A Novel Sybil Attack Detection Mechanism for C-ITS

Marwane Ayaida, Nadhir Messai, Geoffrey Wilhelm
CReSTIC,
Campus Moulin de La Housse
Université de Reims Champagne-Ardenne
Reims, France
{marwane.ayaida, nadhir.messai, geoffrey.wilhelm}@univ-reims.fr

Sameh Najeh
University of Carthage,
Higher School of Communications of Tunis,
COSIM Research Lab.
Tunis, Tunisia
sameh.najeh@supcom.tn

***Abstract*—Cooperative Intelligent Transport Systems (C-ITS) are expected to play an important role in our lives. They will improve the traffic safety and bring about 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 attacks is the Sybil attack, which is even more dangerous than others because it could be the starting point of many other attacks in C-ITS. This paper proposes a distributed approach allowing the detection of Sybil attacks by 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 a comparison between the real accurate speed of the vehicle and the one estimated using the V2V communications with vehicles in the vicinity. The estimated speed is derived by using the traffic flow fundamental diagram of the road's portion where the vehicles are moving. This detection algorithm is validated through some extensive simulations conducted using the well-known NS3 network simulator with SUMO traffic simulator.**

***Index Terms*—ITS, C-ITS, Traffic Model, Sybil attack.**

I. Introduction

The new mobility challenges of vehicles in Smart Cities need the enhancement of Intelligent Transportation Systems (ITS) that helps to reduce congestions, accidents, fuel consumption, etc. Thus, Cooperative Intelligent Transport Systems (C-ITS) , which are a major component of ITS, has 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 other 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 attack, which is considered as the most dangerous one 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 a crash in an accident. Therefore, detecting such attack is very sensitive for several safety, privacy and security reasons.

Many mechanisms that aim to detect Sybil attacks have been proposed in the literature. Among them, we can cite those based on resource testing [3] (i.e. computing ability, storage ability, communication bandwidth, etc.). The idea here is that each vehicle broadcasts to all its neighbors a request that 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 interval time 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.

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. Traditional PKI-based certificates include only key information and do not include any unique physical information related to the vehicle. This makes such approach potentially vulnerable to impersonation attack because any stolen valid key pair and it could be used by malicious vehicle to create fake nodes with valid certificates. In multi-factor authentication scheme [4], the certificate contains not only the public key information but also a set of physical attribute values about the vehicle (i.e. the radio frequency fingerprint, etc.) recorded by the Certificate Authority (CA). Nevertheless, establishing such Public Key Infrastructure for individual

978-1-5386-7747-6/19/$31.00

vehicles [5] takes a long time. The use of a long-term key pairs and certificates can also make the tracking and the collecting of vehicles behaviors easier. PKI-based approaches are complex and expensive to be implemented in terms of equipments that have to be deployed. For example, we have to deploy a Root Authority (RA), a Long-Term Certificate Authority (LTCA) and a Pseudonym Certificate Authority (PCA), which 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 2026 with a cost of 660 M€[6]. Another alternative is to take advantage from 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 [7]. Authors in [8] 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, but 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 strengthen and its arrival angle. Then, this can be used to detect that the received signal could not be the received one from a declared position in case of Sybil attack. However, this technique needs to use many neighbours, and then a high density, to localize well a node. [9] proposes a new way, to detect Sybil attacks, based on electro-acoustic positioning. Simulations showed that the electro-acoustic positioning outperforms RSSI-based positioning. Despite its efficiency, electro-acoustic positioning suffers from a major drawback, since 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.

The authors in [10], have 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 a open way.

Authors of [11] suggested another secure positioning distributed algorithm, which is based on number allocating and mutual guarantee relying on neighbours for Wireless Sensor Networks (WSN). This latter uses a lot of messages in order to guarantee the right location of nodes, which increases the cost of such algorithm. [12] defined a protocol allowing the detection and the prevention from Sybil attacks in Mobile Ad-hoc Networks (MANETs), based principally on clustering and path similarity. This protocol used the packet authentication, based on the electronic signature and on the private keys. The management of such keys is very challenging especially in a mobile networks. For more Sybil attack detection related works, readers are encouraged to read the survey presented in [13]. To tackle some limitations of the overviewed approach, this paper proposed to design an original Sybil attacks detection mechanism, which takes benefit from the traffic flow models already provided to the vehicle in order to detect Sybil attacks. 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 speed that this vehicle estimated in a distributed way by using the information of its surrounding.

The reminder of this paper is as follow. Section II introduces the system model and the problem formulation of the both targeted scenario and used traffic flow model. Section III details how our detection algorithm works. Section IV evaluates this algorithm using a realistic network simulation. Finally, section V concludes this paper.

II. System Model and Problem Formulation

During the two last decades, three classes of traffic flow models have been developed [14] . The microscopic models that consider each vehicle individually constitute the first class. The second one contains the mesoscopic models which consider packets of vehicles that have the same destination. First-order and second-order macroscopic models constitute the third class. Macroscopic models consider the traffic as a compressible fluid that circulates on the p links $(L_j)_{1\leq j\leq p}$ between the q nodes $(N_i)_{1\leq i\leq q}$ of the network. The first-order approach assumes that traffic is described, for each link $(L_j)_{1\leq j\leq p}$, in terms of speed v_j, flow q_j and density d_j, according the 3 following equations:

$$\frac{\partial q_j(x,t)}{\partial x} + \frac{\partial q_j(x,t)}{\partial t} = 0,\ j \in \{1...p\} \qquad (1)$$

$$q_j(x,t) = d_j(x,t).v_j(x,t),\ j \in \{1...p\} \qquad (2)$$

$$q_j = f(d_j(x,t)),\ j \in \{1...p\}, \qquad (3)$$

where, the variable x stands for the spatial measure on the considered link, and t stands for the time.

The flow-density relationship (3) often named fundamental diagram (FD) has the following properties: *zero rate of flow is encountered at zero density and again at maximum of queue density*. Each rate of flow corresponds to two densities and also two speeds. The lower speed is corresponding to the higher density and the higher speed is corresponding to the lower density. The lower density (or higher speed) refers to non-congested flow, and the higher density (or lower speed) refers to congested flow.

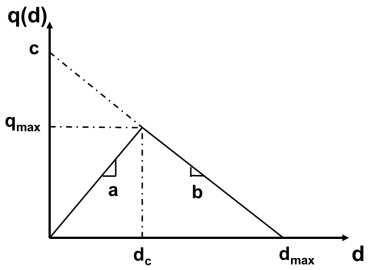

Fig. 1: Fundamental diagram example

Practically, each fundamental diagram is analytically defined by a set of parameters like the free speed which is the only common parameter of all existing models, as shown if figure 1. Whereas, the interpretation of the other parameters changes from one model to another. Based on the relation (2), the slope of the line connecting the origin to any point of the FD gives the speed corresponding to that point.

According to Figure 1, the FD can be characterized by the following equations

$$\begin{cases} \text{if } d < d_c, \text{ then } q = ad \\ \text{if } d_c \leq d < d_{max}, \text{ then } q = bd + c \\ \text{if } d \geq d_{max}, \text{ then } q = 0, \end{cases} \quad (4)$$

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

III. Presentation of the proposed algorithm

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 received message. Otherwise, it adds the new sender of the message as a neighbor and it inserts its location and its timestamp.

Algorithm 1 Updating Neighbors List

```
1: procedure NeighborsUpdating(Packet P, List Neighbors)
2:     if (P.Sender ∈ Neighbors) then
3:         Neighbors[Sender].Timestamp ← P.Timestamp;
4:         Neighbors[Sender].Location ← P.Location;
5:     else
6:         Neighbors.Add(Sender);
7:         Neighbors[Sender].Timestamp ← P.Timestamp;
8:         Neighbors[Sender].Location ← P.Location;
9:     end if
10: end procedure
```

Algorithm 2 Computing Number of Neighbors

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

To better understand the proposed algorithm, we present here its most features depicted in the 4 following algorithms: 1) Algorithm 1 details the procedure of the neighbors' list updating, 2) Algorithm 2 presents the procedure of the neighbors' number computing, 3) Algorithm 3 shows the procedure that uses the traffic model to estimate the speed of the vehicle and 4) Algorithm 4 describes the whole algorithm that uses the three previous algorithms. When a vehicle needs to compute the number of its neighbors, it starts by updating the list of its neighbors as detailed in the Algorithm 2. To do that, it verifies if the time since the receiving of the last message is higher than a given threshold T_{th}. If so, it removes the neighbour from the list. The number of neighbours is then calculated as the cardinality of this updated list. 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. 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 neighbors 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 virtual 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 of the traffic model is estimated using the equation (2). The Algorithm 4 presents the Sybil attack detection procedure. When the vehicle receives a CAM messages, it updates the list of neighbors 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 and the estimated one is computed and compared with a predefined threshold V_{th}, which depends essentially on the road and the traffic model. It could be given also as an input with the FD.

Algorithm 3 Estimating Speed

procedure SPEEDESTIMATION(List Neighbors)
  double V_{est} : estimated speed
  double a : constant of fluid area
  double b : constant of congested area
  double d_c : critical density
  double d_{max} : maximal density
  double $c = -b * d_{max}$: second constant of congested area
  double density : current density
  int Length : length of a segment
  density = (NumberNeighbors(Neighbors)+ 1) / Length;
  if ($density < d_c$) **then** ▷ Fluid Area
    $flow = a * density$;
  else if ($density < d_{max}$) **then** ▷ Congested Area
    $flow = b * density + c$;
  else
    $flow = 0$; ▷ Traffic Jam
  end if
  $Vest = flow/density$;
  $return\ V_{est}$;
end procedure

If this difference 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.

IV. SIMULATIONS AND VALIDATION OF THE PROPOSED ALGORITHM

This section presents the environment and the results of the network simulation, which are used to evaluate the efficiency of our proposed algorithm.

Algorithm 4 Attack Detection

double V_{est} : estimated speed
double V : real speed
double V_{th} : threshold to detect a Sybil attack
List Neighbors : list of Neighbors
Time T_{det} : timer of periodic attack detection triggering
Time DetectionTime : timestamp of the attack detection
int Timeout : maximum waiting time for an attack confirmation
Packet CAM : packet CAM received
while ($ReceiveCAM(CAM)$) **do**
  $NeighborsUpdating(CAM, Neighbors)$;
end while
if (T_{det} is expired) **then**
  $V_{est} = SpeedEstimation(Neighbors)$;
  $V = Node.Mobility.GetSpeed()$;
  if ($|V - V_{est}| > V_{th}$) **then**
    $DetectionTime = Now$;
    $BrodcastMessage(AttackDetected)$;;
    $Wait(Timeout)$;
    if ($ReceiveConfirmation()$) **then**
      $LaunchCountermeasures(Sybil)$;
    end if
  end if
  T_{det} is armed
end if

A. Simulations Environment

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 by the standardization organization European Telecommunications Standards Institute (ETSI). 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 NS3 and a traffic flow simulator SUMO.

The traffic model is based on the road traffic characteristics (i.e. the traffic type and the traveling vehicles movement). First, the FD characteristics presented in the Figure 2 are defined, using a mobility scenario in SUMO based, on a 7 segments (denoted S_0 to S_6) in a circular road without any exit ramp.

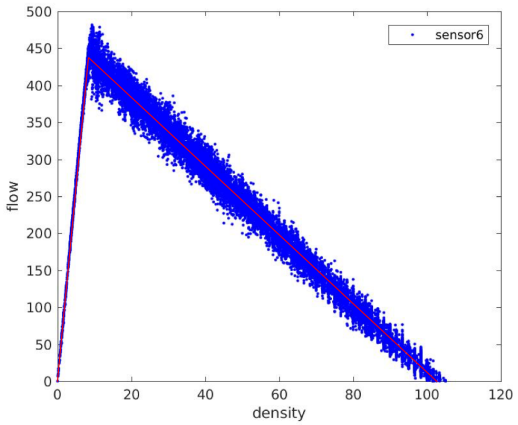

Fig. 2: Fundamental diagram

Second, 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 data needed for the FD parameter identification (speed, density, occupancy rate, etc.) in a CSV file. Once these data are obtained, the parameters of the FD are identified using the procedure proposed in [14]. These parameters are given in Table I.

	a	b	d_c (Veh/Km)	d_{max} (Veh/Km)	q_{max} (Veh/H)
S_6	54.02	-4.77	8.202	101.03	443.1375

TABLE I: Parameters of fundamental diagrams

Attackers in a Sybil attack are some specific nodes that generates identifiers that do not physically exist in the network. Therefore, they send CAM messages for they real identities and for the fake nodes that are generated. To implement such a scenario, we create a number of mobile nodes that play the role of an attacker.

B. Simulations results

This section introduces the results of our simulations. The parameters used in these simulations are summarised in the table II. Each simulation was run 10 times and the presented results were averaged from these 10 executions. The figures 3 and 4 show the right and false negative detection rates compared to the detection speed threshold and the number of the attackers respectively.

Parameters	Default value	Variable values
Number of vehicles	200	
Segment length (m)	600	
Speed threshold (Km/h)	5	1, 10, 20, 30, 40, 50,60, 70, 80, 90 and 100
Number of attackers	15	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
Number of executions	10	

TABLE II: Simulation parameters

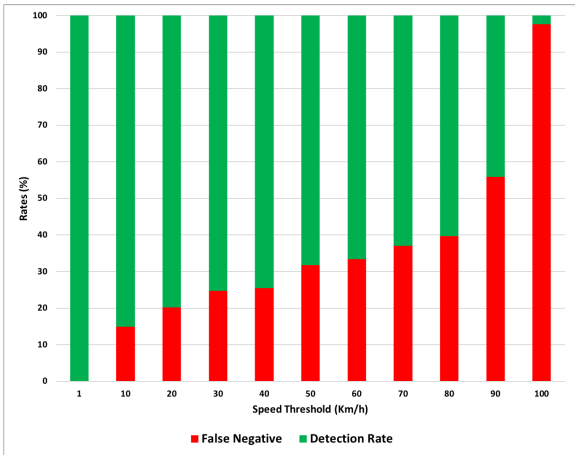

Fig. 3: Attacks detection rate vs the speed threshold

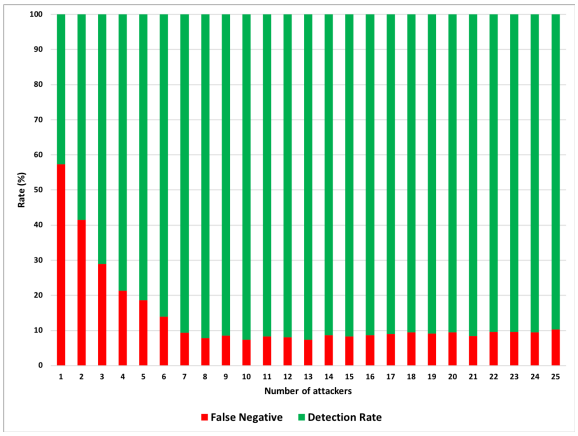

Fig. 4: Attacks detection rate vs the number of attackers

To study the impact of the used threshold, the figure 3 presents the detection rate and the false negative one. It can be noticed 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,

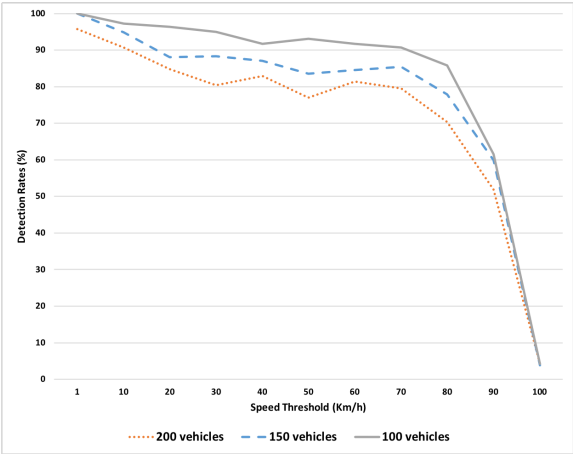

Fig. 5: Attacks detection rate Vs. the speed threshold for different number of vehicles

with a threshold fixed to 30 Km/h, we had around 25% of false negative detection rate and with 80 Km/h it increases to 40%. In the figure 4, 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, specially in the fluid zone, the speed remains the same even with more vehicles. The difference will be more significant in the congested area. This is shown by the figure 4, where we have more than 8 vehicles, we will have almost the same detection rate, which is about 90% of right detection. Figure 5 shows that the detection rate is higher when the speed threshold is lower independently from the number of vehicles. 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.

V. Conclusion

This paper presents a new Sybil attack detection mechanism for C-ITS. We first presented an algorithm that detects the Sybil attack using the CAM messages provided by neighbours. This algorithm allows to estimate 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. Second, 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. Finally, the proposed mechanism, which is easy to be implemented and very powerful, has been also validated through some extensive realistic simulations, where it detects more than 90% of the attacks. In terms of future works, the identification of the attackers and the design of some countermeasures to fight against them, can be investigated.

VI. Acknowledgments

This work was made possible by the C-Roads France project funded by the European Commission Grant No. 2015-FR-TM-0378-S from the INEA Agency within the 2015 CEF Transport Programme. The statements made herein are solely the responsibility of the authors.

References

[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] S. Pal, AK. Mukhopadhyay and PP. Bhattacharya, Defending Mechanisms Against Sybil Attack in Next Generation Mobile Ad Hoc Networks, IETE Technical Review, vol 25, no 4, pp. 209-214, 2008 .

[5] M. Raya and J.P Hubaux. 2007. Securing vehicular ad hoc networks. J. Comput. Secur. 15, 1 (January 2007), 39-68

[6] 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.

[7] 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.

[8] 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.

[9] 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

[10] 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.

[11] 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.

[12] 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.

[13] R. Tiwari, T. Saxena, "A Review on Sybil and Sinkhole of Service Attack in VANET", Recent Trends in Electronics & Communication Systems. 2018; 5(1): pp. 711.

[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.