Presentation of the proposed algorithm

First, we present in this section a high-level description of our proposed algorithm presented under a flowchart in the following subsection. Second, each step of this algorithm is detailed as a pseudo-code algorithm.

4.1. Algorithm high-level description

The Figure 3 depicts a flowchart describing a high-level view of the proposed algorithm. The vehicle waits for the reception of a CAM (Cooperative Awareness Message) to start updating the list of its neighbours in order to add the source node as a neighbour if it does not already exist, since met for the first time. Otherwise, it updates the timestamp of this node. This is done continuously as stated by the CAM standard.

In order to reduce the overcrowding of the vehicle processing hardware, the Sybil attack detection procedure is launched according to the predefined period (Δ_{det}). Once the detection timer (T_{det}) expires, the vehicle collects all the processing needed data. First, it extracts from the list of its neighbours those that still in its vicinity. To do so, it refreshes the list by removing the nodes that were not seen for a given time. The used duration for this cleaning could be tuned depending on the scenario and the used environment (urban, highway, etc.). Secondly, the fundamental diagram is used to estimate the speed of the vehicle (V_{est}). Note here that the FD could be already integrated directly within the On-Board Unit (OBU) of the vehicle or it could be downloaded from a Road Side Unit (RSU) at the city's entry for example. Then, it measures the real speed (V). After that, the detection of the Sybil attack is achieved by a comparison of these two speeds, as $|V - V_{est}|$. If this difference is lower than a predefined threshold value (V_{th}), then no attack is detected. Otherwise, an orchestrated

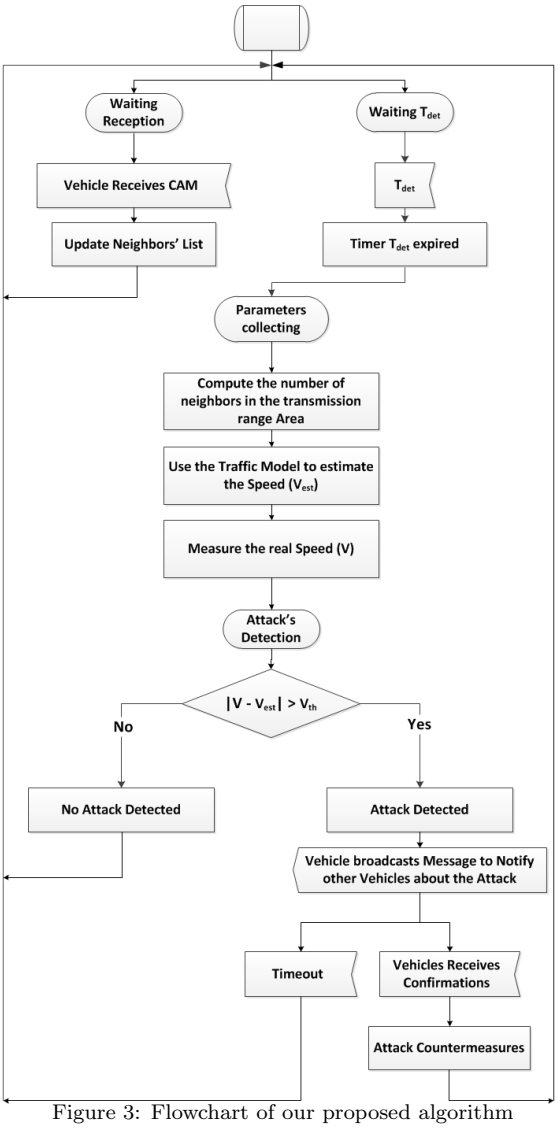


Figure 3: Flowchart of our proposed algorithm

Sybil attack is detected and the affected vehicle will notify its neighbours of this attack. Finally, if the vehicle receives at least a confirmation from another vehicle that has do the same analysis, it will launch some already prepared countermeasures, which are out of scope of this paper. Otherwise, it will ignore this attack detection and restart another attack detection at the next period.

To facilitate the reader's understanding of our contribution, we add a state

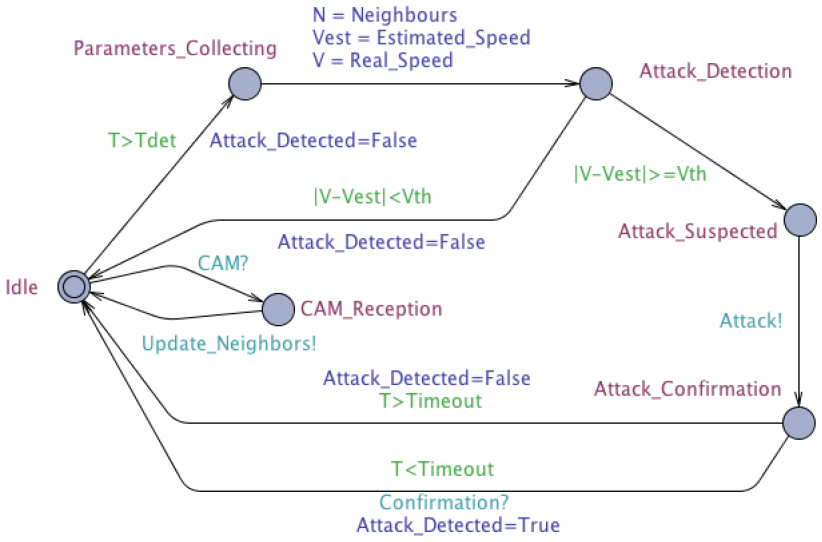


Figure 4: State transition diagram of our proposed algorithm

transition diagram, presented in the figure 4, to support the Flowchart. This diagram shows that a vehicle starts either by receiving a CAM and then updating its neighbours or by collecting the parameters that allow to detect the attack if the detection timer expires. If the difference between the estimated and the real speeds is higher than V_{th}, then an attack is suspected. Thus, a vehicle sends an alert to its neighbours. If it receives a confirmation, the attack is confirmed. Otherwise, it returns to the Idle state to restart another detection cycle.

4.2. Algorithm detailed description

To better understand the proposed algorithm, we present here its most important features depicted in the following algorithms :

- Algorithm 1 details the procedure of the neighbours' list updating.
- Algorithm 2 presents the procedure of the neighbours' number computing.
- Algorithm 3 shows the procedure that uses the traffic model to estimate the speed of the vehicle.

- Algorithm 4 describes the detection algorithm that uses the three previous algorithms.

Algorithm 1 Updating Neighbours' List

```
1: procedure NeighboursUpdating(Packet P, List Neighbours)
2:     if (P.Sender ∉ Neighbours) then
3:         Neighbours.Add(Sender);
4:     end if
5:     Neighbours[Sender].Timestamp ← P.Timestamp;
6:     Neighbours[Sender].Location ← P.Location;
7: end procedure
```

In the Algorithm 1, when a vehicle receives a CAM message, it verifies if it exists already in the neighbors list. If it is the case, it has to update the location and the timestamp of the sender vehicle. Otherwise, it adds the new sender of the message as a neighbor and it inserts both its location and timestamp.

Algorithm 2 Computing Number of Neighbours

```
1: procedure NumberNeighbours(List Neighbours)
2:     int T_th : freshness of neighbours
3:     int N : number of neighbours
4:     foreach (n ∈ Neighbours) do
5:         if ([Now − Neighbours[n].Timestamp] > T_th) then
6:             Neighbours.Remove(n);
7:         end if
8:     end for
9:     N = Neighbours.size();
10:    return N;
11: end procedure
```

When a vehicle needs to compute the number of its neighbours, it starts by updating the list of its neighbours as detailed in the Algorithm 2. To do so, it

verifies if the time since the receiving of the last message is higher than a given threshold, denote T_{th}. If so, it removes the neighbour from the list. The number of neighbours is then evaluated as the cardinality of the derived updated list.

Algorithm 3 Estimating Speed

```
1: procedure SpeedEstimation(List Neighbours)
2:     double V_est : estimated speed
3:     double a : constant of fluid area
4:     double b : constant of congested area
5:     double d_c : critical density
6:     double d_max : maximal density
7:     double c = −b ∗ d_max : second constant of congested area
8:     double density : current density
9:     int Length : length of the segment
10:    density = (NumberNeighbours(Neighbours)+ 1) / Length;
11:    if (density < d_c) then                                  ▷ Fluid Area
12:        flow = a ∗ density;
13:    else if (density < d_max) then                           ▷ Congested Area
14:        flow = b ∗ density + c;
15:    else
16:        flow = 0;                                            ▷ Traffic Jam
17:    end if
18:    Vest = flow/density;
19:    return V_est;
20: end procedure
```

On the other hand, the vehicle's speed is estimated based on the macroscopic traffic model using the Fundamental Diagram (FD) as described by the Algorithm 3. To do this, we exploit the characteristics of the FD of the section in which the vehicle moves. This FD is supposed to be already stored in the vehicle within the map or downloaded from some specific RSUs.

Algorithm 4 Attack Detection

```
1: double V_est : estimated speed
2: double V : real speed
3: double V_th : threshold to detect a Sybil attack
4: List Neighbours : list of Neighbours
5: Time T_det : timer of periodic attack detection triggering
6: Time DetectionTime : timestamp of the attack detection
7: int Timeout : maximum waiting time for an attack confirmation
8: Packet CAM : packet CAM received
9: while (ReceiveCAM(CAM)) do
10:     NeighboursUpdating(CAM, Neighbours);
11: end while
12: if (T_det is expired) then
13:     V_est = SpeedEstimation(Neighbours);
14:     V = Node.Mobility.GetSpeed();
15:     if (|V − V_est| > V_th) then
16:         DetectionTime = Now;
17:         BrodcastMessage(\AttackDetected");;
18:         Wait(Timeout);
19:         if (ReceiveConfirmation()) then
20:             LaunchCountermeasures(Sybil);
21:         end if
22:     end if
23:     T_det is armed
24: end if
```

The first step of the Algorithm 3 is to estimate the density, which depends on the length of the portion's road and the number of neighbours estimated according to the Algorithm 2. Note here that due to the constraints on the transmission range, each real road's portion is decomposed in some virtuel segments with a length (*Length*) corresponding to the OBU's transmission range

and characterized by the same FD. Once the density is estimated, the traffic flow is obtained using the FD characteristics according to the traffic's state (fluid, congested and jam). Finally, the speed is estimated using the equation 1 of the traffic model.

The Algorithm 4 presents the Sybil attack detection procedure. When the vehicle receives a CAM messages, it updates the list of neighbours according to Algorithm 1. Simultaneously, it waits for a notification about the expiration of the timer T_{det}. Once this happens, it estimates the speed of the vehicle using the Algorithm 3. The difference between the real measured speed V and the estimated one V_{est}, which is denoted as $|V - V_{est}|$, is computed and then compared with a predefined threshold V_{th}, which obviously depends essentially on the road and the traffic model. It could be given also as an input with the FD. An example of fixing this threshold for a segment is studied in the subsection 6.1.2.

If this difference $|V - V_{est}|$ is higher than the threshold V_{th}, a Sybil attack is then detected. Therefore, the vehicle broadcasts a notification message to its neighbours. If no confirmation is received after a custom duration (i.e. $DetectionTime$), the detection is considered as a false positive warning and subsequently neglected. On the other hand, if at least one other vehicle makes the same conclusion, the attack is confirmed and the vehicle will launch some countermeasures, which are out of focus of this paper.

One of the most important advantages of this standalone algorithm is the fact that, it is deployed within the vehicle without needing neither extra hardware than the OBU, nor extra messages than CAM messages. It is regularly executed to monitor the neighbours in order to detect any attack.

After this description of the proposed approach, the following sections are dedicated to the validation of our algorithm through a mathematical model and some realistic simulations.

5. Mathematical Model Analysis

In order to validate the results of the proposed algorithm in the previous section, we investigate in this section the theoretical expression of the sybil attack probability, denoted as P_{sya}. First, a proposed scenario is considered in order to define all the parameters of the system and to mathematically formulate the problem. Second, a proposition is investigated in this section, to seek the exact expression of the probability P_{sya}. Third, the performances of our proposed algorithm are detailed in order to validate the obtained theoretical results in terms of sybil attack probability P_{sya}.

5.1. Problem Formulation

Our aim in this section is to determine the closed form of the sybil attack probability P_{sya}, which can be defined as

$$P_{sya} \triangleq \mathbb{P}(|V - V_{est}| \geq V_{th}) \tag{2}$$

We consider in this section N_0 vehicles that actually exist and N_1 attacking vehicles (virtual fake nodes). We denote α as the percentage of attacking vehicles, defined as follows

$$\alpha = \frac{N_1}{N_0} \tag{3}$$

In the rest of the section, the velocity V of the vehicle is defined, according to the kind of traffic, as

$$\begin{cases} V = v_{m1} + \Delta V_1 : \text{ if the traffic is fluid} \\ V = v_{m2} + \Delta V_2 : \text{ if the traffic is congested} \\ V = v_{m3} + \Delta V_3 : \text{ otherwise (traffic jam)} \end{cases} \tag{4}$$

where, v_{mi} is the mean velocity and ΔV_i is a gaussian distribution with zero mean and standard deviation σ_i, according to the traffic state i: $i = 1$: fluid, $i = 2$ congested and $i = 3$ jam. Hence, the density d is expressed as

$$d = \frac{N_0 + N_1}{L} \tag{5}$$

5.2. Sybil attack probability estimation

In order to determine the exact expression of P_{sya}, we assume the following assumptions:

Assumptions:

We assume in this section that:

- H_1**:** $|V - V_{est}| = V - V_{est}$
- H_2**:** According to the three possible types of traffic depicted in equation 1, the following events: 1) the traffic is fluid, 2) the traffic is congested and 3) the traffic is blocked, are statistically independents.

Proposition 1. *Within the hypothesis of triangular FD and the two previous assumptions H_1 and H_2, the closest form of the sybil attack probability P_{sya} is given by*

$$P_{sya} \triangleq \frac{1}{2}[p_1 erfc(\frac{V_{th} - v_{m1} + a}{\sqrt{2}\sigma_1}) + p_2 erfc(\frac{V_{th} - v_{m2} + b}{\sqrt{2}\sigma_2} + \beta) + p_3 erfc(\frac{V_{th} - v_{m3}}{\sqrt{2}\sigma_3})], \tag{6}$$

where, "erfc" is the Complementary Error Function, defined as

$$erfc(x) \triangleq \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-y^2} dy \tag{7}$$

and p_1, p_2, p_3 and β are defined as follows

$$\begin{cases} p_1 = \mathbb{P}(\alpha < \frac{Ld_c}{N_0} - 1) \\ p_2 = \mathbb{P}(\frac{Ld_c}{N_0} - 1 \leq \alpha < \frac{Ld_{max}}{N_0} - 1) \\ p_3 = \mathbb{P}(\alpha \geq \frac{Ld_{max}}{N_0} - 1) \\ \beta = \frac{cL}{(\alpha+1)\sqrt{2}N_0\sigma_2} \end{cases} \tag{8}$$

Proof 1. *Thanks to the assumptions H_1, H_2 and to the total probability theorem, we can obtain*

$$P_{sya} = P_{sya}^1 + P_{sya}^2 + P_{sya}^3, \tag{9}$$

where,

$$\begin{cases} P^1_{sya} = \mathbb{P}(V \geq V_{est} + V_{th}, d < d_c)\text{: fluid traffic} \\ P^2_{sya} = \mathbb{P}(V \geq V_{est} + V_{th}, d_c \leq d < d_{max})\text{: congested traffic} \\ P^3_{sya} = \mathbb{P}(V \geq V_{est} + V_{th}, d_c \geq d_{max})\text{: jam traffic} \end{cases} \tag{10}$$

300 *Next, we propose to determine the expression of each term of the previous expression, according to the kind of traffic:*

1. *Fluid traffic: By applying the Bayes theorem, the first term P^1_{sya}, of the equation (9) can be expressed as*

$$P^1_{sya} = \mathbb{P}(V \geq V_{est} + V_{th} | d < d_c)\mathbb{P}(d < d_c) \tag{11}$$

Using equation 1 and according to the fluid state, we get

$$\mathbb{P}(V \geq V_{est} + V_{th} | d < d_c) = \mathbb{P}(V \geq \frac{ad}{d} + V_{th} | d < d_c) \tag{12}$$

Based on equation (4) and on the fact that ΔV_1 is a gaussian distribution with zero mean and standard deviation σ_1, we get

$$\mathbb{P}(V \geq V_{est} + V_{th} | d < d_c) = \mathbb{P}(\Delta V_1 \geq V_{th} - v_{m1} + a) = \int_{V_{th}-v_{m1}+a}^{+\infty} \frac{e^{\frac{-x^2}{2\sigma_1^2}}}{\sqrt{2\pi\sigma_1^2}} dx \tag{13}$$

Using the change of variable $y = \frac{x}{\sqrt{2}\sigma_1}$ and the expression of the "erfc" defined in equation (7), we have

$$\mathbb{P}(V \geq a + V_{th}) = \frac{1}{\sqrt{\pi}} \int_{\frac{V_{th}-v_{m1}+a}{\sqrt{2}\sigma_1}}^{+\infty} e^{-y^2} dy = \frac{1}{2} erfc(\frac{V_{th} - v_{m1} + a}{\sqrt{2}\sigma_1}) \tag{14}$$

In the other hand, based on equations (3) and (5), we get

$$\mathbb{P}(d < d_c) = \mathbb{P}(\frac{(\alpha + 1)N_0}{L} < d_c) = \mathbb{P}(\alpha < \frac{Ld_c}{N_0} - 1) = p_1 \tag{15}$$

Thereby, the first term of the expression (6) is proven.

2. *Congested traffic: Based on the Bayes theorem, the 2^{nd} term P^2_{sya} of the equation (9) can be expressed as*

$$P^2_{sya} = p_2 \mathbb{P}(V \geq V_{est} + V_{th} | d_c \leq d < d_{max}) \tag{16}$$

where, $p_2 = \mathbb{P}(d_c \leq d < d_{max})$.

Based on equation (1), for the congested state, the following statement holds

$$\mathbb{P}(V \geq V_{est} + V_{th} | d_c \leq d < d_{max}) = \mathbb{P}(V \geq \frac{bd + c}{d} + V_{th} | d_c \leq d < d_{max}) \tag{17}$$

In accordance with the congested traffic and by using equations (3)and (5), we can get

$$\mathbb{P}(V \geq V_{est} + V_{th} | d_c \leq d < d_{max}) = \mathbb{P}(V \geq V_{th} + \frac{cL}{(\alpha + 1)N_0} + b) \tag{18}$$

Likewise, based on equation (4) and on the fact that ΔV_2 *is a gaussian distribution with zero mean and standard deviation* σ_2*, we get*

$$\mathbb{P}(V \geq b + \frac{cL}{(\alpha + 1)N_0} + V_{th}) = \mathbb{P}(\Delta V_2 \geq V_{th} - v_{m2} + \frac{cL}{(\alpha + 1)N_0} + b) \tag{19}$$

Hence, we have

$$\mathbb{P}(V \geq b + \frac{cL}{(\alpha + 1)N_0} + V_{th}) = \int_{V_{th} - v_{m2} + \frac{cL}{(\alpha+1)N_0} + b}^{+\infty} \frac{e^{\frac{-x^2}{2\sigma_2^2}}}{\sqrt{2\pi\sigma_2^2}} dx. \tag{20}$$

Using the same change of variable as previously, we have trivially,

$$\mathbb{P}(V \geq b + \frac{cL}{(\alpha + 1)N_0} + V_{th}) = \frac{1}{\sqrt{\pi}} \int_{\frac{V_{th} - v_{m2} + b}{\sqrt{2}\sigma_2} + \beta}^{+\infty} e^{-y^2} dy, \tag{21}$$

where β *is defined in equation (8).*

From which it follows that

$$\mathbb{P}(V \geq b + \frac{cL}{(\alpha + 1)N_0} + V_{th}) = \frac{1}{2} erfc(\frac{V_{th} - v_{m2} + b}{\sqrt{2}\sigma_2} + \beta) \tag{22}$$

In the other hand, based on equations (3)and (5), we get

$$p_2 = \mathbb{P}(\frac{Ld_c}{N_0} - 1 \leq \alpha < \frac{Ld_{max}}{N_0} - 1). \tag{23}$$

Hence, the second term of the expression (6) is proven.

3. *Jam traffic: By applying the Bayes theorem, the* 3^{rd} *term* P^3_{sya} *of the equation (9) can be expressed as*

$$P^3_{sya} = \mathbb{P}(V \geq V_{est} + V_{th} | d \geq d_{max}) \mathbb{P}(d \geq d_{max}). \tag{24}$$

Since it is a traffic jam ($d \geq d_{max}$)*, the equation (1) gives*

$$\mathbb{P}(V \geq V_{est} + V_{th} | d \geq d_{max}) = \mathbb{P}(V \geq v_{th} | d \geq d_{max}). \tag{25}$$

By adopting the same approach as before, we can easily deduce the expression of p_3 *and the following expression*

$$\mathbb{P}(V \geq V_{est} + V_{th} | d \geq d_{max}) = \frac{1}{\sqrt{\pi}} \int_{\frac{V_{th} - v_{m3}}{\sqrt{2}\sigma_3}}^{+\infty} e^{-y^2} dy = \frac{1}{2} erfc(\frac{V_{th} - v_{m3}}{\sqrt{2}\sigma_3}), \tag{26}$$

This completes the proof.

Hence, The expression of the P_{sya} depicted in equation (6), confirms again that the three regimes shown in the FD diagram are taken into account in terms of sybil attack probability. In fact, under the fluid, congested or jam regimes, the probability of sybil attack varies. This is an expected result since the state of the road in terms of number of vehicles is among the most efficient indicators to decide if there is a sybil attack or not.

5.3. Performance evaluation of the proposed algorithm

To test the effectiveness of our sybil attack detection mechanism, we have performed some simulations based on the mathematical formulation given in the previous *Proposition 1.* These simulations are carried out using the parameters of the FD that will be detailed in the simulation results (section 6) as well as the parameters given in Table 1. The values given in this table could be divided into three parts:

- V_{th}, which has been chosen by simulation in order to enhance the detection performances.

Parameters	v_{th}	v_{m1}	v_{m2}	v_{m3}	σ_1^2	σ_2^2	σ_3^2
Values (m/s)	2	55	13.8	0.27	5.55	2.77	0.27

Table 1: Simulation parameters

- V_{m1}, V_{m2} and $_{V}m3$ correspond to logical maximum vehicle's speeds in the different traffic states (fluid, congested and saturated).
- σ_1^2, σ_2^2, and σ_3^2 correspond to some noises that are added on the speed to guarantee that the simulation is close to the reality.

In figures 5, 6 and 7, the P_{sya} is depicted versus N_0 ($10 \leq N_0 \leq 50$), α ($0.5 \leq \alpha \leq 5$) and V_{th} ($0 \leq V_{th} \leq 10$) respectively.

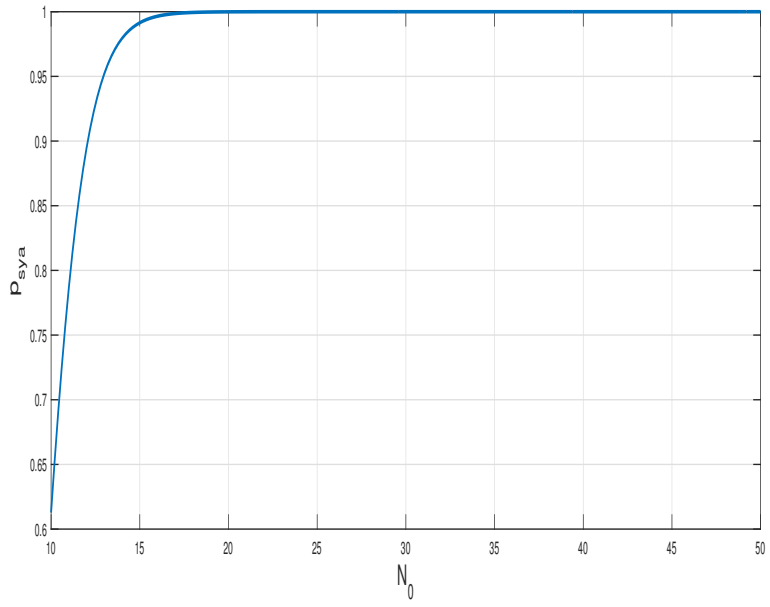


Figure 5: Sybil attack probability vs N_0

In the figure 5, we assume that we have 8 attackers (i.e. $\alpha = \frac{8}{N_0}$). This figure shows that the higher the number of the honest vehicles N_0 on the section, the higher the detection rate is. This can be explained by the fact that when the

number of the vehicles is low, the traffic is fluid and the few chosen number of attackers does not have an effect on the speed estimation (the vehicles moves at the free speed). Note here that, it can be possible to enhance this performance by choosing another form of the FD ([14]).

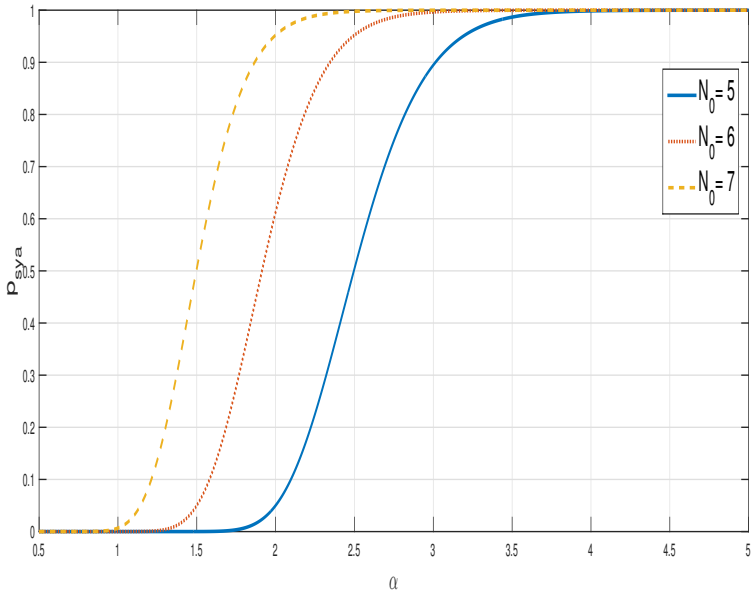


Figure 6: Sybil attack probability vs α

It is also interesting to note that if we increase the attacker number, the proposed algorithm can easily detect the Sybil attack as confirmed by Figure 6. In fact, a Sybil attack is generally launched when the traffic is fluid in order to simulate a fake congestion and consequently it needs a high number of attackers. Figure 6 shows clearly that the higher the number of attackers (α increasing), the higher the probability of detection is, even when the traffic is fluid. It shows also that the probability detection is higher when the number of honest nodes increases with the same number of attackers. This could be explained by the fact that the detection is easier in the congested area. Then, when the number of honest vehicles increases, adding attackers will change rapidly the density from fluid to congested.

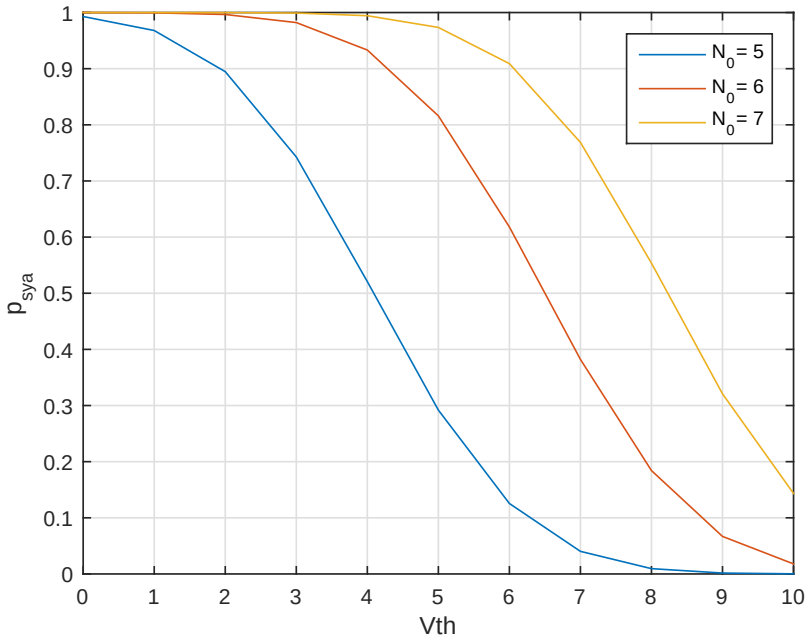


Figure 7: Sybil attack probability P_{sya} vs Vth ($\alpha = 3$)

The figure 7 shows how the selected speed threshold Vth impacts the detection probability P_{sya}. It can be noticed that when the threshold is low, the detection rate is high since we will identify all the attacks. However, this hides the false positive detection. However, when Vth increases, the detection probability P_{sya} decreases since we can miss some detection due to the high threshold. As a consequence, the speed threshold Vth has to be chosen properly in order to well detect and to avoid false positive detections. This issue will be studied in the next section using simulations.

6. Simulations and validation of the proposed algorithm

This section presents the environment and the results of the network simulation that we use to evaluate the efficiency of our proposed algorithm.

6.1. Simulations Environment

6.1.1. Simulations Tooling

The aim of these simulations is to target a realistic scenario while using standardized CAM messages. The used version of CAM messages corresponds to the ETSI EN 302 637-2 v1.3.2 updated on November 2014 [19] by the standardization organization European Telecommunications Standards Institute (ETSI) [20]. Therefore, the proposed algorithm was principally implemented over the application and the facilities layers into the ITSS architecture as standardized in the ETSI communication stack. For the development of our attack detection mechanism, we use the open-source simulation framework iTETRIS, which is a platform that integrates a network simulator and a traffic flow simulator. It is composed principally of four blocks:

- The application block: it contains all the applications that can be developed within the vehicular context.
- The network simulator block ns-3: it is a well known network simulator, which allows to test and validate communication protocols.
- The mobility simulator block SUMO: it is a well known mobility simulator, which allows to generate a realistic mobility model of the vehicles that will be used in the simulation.
- The controller block iTETRIS Control System (iCS): It coordinates the actions between the other three blocks. It is somehow the coordinating block of this platform. So, when an action is triggered by an application command, ns-3 will simulate a V2X transmission in the communication scenario within a wireless network. The results of this exchange are sent by iCS to the application which, as a result will produce a specific action that will be undertaken in the simulated road traffic scenario by SUMO (vehicle rerouting, emergency braking, etc.). The latter continually feeds others blocks with updated vehicle positions via iCS. As a central module

in the architecture, iCS facilitates the exchange of information between the different blocks [21].

6.1.2. Traffic scenario and attack simulation

In this subsection, we will detail the used mobility model for the evaluation of our Sybil attack detection mechanism. The first step consists in defining the FD characteristics. To do that, we used a mobility scenario in SUMO based on a 7 segments (denoted S_0 to S_6) in a circular road without any exit ramp. Then, vehicles are introduced one by one until reaching the traffic jam. Thus, we can pass through all the possible traffic phases. During this simulation, we integrate sensors (magnetic loops), which are located on the junctions between segments, to collect in a CSV file the needed data for the FD parameters identification (i.e. speed, density, occupancy rate, etc.).

Once these data are obtained, the parameters of the FD are identified using the procedure proposed in [14]. These parameters are given in the table 2.

Length	a	b	d_c(Veh/Km)	d_{max}(Veh/Km)	q_{max}(Veh/h)
S_6	108	-30,657	26,65	238,3	2160

Table 2: Parameters of fundamental diagrams

6.2. Simulations results

This section introduces the results of our simulations, where the used parameters are summarized in the table 3 in which we depict the range of the used values for each parameter when it is varying. Besides, it specifies its default chosen value when it is fixed. Each simulation was run 10 times and the presented results were averaged from these 10 executions.

6.2.1. False positive attack detection rate analysis

a) Impact of the segment length:

The figure 8 shows a comparison of the false positive detection rates for two different segment lengths ($600m$ and $1200m$), in terms of the speed threshold

Parameters	Default value	Variable values
Number of vehicles	200 vehs	100, 200 and 500 vehs
Segment length	1200m	600m and 1200m
Speed threshold	5 Km/h	1, 10, 20, 30, 40, 50,60, 70, 80, 90 and 100 Km/h
Number of attackers	15 vehs	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 22, 23, 24 and 25 vehs

Table 3: Simulation parameters

V_{th}. It represents the rate of the attack detection when no attack is launched. Therefore, it has to be minimized to avoid false alert warning.

From the figure 8, we can see that increasing the speed threshold minimizes the false positive detection rate. This is predictable from the mathematical model depicted in figure 7, since the lower speed threshold is, the higher probability to make a false attack detection, which is lower than $0,2\%$ with a speed threshold of 20 Km/h for both 600m and 1200m segment length.

The length of the segment has a weak impact on the false positive detection rate. In fact, the segment with a length of 600m has a higher false positive detection rate. This could be explained by the fact that the fundamental diagram is more accurate with longer segments which allows a better detection rate. Besides, the congestion is smoother in long segment since the traffic is less jerky, and therefore this reduces the false positive detections.

We can notice from the figure 8 that the false positive detection rate is relatively low. In all the cases, it is below 1%, which could be considered as enough powerful.

b) Impact of the number of vehicles:

The figure 9 presents the false positive attack detection rate for different

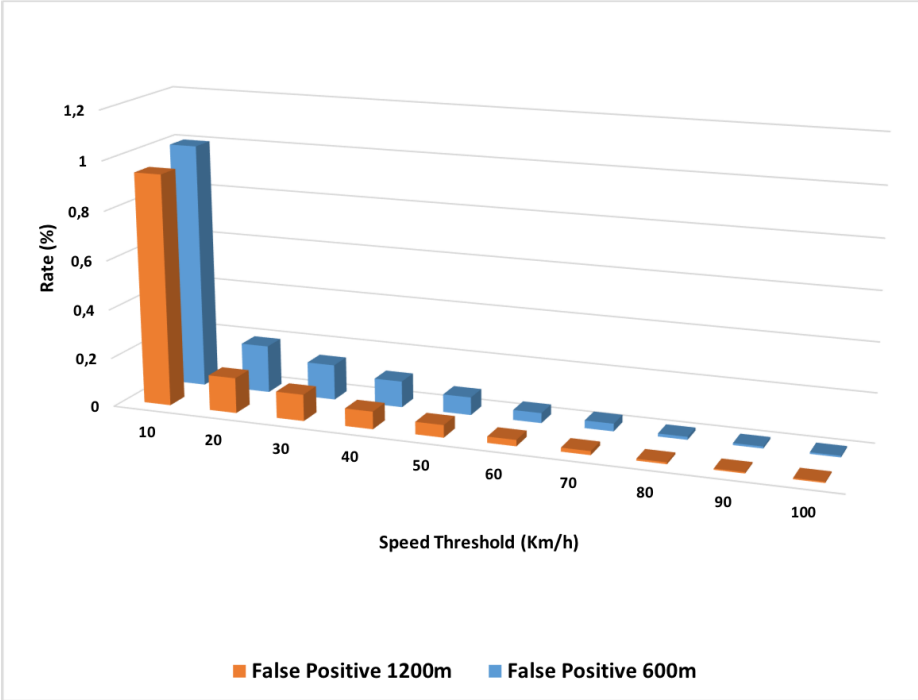


Figure 8: False positive attack detection rate Vs. the speed threshold

number of vehicles (100, 150 and 200) against the speed threshold.

In the figure 9, we can see that the false positive is reducing when the speed threshold is decreasing for the different number of vehicles. As stated previously, this is due to the fact that using a low threshold will make a lot of false detections.

When the number of vehicles increases, the positive false detection rate is reduced independently from the speed threshold (figure 9). This is understandable since the detection is more powerful in the congested region. Thus, when we insert more vehicles, we move from fluid area to the congested one, which allows to reduce the false positive detections.

6.2.2. Attacks detection rate analysis

a) Impact of the segment length:

Figures 10 and 11 show the correct (i.e. true positive) and the false negative

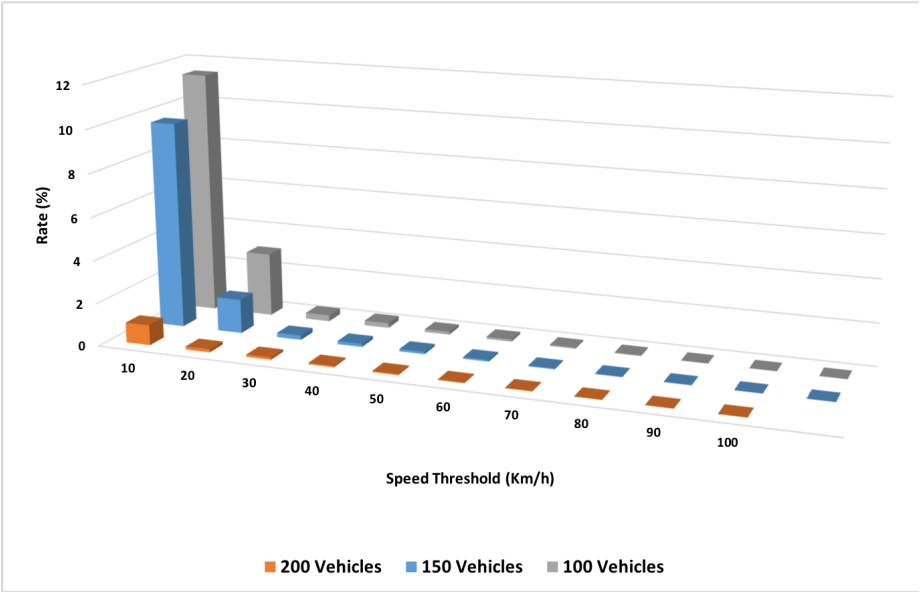
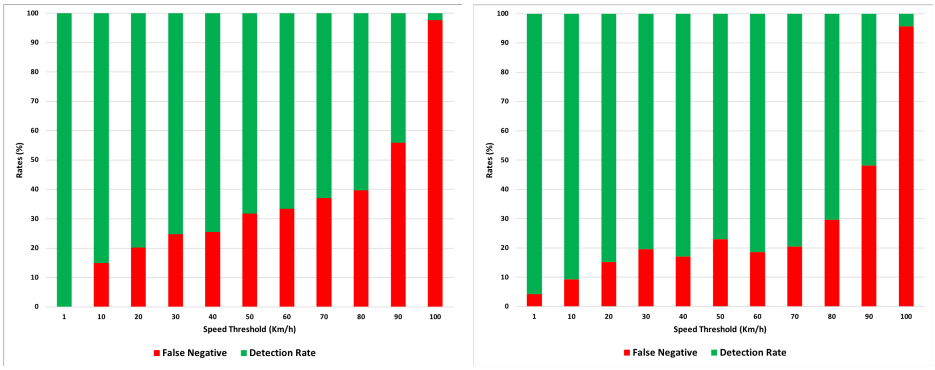


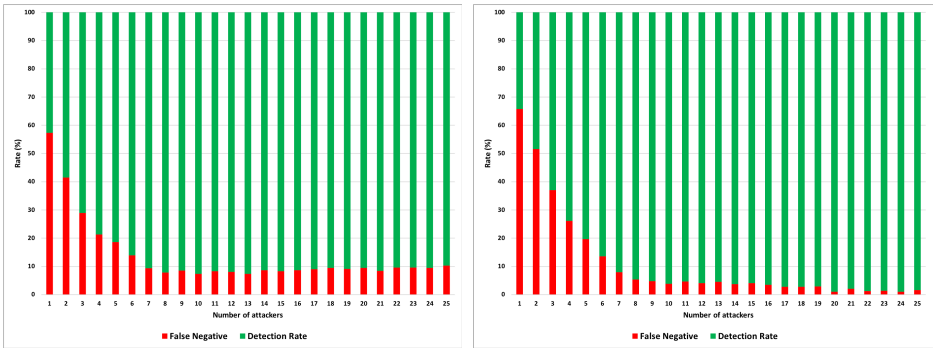
Figure 9: False positive attack rate Vs. the speed threshold for different number of vehicles



(a) 600m (b) 1200m

Figure 10: Attacks detection rate Vs. the speed threshold for different segment lengths

detection rates compared to the detection speed threshold and the number of the attackers, respectively, for two segments with a length of 600m and 1200m. A false negative detection means that the attack was not detected by the vehicle when it is in progress.



(a) 600m (b) 1200m

Figure 11: Attacks detection rate Vs. the number of attackers for different segment lengths

To study the impact of the used speed threshold, figures 10a and 10b present the correct detection rates and the false negative ones for two segments with a length of 600m and 1200m respectively. It can be noticed that, when the threshold is increased, the rate of false negative detection is higher. This could be explained by the fact that with a high speed threshold, we can miss a lot of attacks. For example in the figure 10a, with a threshold fixed to 30 Km/h, we had a rate around 25% of false negative detection, and with 80 Km/h, this rate increases to 40%.

In figures 11a and 11b, we fixed the speed threshold to 5 Km/h and we varied the number of attackers. With a low number of attackers, the detection is difficult since the difference between the estimated speed and the real one could not be very different, especially in the fluid zone, the speed remains the same (i.e. free speed) even with more vehicles. The difference will be more significant in the congested are. This is shown by the figure 11a, where, when we have more than 8 attacking vehicles, we will have almost the same detection rate, which is about 90% of correct detections.

From figures 10 and 11, one can remark that the correct detection rate is slightly higher for a longer segment. Since the segment is longer, the congested area will last more, than a shorter segment, before the total congestion

of the road, allowing more efficient attack detection. Moreover, a vehicle moving through a long segment will take more time, and then it will have more opportunities to detect the attack.

b) Impact of the number of vehicles:

Figures 12 and 13 show the correct and the false negative detection rates compared to the detection speed threshold and the number of attackers, respectively, using three different numbers of vehicles 100, 150 and 200 vehicles.

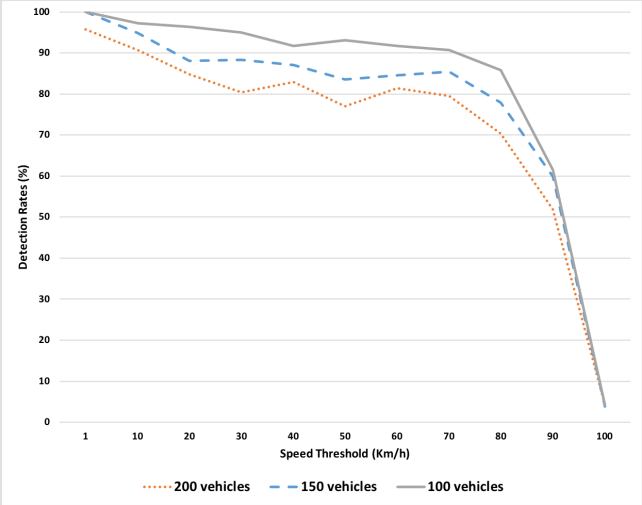


Figure 12: Attacks detection rate Vs. the speed threshold for different number of vehicles

The figure 12 shows that the detection rate is higher when the speed threshold is lower independently from the number of vehicles. As stated in the previous sub-section 6.2.2.a, this is due to the missing of attack detection when using high speed threshold. We can also notice that the detection rate, when varying the speed threshold, is impacted negatively by the number of the vehicles. The higher the number of vehicles, the lower the detection rate is. This is due to the weak number of attackers, which is fixed to 15 vehicles. Therefore, it is easier to identify the attack when the rate of attackers over all the vehicles is higher.

To be not redundant, we do not present here the results for false negative detection rates. However, they could be easily forecast.

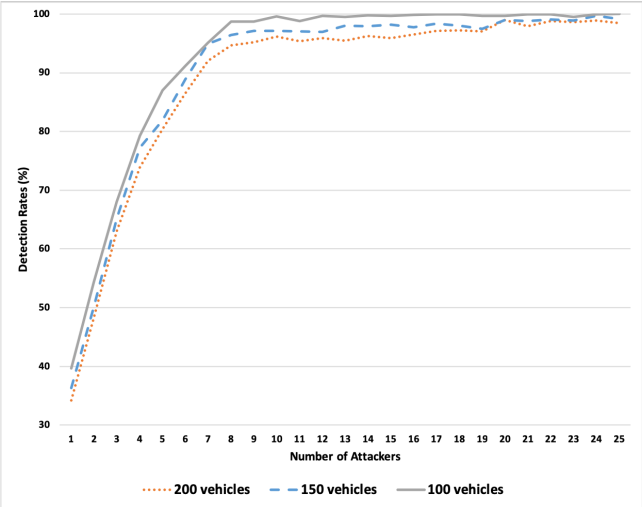


Figure 13: Attacks detection rate Vs. the number of attackers for different number of vehicles

The figure 13 shows that the detection is higher when the number of attackers increases independently from the number of vehicles. Unlike the speed threshold, the number of vehicles has almost no impact on the detection rate when varying the number of attackers since the number of used attackers remains insignificant compared to the number of involved vehicles.

Parameters	Best value	Comments
Number of vehicles	150 vehs	Compromise between false positive and true positive detection rates
Segment length	1200m	Best performances
Speed threshold	15 Km/h	Compromise between false positive and true positive detection rates
Number of attackers	10 vehs	Compromise between false positive and true positive detection rates

Table 4: Best parameters

As a conclusion to these simulations, which demonstrate that our proposed

model, when well tuned, has good performances (over 90% detection rate in general). Given the simulation parameters provided in Table 3, we discovered the "best" combination of parameter settings in Table 4, where "best" is defined as a compromise between the performances in terms of false positive, false negative and true positive detection rates. These parameters obviously depend on the fondamental diagram of the road's segment.

7. Conclusion

This paper presents a new Sybil attack detection mechanism for IoV. This mechanism considers each vehicle, as a Cooperative-Intelligent Transportation System Station, exchanging CAM messages. It takes advantage from a well-known macroscopic traffic flow models, that are supposed to be already provided to the vehicles. We first presented an algorithm that detects the Sybil attack using the CAM messages provided by neighbours, which estimates the speed of the vehicle using the fundamental diagram of the road's segment. If this estimated speed is too different from the real one, it detects an attack and broadcasts an alert to other nodes. Once the attack is detected, the trigger node waits for a confirmation from its neighbours in order to consider it as an attack and not a false detection one. This mechanism, which is easy to be implemented and very powerful, has been also validated through a mathematical model and some realistic simulations that prove its efficiency since it detects more than 90% of the attacks, when well tuned.

Our future works will deal with the identification of the attackers and the design of some countermeasures to fight against them.

8. Bibliography

[1] Al-Sultan S, Al-Doori MM, Al-Bayatti AH, Zedan H. A comprehensive survey on vehicular Ad Hoc network, J Netw Comput Appl 2014 Jan 31; 37: 380-92.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[3] J. Newsome, E. Shi, D. Song and A. Perrig, "The Sybil attack in sensor networks: analysis & defenses," Third International Symposium on Information Processing in Sensor Networks, 2004. IPSN 2004, Berkeley, CA, USA, 2004, pp. 259-268. doi: 10.1109/IPSN.2004.239019.

[4] M. T. Garip, P. H. Kim, P. Reiher and M. Gerla, "INTERLOC: An interference-aware RSSI-based localization and sybil attack detection mechanism for vehicular ad hoc networks," 2017 14th IEEE Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, 2017, pp. 1-6. doi: 10.1109/CCNC.2017.8013424.

[5] S. Han, D. Ban, W. Park and M. Gerla, "Localization of Sybil Nodes with Electro-Acoustic Positioning in VANETs," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6. doi: 10.1109/GLOCOM.2017.8253994.

[6] M. Khalil and M. A. Azer, "Sybil attack prevention through identity symmetric scheme in vehicular ad-hoc networks," 2018 Wireless Days (WD), Dubai, 2018, pp. 184-186. doi: 10.1109/WD.2018.8361717.

[7] Q. Tang and J. Wang, "A secure positioning algorithm against Sybil attack in wireless sensor networks based on number allocating," 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, 2017, pp. 932-936. doi: 10.1109/ICCT.2017.8359771.

[8] V. Gaikwad and L. Ragha, "Mitigation of attack on authenticating identities in ad-hoc network," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 1027-1032. doi: 10.1109/ICECDS.2017.8389593.

[9] S. Chang, Y. Qi, H. Zhu, J. Zhao and X. Shen, "Footprint: Detecting Sybil Attacks in Urban Vehicular Networks," in IEEE Transactions on Par-

allel and Distributed Systems, vol. 23, no. 6, pp. 1103-1114, June 2012. doi: 10.1109/TPDS.2011.263.

[10] P. Gu, R. Khatoun, Y. Begriche and A. Serhrouchni, "k-Nearest Neighbours classification based Sybil attack detection in Vehicular networks," 2017 Third International Conference on Mobile and Secure Services (MobiSecServ), Miami Beach, FL, 2017, pp. 1-6. doi: 10.1109/MOBISECSERV.2017.7886565.

[11] Raksha Tiwari, Tripti Saxena. A Review on Sybil and Sinkhole of Service Attack in VANET. Recent Trends in Electronics & Communication Systems. 2018; 5(1): 7–11p.

[12] Study on the Deployment of C-ITS in Europe: Final Report, Website available at: https://ec.europa.eu/transport/sites/transport/files/2016-c-its-deployment-study-final-report.pdf.

[13] M. T. Garip, P. Reiher and M. Gerla, "Ghost: Concealing vehicular botnet communication in the VANET control channel," 2016 International Wireless Communications and Mobile Computing Conference (IWCMC), Paphos, 2016, pp. 1-6. doi: 10.1109/IWCMC.2016.7577024.

[14] A. Zeroual, N. Messai, S. Kechida, F. Hamdi, A Piecewise switched linear approach for traffic flow modeling, International Journal of Automation and Computing, Vol. 14, pp. 729-741, 2017.

[15] S. Boubaker, F. Rehimi, and A. Kalboussi, "Comparative analysis of microscopic models of road traffic data," in 2011 4th International Conference on Logistics, 2011, pp. 474–478.

[16] Jayakrishnan R., Mahmassani H.S., Hu T.Y., 1994, An evaluation tool for advanced traffic information and management in urban networks, Transportation Research C, no. 2, pp. 129-147.

[17] M. J. Lighthill and G. B. Whitham, "On Kinematic Waves. II. A Theory of Traffic Flow on Long Crowded Roads," Proc. R. Soc. A Math. Phys. Eng. Sci., vol. 229, no. 1178, pp. 317–345, May 1955.

[18] H. J. Payne, Models of Freeway Traffic and Control. Simulation Councils, Incorporat, 1971.

[19] Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. ETSI EN 302 637-2 v.1.3.2 (2014-11).

[20] European Telecommunications Standards Institute (ETSI), Available at: http://www.etsi.org.

[21] http://www.ict-itetris.eu/ (access October, 21st 2018)