Detection of Malicious Vehicles with Demerit and Reward Level System

Jihene Rezgui[1], Cédryk Doucet[2]
[1 2]*Laboratoire Recherche Informatique Maisonneuve (LRIMa), Montreal, Canada*
jrezgui@cmaionneuve.qc.ca

***Abstract*— In Vehicular Ad hoc Networks (VANETs), to ensure the reliability of safety applications, each vehicle periodically receives crucial information from neighboring vehicles, in its transmission range, about their respective current state including position, direction, etc. Some vehicles will want to inject false information on the network, whether accidentally or intentionally. These vehicles, that we will call malicious vehicles, may affect the timeliness and the pertinence of safety related data. Our proposed scheme called *TrustLevel*, ranks a vehicle sharing inaccurate information repeatedly as malicious. To ensure system reliability, the *TrustLevel* scheme is able to promptly detect the malicious vehicles and ensure the accuracy of the vehicle perception of its environment. To do this, *TrustLevel* collects data for each vehicle on its environment. Thereafter, it creates a perception map for each vehicle representing its surroundings. *TrustLevel* can then cross these maps in order to increase each vehicle's perception. When doing so, the information sent by malicious vehicles will be incoherent, so *TrustLevel* will be able to detect these inconsistencies and isolate the faulty vehicles. The isolated vehicles will then be ignored by the other vehicles when exchanging their perceptions, increasing the system reliability. Simulation results demonstrate the efficiency of the *TrustLevel* scheme under high density of malicious vehicles.**

Keywords*— *VANET, Malicious vehicle, active safety, global perception accuracy, routine messages, TrustLevel;

I. INTRODUCTION

Having a reliable system is a necessity in vehicular networks because its reliability ensures the safety of drivers. The reliability depends on the information sent and received by each vehicle via routine messages. In VANETs each vehicle operates as wireless entity that communicates with each other in a range of 200-300 meters (V2V communication). The purpose of VANETs is an overwhelming increase of road networks safety. The protocol used is IEEE 802.11p, which uses the frequency band DSRC (Dedicated Short Range Communication). It is performed by way of routine heartbeat safety messages sent every 10ms as per DSRC standard [1].

The vehicles communicate with each other by sending messages. There are two types: safety and entertainment. Messages on which we focus in this paper are safety messages. They are composed of alert and routine messages. Alert messages warn the driver of any danger nearby. The routine messages, allow us to know the network status at a specific time.

Identifying misbehaving vehicles plays an important role in road safety. A vehicle can be defined as malicious if it sends false information about its position, direction, etc. or if the speed of the vehicle suddenly changes or the frequency of sudden braking exceeds a predefined threshold value. Such malicious vehicles have to be isolated and should not be allowed to participate in the further building of the network perception.

Our contributions in this paper can be summarized as follows:

(i) we propose a scheme called ***TrustLevel***, to establish demerit and reward level system that takes into account the relative severity of each violation of the Highway Safety Code to construct initial decision Tree,

(ii) we provide a fast detection of all malicious vehicles sending false information by crossing of perception maps to highlight incoherence between local perceptions [1], and

(iii) we achieve a better detection accuracy of invalid perception with the ***TrustLevel*** scheme, which computes the accuracy of perception in the area on ignored and non-ignored malicious vehicles' perception.

The novelty, as far as one can tell from the literature, no studies have addressed the problem of extracting data in VANETs and mining routine messages to derive incoherence regarding vehicles' perception maps.

The organization of paper is as follows: Section 2 briefly describes related works on the detection of malicious vehicles. Section 3 describes in details the proposed algorithms to accomplish our goals. Section 4 presents a performance study of our proposed scheme ***TrustLevel*** and the paper concludes in Section 5.

II. RELATED WORK

To the best of our knowledge, this work is the first to detect malicious vehicles behaviour to set up an accurate extended perception in a vehicular network environment. For detecting malicious vehicles in VANETs, only few studies [2] to [6] have been done. In [3], the authors proposed a model-free

approach for detecting anomalies in un-manned autonomous vehicles. Their approach is based on the sensor readings. In [4], authors proposed the use of mining to detect malicious vehicles giving false information. For that, they developed a mechanism called VANETs Association Rules Mining (*VARM*) that collects, on a single vehicle, data regarding each neighbour transmission and extracts the temporal correlation rules between vehicles implicated in transmissions in the neighbourhood. With *VARM*, a mining process will take place during a-priori constant historical period to gather information and to detect malicious vehicles. In order to limit the number of the reported rules in their scheme, the authors use the mathematical foundations of the Formal Concept Analysis (FCA), which provided a battery of results, known as generic bases of association rules as shown in [4].

The authors in [5] addressed the problem of preventing the data packets to move to the malicious nodes in the network with the help of *bacterial foraging optimization (BFO)* algorithm. If the sender wants to send data to a destination, it will check its distance. If there is no direct path, then it will find the path from coverage set and after that, transmission of data packet is checked. It there is loss in packets, then *BFO* algorithm is used. The scheme proposed in [6] is able to control the traffic by maintaining the distance between the vehicles. The malicious vehicles are isolated and further communication requiring them is stopped. The existing *Ad-hoc On Demand Distance Vector (AODV)* protocol has been suitably modified to achieve the above mentioned road safety measures and has been derived as *Robust AODV protocol (RAODV)*.

However, many researchers [7] to [12] have recently studied security issues to detect malicious VANETs. The authors in [7] proposed a guard node based scheme that detects malicious nodes. Each node calculates the trust level of its neighbours based on the opinions of other nodes. If the trust value of a node is lower than a certain value, the node is identified as malicious and is isolated. The scheme has been evaluated for three different types of malicious attacks: impersonation attacks, colliding nodes attacks and black hole attacks. In [8], the authors proposed a "light-weight" and scalable framework to detect malicious behaviors. The advantage of their proposed scheme is to not require any vehicle in the network to reveal its identity. Studies in [9] to [12] proposed to preload each vehicle, during vehicle registration, with a pool of pseudonyms generated by some government entity. The pseudonyms are used to hide a vehicle's unique identifier. When a vehicle needs to report an event, it randomly picks one pseudonym and signs the message with it, using public key cryptography. This makes it easy to verify the identity of a malicious vehicle.

However, our work aims to guarantee road safety and, secondly, the network security while building an accurate extended perception while bypassing the participation of malicious vehicles.

III. PROPOSED APPROACH AND ALGORITHMS

Our proposed scheme is comprised of the following five major steps. Throughout this paper, we will consider the notations shown in Table I.

Table 1. LIST OF SYMBOLS/PARAMETERS

neighborsList (v)	Set of vehicles in the range of vehicle v
∂	Vehicle
tree	Decision tree
x	Number of sudden brakings of a vehicle
y	Number of sudden accelerations of a vehicle
z	Total number of kilometres travelled in a vehicle (odometer)
P(∂)	Probability (score) of vehicle ∂
β	Infraction weight coefficient
k	Integer representing the number of offenses committed
P(ε)	Infraction penalty
P(α)	Reward gain
t	Time spent since last penalty (ms)

Step 1: Let's start with the points-level system we call ***TrustLevel***. According to ***TrustLevel***, every driver's skills are initially based on what follows:

- x : Number of sudden brakings
- y : Number of sudden accelerations
- z : Total number of kilometres travelled

A braking is considered sudden when the vehicle decelerates at a rate of at least 4.16 m/s^2 (15km/h in 1 second). An acceleration is considered sudden if there is a speed gain of at least 4.16 m/s^2. This data will be used as a baseline. It will help us determine the initial trust level of each driver. β is determined through extended simulations.

1. Initial Points Attribution Algorithm (AIP)

$$\forall \partial \in V,\ P(\partial) \leftarrow 1 - \beta\left(\frac{x+y}{z}\right)$$
$$0 < \beta < 1$$

Step 2: Then, the data structure chosen is a decision tree built according to each vehicle's communication range. Here is

an example of a road network as well as its matching decision tree *(see Figure 2)*:

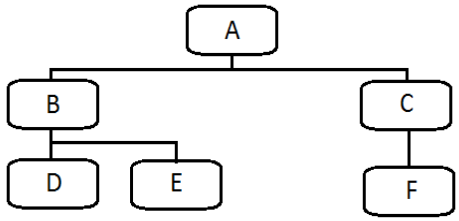

Figure 2. Decision tree example matching the Figure.3

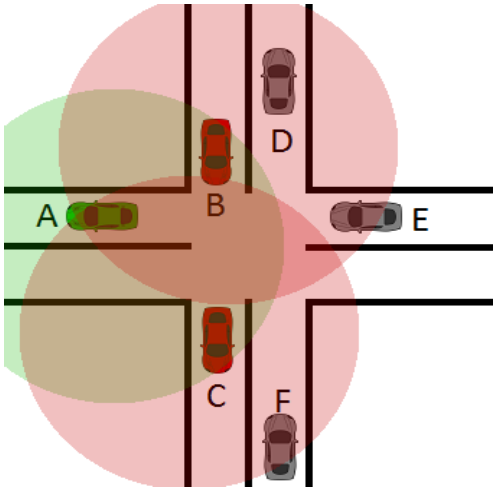

Figure 3. Road network example with range shown

We can interpret the Figure 2 as following: vehicles B and C are within vehicle A's range. Vehicles D and E are within vehicle B's range. Also, vehicle F is within vehicle C's range. It is important to note that when we get to the node B, it is possible to observe that the vehicle A is also within its range. The vehicle A will not be added as node B's child, because there is already a node in the tree with the label A (see TBA algorithm).

2. Tree Building Algorithm (TBA)

1. tree.root ← ∂
2. ∀vehicle *i* ∈ *neighborsList of* ∂
   2.1 **if** *i* is not in *tree*
   tree.addChildren(*i*)
   tree.childrenCount ++
   2.2 **if** *i* is in *tree*
   Algorithm 3 *(IDA)*
   tree.updateInfo*(i)*
3. ∀ children *j* ∈ *childrenList, redo* 1

Step 3: To continue, the incoherence detection on the road network will be done as follows: every vehicle knows the information relative to its environment. In other words, any node in the tree (Fig. 2) will have the list of vehicles directly in its range.

Vehicles B and C perceive vehicle A at different locations, according to an augmented perception (see ref [1]). The intersection between the perception maps allows us to find vehicle A. This means that the vehicle shares a false position with the other entities on the network.

Explained differently, the vehicles B and C receive one-time messages (beacons) from vehicles in their range. B will receive a message from vehicle A telling its position as well as the list of adjacent vehicles. C will also receive information from A, but with different information. There is incoherence between the messages transmitted by A, so, the TLM algorithm will be called with A.

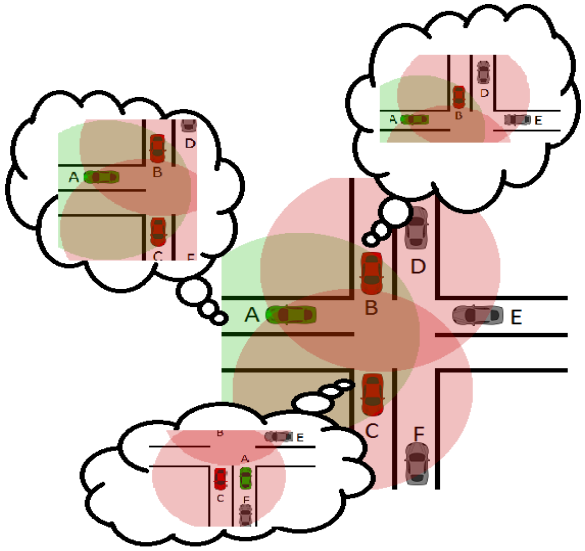

Figure 4. Incoherence in perceived information example

3. Incoherence Detection Algorithm (IDA)

1. Perception maps building (ref [1])
2. Algorithm 2 (TBA), line 2.2
3. **if** *tree*(∂).info ≠ ∂.info
   - ∂ sends incoherent information.

On line 3 of IDA, it is possible to say, as an example, that B's tree already contains information relative to A's position because of the first beacon received. B receives a new beacon from C and needs to update its tree. B detects incoherencies with A's node while updating A's information sent by node C.

Step 4: When a false message is detected, the malicious vehicle's score will diminish. A vehicle will be able to regain its lost points. It works in a way like the attribution of demerit points on Quebec's driver licenses for example. A driver that has committed an infraction will have demerit points added on its license. The amount of points added depends on the infraction's severity.

The idea is the same with ***TrustLevel***. The amount of infractions committed behaves similarly in relation to the

severity notion. The more a vehicle transmits a big number of false messages, the more it will be penalized on its next false message.

Also, concerning the driver licenses, it is possible to erase the demerit points obtained if the driver does not commit an infraction for 2 years (e.g. in Quebec, Canada). In our case, if the vehicle did not send any malicious messages for a predefined duration, it will regain a part of its points.

So, every vehicle has a meter *k* representing the count of sent malicious messages.

Step 5: The classification of malicious vehicles is done as follows: as seen in *TLM* algorithm, when the vehicle's trust level is between 0 and 0.3, it means that the vehicle has sent false information multiple times. At this moment, it is possible to classify it as a malicious vehicle. The messages sent by the malicious vehicle will then not be considered in the extended perception.

4. Trust Level Modification Algorithm *(TLM)*

1. $k \leftarrow 0$ // the count of sent malicious messages
2. **if** ∂ sends a false message ($k \leftarrow k + 1$)
   - $P(\varepsilon) \leftarrow \frac{kP(\partial)}{k+1}$ // Infraction penalty
   - $P(\partial) \leftarrow P(\partial) - P(\varepsilon)$ // currently score
   - **if** *P(∂)* is between 0 and 0.3
     - ∂ is malicious
3. **else if** ∂ is not malicious $\wedge$ ((t >= t_{max}) $\vee$ (t is null)) /* t_{max} is a predefined duration to erase the demerit points */
   - $P(\alpha) \leftarrow \frac{1-P(\partial)}{2}$ // reward gain
   - $P(\partial) \leftarrow P(\partial) + P(\alpha)$// currently score

IV. RESULTS ANALYSIS

A. Initialization of our Simulations

The route is initialized as follows: it is a district comprising of a total of 21 intersections. The road network is arranged in a typical quadrilateral way, meaning most roads are perpendicular to each other. Here is an image of the configuration (*Figure 5)*.

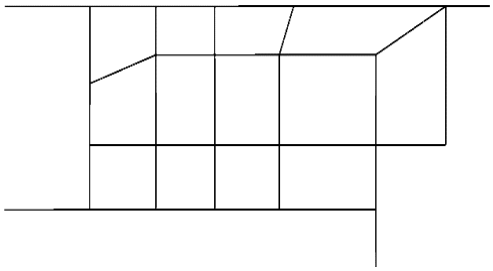

Figure 5. Appearance of the road

The black lines are the roads in the network. Each intersection is equipped with a traffic light.

The vehicles are configured to follow certain paths. In fact, the 4 corners have a road that acts as an entrance or an exit for vehicles. A vehicle which enters from a given corner will follow a road that brings them to one of the three remaining corners; depending on the route it was given.

For our tests, we will use SCV (see [15-16]), a simulator aimed at vehicular communications (DSRC standard). To tell SCV how we want the vehicles to navigate the network, we have to write an XML (eXtensible Markup Language) file detailing the different routes vehicles will take. Here is an example:

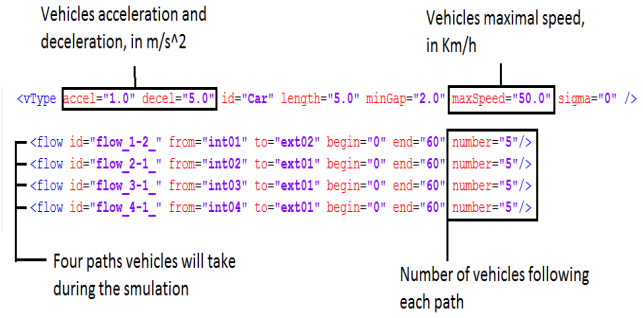

Figure 6: Example of an XML input

The complete initialization file we use dictates that 5 vehicles will go from each corner to the three other corners. This means that there is a total of 60 vehicles in the simulation (15 coming from each corner and 4 corners).

The initial trust score of each vehicle will be the same. It will not be given randomly in order to have a better observation of the behaviour.

During the simulations, we will vary the number of vehicles that send false information as well as the number of vehicles equipped with DSRC-enabled technology.

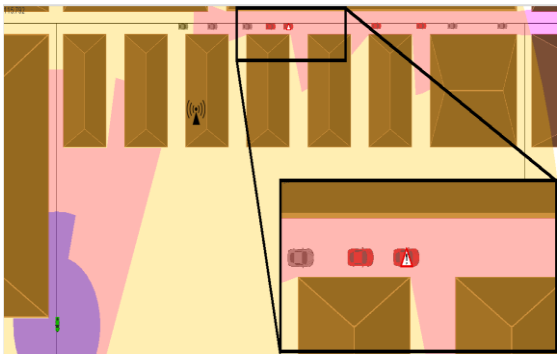

Figure 7. Extended perception in SCV using the ***TrustLevel*** scheme

In Figure 7, we can observe a certain DSRC-enabled vehicle (the green vehicle in the bottom-left corner) perceives

the elements contained in the purple and pink zone. It also recognizes that a vehicle outside of its line of sight is accidented causing slowdowns.

This complex perception seen in Figure 7 is possible with the cooperation of diverse trusted DSRC-enabled vehicles in the area. A simple malicious node in the system can cause numerous incoherencies and lowers the perception's accuracy (see Figure 9).

The goal is to judge the efficiency of the system in relation to the number of malicious vehicles in ideal and non-ideal scenarios (when not all vehicles are able to communicate). Firstly, we want to know if we are able to detect the malicious vehicles and then how fast they are detected.

As presented in Figures 8a and 8b, the malicious vehicle has been flagged by the neighbouring vehicle perception crossing.

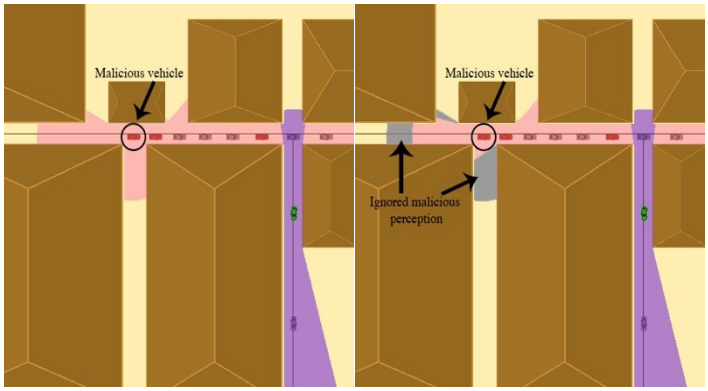

Figure 8a. Figure 8b.

Figure 8a. The extend perception of the green vehicle including the malicious vehicle's invalid perception.

Figure 8b. The extend perception of the green vehicle without the malicious vehicle's invalid perception.

B. Results of detection speed versus DSRC-enabled vehicle density

In each set of tests, we start with a third of the vehicles with DSRC capabilities. We then increase this number to two thirds of the vehicles and finally, 100%. It is important to note that only DSRC capable vehicles can send false information and be malicious, so when we say, for example, 25% malicious vehicles, it means 25% of the vehicles with DSRC capabilities.

For the first set of tests, 1 vehicle will be malicious, regardless of the number of vehicles. These are light cases. For set #2, 25% of the vehicles will be malicious. These are moderate cases. Finally, for set #3, 50% of the vehicles will be malicious. These are extreme cases.

In Tables 2 to 4, the decrease of the detection time of malicious vehicle(s) is inversely proportional to the number of DSRC-enabled vehicles in the area; more perceptions to cross, the easier the detection.

Table 2. With 1 malicious vehicle

	Time to detect half of the malicious vehicles (s)	Time to detect all of the malicious vehicles (s)
20 DSRC vehicles		81.640
40 DSRC vehicles		54.128
60 DSRC vehicles		37.44

Table 3. With 25% of malicious vehicles (set #2)

	Time to detect half of the malicious vehicles (s)	Time to detect all of the malicious vehicles (s)
20 DSRC vehicles	84.160	123.840
40 DSRC vehicles	54.976	67.456
60 DSRC vehicles	39.408	60.528

Table 4. With 50% of malicious vehicles (set #3)

	Time to detect half of the malicious vehicles (s)	Time to detect all of the malicious vehicles (s)
20 DSRC vehicles	82.624	N/A
40 DSRC vehicles	55.616	68.096
60 DSRC vehicles	51.168	64.624

For Table 3 and 4, the short delay between the detection of half of the malicious vehicles and the full detection is explained by the ***TrustLevel*** scheme fed with extra data coming from a larger pool of DSRC-enabled vehicles.

As highlighted in yellow in Table 3 and 4, the difference between half and full detection is about 13 seconds for 40 DSRC-enabled vehicles. With the increase of time comes newer trusted vehicles resulting in a better detection accuracy and a fast flagging of all malicious vehicles in the area.

The "N/A" highlighted in brown in Table 3 is caused by the lack of trusted vehicles to establish an accurate detection of malicious vehicles.

C. Results of packet reception reliability ratio

In Figure 9, the graphic presents three series of tests analysing the packet reception reliability according to the density of DSRC-enabled vehicles.

The first test was done without any incoherent perception, in other words: no malicious vehicles' perception. The second and third were done with a ratio of 25 and 50 percent of malicious vehicle versus the initial density of DSRC-enabled vehicles respectively. During our tests, we did not isolate the invalid perception of malicious vehicles to highlight the impact on the perception accuracy as low as 15% and 3% on

25% and 50% malicious ratio respectively for 60 DSRC-enabled vehicles.

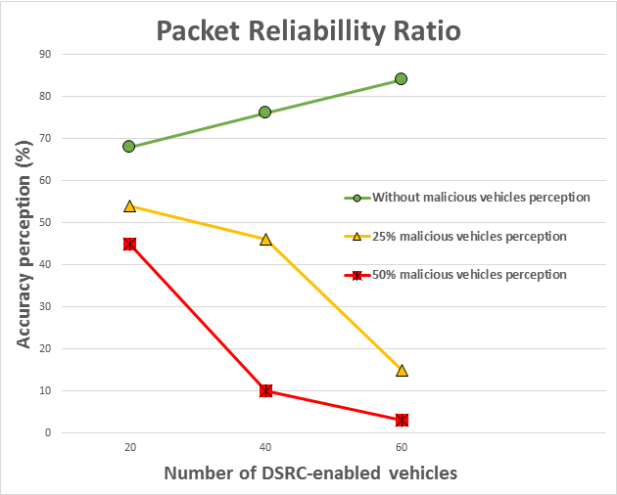

Figure 9. Reliability of perception computed by the ***TrustLevel*** scheme on without malicious vehicles' perception and on non-ignored malicious vehicles' perception

V. CONCLUSION

To summarize, we started, in step 1, by defining the initial attribution of points as well as the penalty and reward model. In step 2, a data structure was proposed which allowed us, in step 3, to detect false messages. Then, in step 4, the vehicles sending false information are penalized. Finally, in step 5, the detected vehicles have been flagged as malicious. In this paper, we show that it is possible to detect malicious vehicles using perception crossing. In the future, we plan to optimize the algorithms used in order to reduce the amount of data sent and improve its performance.

Acknowledgement

This research was financially supported by the "Fonds Québécois de la recherche sur la nature et les technologies (FRQNT)". We would like to thank the LRIMA laboratory member's Philippe Rivest and Michael Oliveira-Silva for their valuable comments.

REFERENCES

[1] N. Chaabouni, H. Abdelhakim, J. Rezgui and S. Cherkaoui, "Setting up an extended perception in a vehicular network environmnent: A proof of concept", Proceedings of IEEE Conference on wireless communications and networking (WCNC 2016), 3-6 Avr 2016.

[2] S S. Hakami, et al., "Detection and Identification of Anomalies in Wireless Mesh Networks Using Principal Component Analysis (PCA) ", in Proc. of Parallel Architectures, Algorithms, and Networks, pp. 266-271, 2008.

[3] R. Lin, E. Khalastchi and G. A. Kaminka, "Detecting anomalies in unmanned vehicles using the Mahalanobis distance", in Proc. of ICRA, pp. 3038-3044, 2010.

[4] J. Rezgui, S. Cherkaoui, "Detecting Faulty and Malicious Vehicles Using Rule-based Communications Data Mining", Proceedings of 36nd IEEE Conference on Local Computer Networks (LCN 2011), Bonn 4-7 Oct. 2011.

[5] R. Kaur et al., "Detection and Prevention of Malicious Vehicles with Bacterial Foraging Optimization in VANETs", International Journal of Advanced Trends in Computer Applications (IJATCA) Volume 2, Number 2, July - 2015, pp. 1-6 ISSN: 2395-3519.

[6] V. Lakshmi Praba, "Detecting Malicious Vehicles and Regulating Traffic in VANET using RAODV Protocol ", International Journal of Computer Applications (0975 – 8887) Volume 84 – No 1, December 2013.

[7] Imran Raza, S.A. Hussain, "Identification of malicious nodes in an AODV pure Ad-hoc network through guard nodes" Elsevier Computer Communications vol 31, Issue 9, June 2008, pp 1796–1802.

[8] M. E. Zarki, et al., "Security issues in a future vehicular network", in Proc. of EuroWireless, 2002.

[9] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks", in Proc. of SASN, 2005.

[10] K. Sampigethaya, et al., "Caravan: Providing location privacy for vanet", in Proc. of ESCARworkshop, 2005

[11] J. Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks", in Proc. of Q2SWinet, 2005.

[12] E. Coronado, S. Cherkaoui,"A secure service architecture to support wireless vehicular networks", Special Issue on "Security, Trust, and Privacy in DTN and Vehicular Communications", International Journal of Autonomous and Adaptative Communications Systems (IJAACS), Inderscience, Vol3, No.2, pp 136-158, 2010.

[13] G. Grahne, J. Zhu, "Efficiently using prefix-trees in mining frequent itemsets", in Proc. of the Workshop on Frequent Itemset Mining Implementations (FIMI), Florida, USA, 2003.

[14] K.K. Loo, I. Tong, B. Kao, and D. Chenung, "Online Algorithms for Mining Inter-Stream Associations from Large Sensor Networks", in Proc. of PAKDD'2005, Hanoi, Vietnam, The Ninth Pacific-Asia, 2005.

[15] J. Rezgui, C. Doucet and P. Alexandre, "An Overview of the SCV Simulator for Vehicular Networks ", Proc of International Conference on Computer Networks and Communication Technology, DOI:10.2991/cnct-16.2017.72 (CNCT 2016).

[16] C. Doucet, P. Alexandre, and J. Rezgui "Simulateur de communications véhiculaires", https://gitlab.com/polkio7/SCV, 2016.