Trust and exclusion in vehicular ad hoc networks: an economic incentive model based approach

Nadia Haddadou, Abderrezak Rachedi, Yacine Ghamri-Doudane

Trust and Exclusion in Vehicular Ad Hoc Networks: An Economic Incentive Model based Approach

Nadia Haddadou*, Abderrezak Rachedi*, and Yacine Ghamri-Doudane*,†
*Université Paris-Est - Gaspard Monge Computer Science Laboratory (LIGM - UMR 8049)
75420 Champs sur Marne, France
† ENSIIE, 1 Square de la résistance, 91025 Evry Cedex, France

***Abstract*—In this body of work, we are interested in road safety applications such as advanced driver assistance systems, based on Vehicular Ad Hoc Networks (VANETs). One of the particular characteristics of this kind of networks is the continuous sharing of safety information by its nodes. Since this kind of information is time sensitive, a node cannot spend much time to verify its validity with an authority. However, the presence of *malicious* and *selfish* nodes in VANETs corrupts exchanged data, and lowers the overall data reception ratio in the network. To tackle this, we propose a new *incentive model with exclusion for malicious nodes* called VIME. VIME is inspired from the signaling theory from economics. It is based on managing a credit count that each node receives at the initialization of the application. Straightforwardly, VIME is based on two pillars. On the one hand, a node pays an appropriate cost for each sent message, which is seen by the receivers as a *guarantee* from the source about the truthfulness of the information. On the other hand, nodes get rewarded for cooperating in the network. The proposed economic model allows computing the amounts to be paid and those to be awarded in order to fight selfish and malicious nodes. We validate our approach via simulations. We show that VIME is able to detect and evict gradually all malicious nodes in the network, and decreases the ratio of corrupted and false sent data until reaching zero. Moreover, it has a positive impact on the participation of selfish nodes, as our approach increases the average ratio of sent data as to equal the ideal case's percentage, when no selfish node is present.**

I. Introduction

Road traffic has greatly benefited from the recent network expansion. As a result, several applications have emerged having as support Vehicular Ad Hoc Networks (VANETs). Communication between vehicles is ad-hoc, using dedicated short-range communications (DSRC). This type of networks has several proprieties to take into consideration, such as high mobility, extended area, and frequent topology changes.

In this paper, we are interested in safety applications, such as advanced driver assistance systems, built on top of VANETs, where data related to road traffic is broadcasted. This kind of applications is usually time-constrained, and carries important contents whose transmission has to be completely and trusty achieved. Some solutions like ADCD [1] and EMPR-PD [2] proposed to increase the data reception ratio and to reduce the overhead as well as long latency. Nevertheless, there are still remaining security and truthfulness issues. A number of attacks are caused by the diffusion of false information or the removal of an accurate one. There are two kinds of nodes that can affect the good functioning of such systems using VANETs. The first kind is the *malicious* node. For example, a malicious node can constantly alternate between bad and good behavior by sharing true and false information (i.e. to avoid being detected). The second one is qualified as *selfish but rational*. It refuses to cooperate by relaying messages, in order to save its own resources. A large number of these nodes cause a decrease in the cooperation in a network, which weakens its connectivity and reduces its offered services.

The usual solution for establishing a trust metric between members of a network is a reputation model [3], where members are assigned a reputation within a numeric interval. This allows nodes to have a local view of the network members, and to share it with their neighbors. A node's reputation is negatively affected among its neighbors if a bad behavior (*e.g.* sending false information) is detected. However, if the deployed reputation system is *distributed*, due to the lack of infrastructure in the roads, a node can easily be unrecognized in another area and recover a neutral reputation among its new neighbors, by simply changing directions. Due to VANET's characteristics, such as high mobility, extended deployment area, and large number of members, *the lack of a global view of the network and its members makes a reputation model unsuccessful and insufficient for safety applications in VANETs*.

In order to obtain a larger view and make safer exchanges, we have to impinge resources to nodes, as it is done in credit-based models [4], [5] and [6]. Unlike a reputation value, a credit amount does not change because of the mobility of a particular node, which allows to quickly detect and evict a malicious node based on this value. These credit-based solutions focus only on improving nodes' cooperation. To tackle these limitations, we propose VIME, a VANET Incentive Model with Exclusion for malicious nodes, to deal with both selfish nodes to improve their cooperation, and with malicious nodes to detect and exclude them from the network. VIME implements cost and reward functions to handle the account of each member according to its behavior, thus inciting nodes to act well. Each node is credited at its first connection to the network with a fixed amount of credits. For each sent message, the source pays some cost depending on different factors, such as the relevance of the information and, the node's reputation among its direct neighbors. This represents a guarantee of trustfulness for the receivers, and an investment

for the source since a corresponding reward will be given to the node if the shared data is considered as valid. Therefore, a node can only receive and send messages if it has credits left, or else it is detected and excluded from the network.

The system described here is similar to the signaling theory in economics [7], whose main concept is to deal with the information asymmetry in a market, between customers and sellers, regarding the quality of the offer. As in a market, the information asymmetry about the nodes' behavior in a network, between message senders and recipients, is caused by the lack of overview among them, thus inducing the emergence of malicious nodes. To detect them and to incite selfish nodes to cooperate, we propose using a signal, which is an amount with a corresponding cost for the source node, used for each sent message, to influence the neighbors' decision about the message's validity. To reward good members, some credits are given to them from their neighbors. This reward depends on the chosen signal value, and represents an incentive to cooperate and to continually behave well, as the credit is used to have access to many privileges in the network such as receiving neighbors' messages. If a node runs out of credits, then it is considered as malicious, and is evicted from the network.

The remainder of this paper is organized as follows. In Section II we present some related work in incentive solutions and their applicability to VANETs. In Section III we present VIME, our proposed approach. A simulation-based performance study is presented in Section IV. Finally, Section V concludes this paper.

II. Related Work

Existing solutions about cooperation propose to motivate nodes to forward other's packets in return for a reward, in order to minimize the number of selfish nodes. This is the general concept of incentive methods using cost/reward. The approach presented in [4] provides a solution using nuggets as method of payment for the incentive part. A node loads into its packet a number of nuggets corresponding to an estimate of the reward for the required intermediate nodes in order to reach the destination node. The main limitation of this method is the reward estimation since, in both cases of underestimation and overestimation, a part of the nuggets is lost. A second scheme presented in [4] is the packet Trade Model. It uses a negotiation between intermediate nodes, such that an intermediate node buys a packet and sells it to the next node in such a way that it will make a benefit. But the destination node, being the last receiver, supports all the forwarding costs.

In [5], authors propose rules for the nodes to choose their degree of cooperation within the network according to the node's amount of nuggets. This method requires nodes to cooperate quantitatively, which does not always ensure a constant cooperation over time. Our system differs in the effort required by a node for cooperation. Reputations are frequently updated and their value have a great impact on the costs and the rewards for sending messages. This incites nodes to cooperate well and regularly, especially since a good reputation greatly reduces the cost of sending messages.

Another incentive solution proposed in [9], introduces a sweepstakes component to enhance more nodes' collaboration. The authors incite intermediate nodes by proposing them weighted rewards, in addition to a fixed reward to one of them, chosen probabilistically. This solution depends on the existence of infrastructures, responsible of the rewarding and the reception of nodes' receipts about their cooperation actions. Nevertheless, this solution does not deal with the presence of malicious nodes.

Another approach presented in [6] proposes an incentive model for mobile nodes using infrastructures as a Credit Clearance Service (CCS). Each node has to keep a receipt of its actions in the network. Hence, upon verification by the CCS, it receives a reward. The main drawback of this solution is the multiple verifications to ensure a correct functioning of the whole chain. It introduces a lot of latency and overhead which do not guarantee the scalability in VANET. In [8], authors propose to use an identity-based cryptosystem, to allow network authorities to handle misbehaved nodes. This solution needs traceability information about nodes, obtained with the deployment of roadside infrastructures and law enforcement authorities. However, we argue that solutions are more scalable and heterogeneous if their operations do not depend on an infrastructure, which is the case of our approach, VIME.

Most existing incentive models do not deal with message diffusion. Moreover, regarding unicast, some solutions rely on estimation of rewards, which is not a useful parameter in VANETs as routes change very often; or they use a constant price for the reward, which does not always represent an incentive for the nodes. Others aim at solving either the selfish nodes problem or the malicious node problem. In our case, we believe that both should be handled at the same time. Considering these limitations, our work uses different parameters to calculate a fitted price so that each node is a winner while applying our mechanism. Moreover, it links an incentive and an exclusion module to eradicate malicious nodes and motivate selfish ones.

III. VIME: A VANET Incentive Model with Exclusion for Malicious Nodes

VIME is a novel approach to ensure the truthfulness of shared information and handle high mobility in VANETs. It uses a credit management system to eradicate dishonest nodes, and to increase the participation of selfish nodes. VIME implies that each node has a unique identification in the network, and a tamper-proof credit count [10], which is credited initially with a fixed value. A node uses its credit for two cases. The first one is for sending a message and the second one, is for rewarding a neighbor about a shared information if it considers the content as well as the paid cost of sending it as valid. The enrichment or impoverishment of a node are related to its behavior and its degree of cooperation in the network.

A. Credit Management Functions

1) Sending Cost: According to the market strategy, a source node takes on the role of a service provider that must offer a guarantee about its products to its customers. This guarantee depends on many parameters such as the source node's reputation, a standard cost set by the application, and the importance of the data. A guarantee cost for a message is individual and differs for the same information, even between the source node and the relayed node for this message. In our environment, this guarantee is translated into a cost called $Cs_i^t(N(i))$ used to diffuse a message from node i to its neighbors $N(i)$. This cost has to be substantial for the source node according to its initial amount of credits given at its first connection to the network, $init\_credit$. It also has to be consistent with respect to the importance of the shared information, i_msg, set on a range of 0 to 1 as discussed in [1]. Finally, it has to meet at the same time all of the neighbors' expectations in terms of guarantee, so that the receivers consider and accept the message. The source node pays a cost as computed in equation (1) by decrementing its credit count via a tamper-proof counter.

$$Cs_i^t(N(i)) = \frac{init\_credit}{G} \times i\_msg \times (2 - R_i^t(i)) \quad (1)$$

A parameter G is used for the cost calculation. It divides the initial credit that a node receives, in order to establish a reference value for message costs. G allows to increase and to decrease the number of sent messages whose cost is paid only with the initial credit. Using a small value for G, allows quickly depleting a malicious node's credit count (e.g. in case of no rewards), thus making it unable to participate in the network. Using a large value for G allows the unknown nodes, for example, to have more chances of sharing their messages in order to increase their reputation among their neighbors, especially in the case of low density and high mobility. The second parameter of the equation, i_msg, is about the diffused data relevance. Indeed, the more the data is important, the more there is consequences if it is corrupted. Therefore there is a correlation between it and the cost.

The third parameter is $R_i(i)$ (i.e. a reputation about the node i held by itself at time t), based on the received reputations from its neighbors about it. The reputation values are computed as described in [11], these values are continuous and belong to the $[-1, 1]$ range. $R_i(i)$ represents a personal and momentary estimation of the node's reputation among its current neighbors. Hence, the cost is also dependent on the reputation of the source among its neighbors, which are the receivers and the decision-makers about the truthfulness of the data. The receivers check if the paid cost corresponds to the message and their estimations of it, and then check the validity. A reputation makes the cost more or less expensive in order to offer a greater guarantee, especially when the reputation is bad. To meet the receivers' expected guarantees, and to receive rewards from them, a source node has to be able to know or estimate its reputation among them.

2) Reputation Relevance for the Reward: VIME uses a function for the reputation relevance, $\mathrm{W}(R_i^t(j))$. It assigns a weight to $R_i^t(j)$, the reputation value of a sender/forwarder node j computed by a receiver node i at time t. This weight is used for the reward estimation. It allows higher rewards for high reputation nodes compared to lower reputation nodes for the same action, thus inciting nodes to maintain good reputations.

$\mathrm{W}(R_i^t(j))$ is computed with equation (2), where a hyperbolic tangent is used because of its characteristics, namely its behavior that closely resembles that of an exponential function for both positive and negative values. The $\tanh$ result is multiplied with a factor in order to obtain results in $[-1, 1]$ range.

$$W(R_i^t(j)) = \tanh(R_i^t(j)) \times \frac{e^2 + 1}{e^2 - 1} \quad (2)$$

3) Reward Value Computation: At the reception of a message, a node checks the cost paid by the source. To do that, the receiver calculates a range of tolerated paid cost for the sent message, $[Cs_i^t(j)_{min},\ Cs_i^t(j)_{max}]$ as described in equation (3). This is performed in order to cope with information asymmetry in the network, i.e. in case of divergence about the reputations of each node, which may be caused by the node's mobility and topology changes.

$$\begin{cases} Cs_i^t(i)_{min} = \frac{init\_credit}{G} \times i\_msg \times (2 - \max(R_{N(j)}^t(i))), \\ Cs_i^t(i)_{max} = \frac{init\_credit}{G} \times i\_msg \times (2 - \min(R_{N(j)}^t(i))), \end{cases} \quad (3)$$

The minimum boundary of this range is set to maintain a mandatory guarantee for the message. The maximum boundary is to prevent any abuse from malicious nodes. For example, they could use a greater cost for the only trusted information that they send in order to obtain a greater reward corresponding to their paid cost, thus sharply increasing their credit count. The minimum cost calculated by a receiver node j is based on the maximum held reputation of the source node attributed by it or received from its neighbors $N(j)$, and vice versa for the maximum cost.

The reward's aim is to make sure that the system remains incentive for the good nodes and dissuading for the malicious one. Moreover, it represents a cost to receive information. This decreases nodes' credit count when rewarding source nodes. Furthermore, in order to earn credits, and reward received messages, selfish nodes are encouraged to cooperate.

When the cost paid by a source node for its message lives up to the receiver's expected guarantee, the receiver examines the message and decides about its confidence on the data. If the receiver considers the message as valid, it rewards the source node by sending to it some credit via its tamper-proof credit count. The amount depends on the cost paid by the source, the density around it, which correspond to the potential rewarder nodes, and the held reputation value about the source node. The reward $Rew_j^t(i)$ from a node j to a node i at time t, is then calculated as follows in equation (4).

$$Rew_j^t(i) = \frac{(W(R_j^t(i)) + 1) \times Cs_i^t(N(i))}{|N(j)|} \quad (4)$$

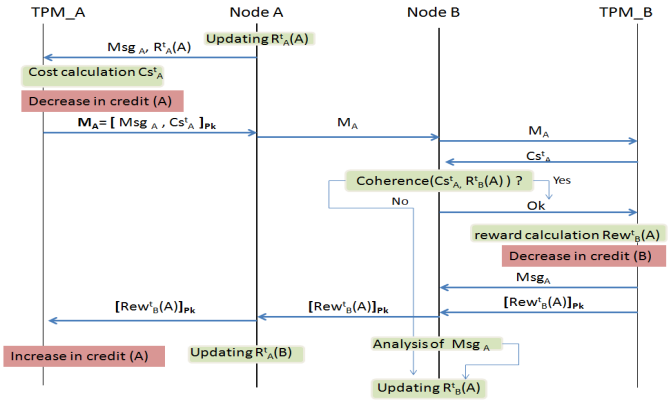

Fig. 1. Message Exchanges Using TPM

A reward, Rew, is estimated according to a weight , W, given to the source node's reputation, $R_i^t(j)$, in order to provide higher remunerations to the nodes with greater reputation values, and to delay the credit count increase of the less reputed ones. For nodes with average reputation, their reward only recovers their paid cost, Cs_i^t, for the message, so there is no loss and no profit for them. However, their reputation can increase which promises them a weighty gain later. To maintain consistency, the reward value depends also on the cost paid by the source, and on the number of neighbors, $N(i)$, which are supposed to reward the source. In order for a reward to be divided among them, even if the estimation of each one of them differs because of asynchronous information, an approximate reward can be reached, ensuring a certain level of coherence for the reward of the source node.

B. Tamper-Proof Credit Count

To guarantee a smooth functioning for VIME, we assume that each node has a Trusted Platform Module [10], TPM, which is a secure piece of hardware with cryptographic capabilities, able to implement elliptic curves in order to generate keys according to the IEEE 1609.2 Elliptic Curve Digital Signature Algorithm [12], and to store data in shielded locations. To guarantee its security and integrity, the TPM stores a fingerprint of the application, which leads to detect any change by an attacker.

TPM is frequently used to build secure architectures for Vehicular Ad Hoc Networks [13]. In our case, the TPM is used to manage a tamper-proof credit[1]. The security requirements of our solution are about trustworthiness of paid costs, which aims at deducing the paid sum from the node's account, and updating its credit amount, while supporting large-scale deployment. To do this, TPM encrypts the signaling cost and reward, and stores the credit account into its shielded storage. The use of a TPM allows VIME to be distributed, and not based on the presence of any infrastructure. Fig. 1 illustrates the message exchange for either new captured information or forwarded messages, between nodes A and B using their

[1]The TPM is not exclusive to VIME. It can also be used by other applications and for other purposes

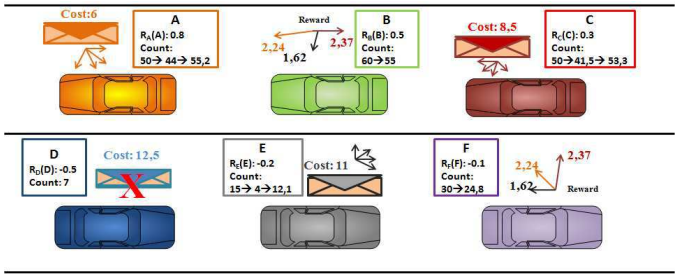

Fig. 2. VIME process

TPM (TPM_A and TPM_B, respectively). Firstly, node A updates its own reputation, $R_A^T(A)$, using its neighbor's shared impressions about it. This value is added to the data to share, and is sent to TPM_A as Msg_A, in order to calculate the corresponding cost, Cs_A^t, using equation (1). Then, TPM_A returns a signed message M_A to node A, which is the cost added to the data encryption including a timestamp, by using its private key *pk*. A will then broadcast M_A to its neighbors.

In our example, node B is a neighbor of A and receives M_A. Foremost, node B asks its TPM about the used cost by A and compares it to the reputation that it holds about it. If the two values are coherent, according to equation (3), and if node B holds no information that would cast doubt on the veracity of the received data, (e.g. a too bad reputation of the source, or an incoherence in the timestamps and the localization of the event as discussed in [14]), then node B accepts the information, and therefore has to reward node A. Then, TPM_B decrypts the data and returns it to node B, and subtracts a corresponding reward for node A according to its deployed cost and its reputation, using equation (4). Finally, it returns the new message, containing the signed reward value to node B, which is sent subsequently to node A, in order that its TPM increases its credit count. Thereafter, the reputation of the two nodes can be updated. A's reputation is updated because of the validity of its sent data, and B's reputation is updated for the cooperation by sending it a reward.

To avoid frauds, the calculations of the cost and the reward are made and signed by the TPM, which adds or subtracts immediately this value from the credit account.

C. Example of VIME's Operations

Fig. 2 illustrates the workings of VIME. Vehicles are able to send messages after paying a cost, calculated using equation (1). According to their reputation, vehicle A should pay 6, and vehicle C $8,5$ when $init\_credit$ is 100, G parameter is 20, and i_msg is 1. The transaction is secured using the TPM of the sending node, which authenticates the subtraction from its credit count. Vehicle E pays more than vehicle C, which in turn pays more than vehicle A because of their different reputations. When credits are insufficient to pay the corresponding cost, as in the case of vehicle D, a node cannot send its message, and when its credit is exhausted, it is evicted from the application and considered as malicious.

As sent messages are encrypted, a receiver has to choose

to accept it or to refuse it beforehand. Therefore, at each message reception, nodes verify if the cost paid by the source is coherent with equation (3). Once this verification is done, a node informs its TPM about its decision, and if it considers the data as valid, then the TPM decrypts it and delivers it to the node, while computing a reward for the source node, using equation (4) (e.g. vehicle B rewards A with 2.24, E with 1.62, and C with 2.37). Finally, the TPM deducts the reward from the credit count of the receiver. As for the sending node, the reward process is secured by the TPM of the receiving nodes. By performing this entire procedure, VIME encourages cooperation among nodes, and aims at automatically selecting good members in the network by proposing them interesting rewards and low costs (e.g. vehicle A earns after deducting its sending cost 6.7 in the case where the five neighbors participate in the reward process, while vehicle C earns 3.35, and E loses 2.9. Credit counts of the other vehicles diminish during the reward process, because of the reward they pay to remain informed. Conversely, they will earn credits when sending correct data. Indeed, to be able to be informed at all times, nodes have to always posses enough credits to decrypt the received messages. This is possible provided that a node cooperates and behaves well, which incites selfish nodes to cooperate more.

IV. Performance Evaluation

We evaluate the performance of our model by analyzing its effectiveness on detecting malicious nodes and inciting selfish nodes to cooperate, in order to improve the received ratio of truthful data and decrease that of corrupted data. We conducted a set of simulations using NS2 [15] for 100 nodes moving in a 5 kilometer highway, making round trips and some stops at both ends, according to a mobility scenario with a velocity in the range of $90 - 160$ km/h during 3600 seconds, such as an event is detected and transmitted each one second. We used VanetMobiSim [16] as vehicular mobility model. The simulated radio transmission range is 250 meters, using IEEE 802.11 as the MAC layer protocol. We use ADCD [1] to optimize the diffusion of data and to minimize the communication overhead.

During the simulation, we introduce different ratios of malicious and selfish nodes to evaluate their effects on the network, and how much our solution remedies to that. For our evaluation, we selected the three following performance metrics:

- Percentage of detected nodes: Either malicious nodes or false positives (non-malicious nodes detected as malicious).
- Average ratio of corrupted data: Measures the impact of malicious nodes on the network.
- Average ratio of received data: Measures the impact of selfish nodes on the network.

In our scenario, a malicious node always sends false data and, during the forwarding of a received message, it corrupts the data before retransmitting it. On the other hand, a selfish node is rational. It cooperates and forwards others' messages when it requires credits to provide for its own needs. For these simulations, we set the threshold of needed credits for a selfish node to the half of the initial credits received in the beginning, fixed to 1150 during the simulation.

First, we studied the detection percentage of malicious nodes, and the percentage of false positives in a network composed of 16% and 25% of malicious nodes. Results are presented in Fig 3. We can evidence that the percentage of malicious node detection gradually increases, which occurs when they run out of credit. By the end of our simulated time, 100% of the total number of malicious nodes in the network have been detected and then removed from the network application. Moreover, the percentage of false positives, which is the percentage of good nodes detected as malicious, does not exceed 4% in both cases.

By analyzing in detail the simulation results, we found out that nodes that have been mistakingly evicted are those who send/forward very few packets (between 10 and 20 packets during the simulation time), compared to the number of packets they receive. In comparison, other nodes send/forward five to ten times more packets (around 100 packets). These "false positive" nodes are forwarding less packets, not because they behave selfishly, but because of the use of ADCD [1]. Indeed, due to ADCD's optimizations with the identification of target areas to the broadcasting strategy, according to the data importance in order to avoid overhead, some nodes that are in the boundary of the diffusion zone often consume packets (i.e. send rewards), but do not forward them (i.e. do not receive rewards). One possible solution for reducing the number of false positive exclusions is to have a threshold under which a node in the boundary of the diffusion zone (i.e. the last one to receive the packet) will not accept any packet as it will not be able to forward them and thus get rewarded. Even though this solution will decrease the performance of the diffusion approach, it allows avoiding the false positive exclusions. This feature can be tied to message priority. High priority messages will always be accepted, while low priority ones can be rejected.

The second studied metric is the average ratio of corrupted data sent through the network during the simulation time. Since a new event is detected every second and sent using the target diffusion protocol, ADCD, the aim is to only deliver the data to the concerned vehicles. In the case where a malicious node sends a false data or a corrupted one about a truthful previous one, a receiver well-behaving node may not detect this immediately and may thus retransmit the false data, which increases the amount of false data in the network. Therefore, it is important to rapidly detect and remove the malicious nodes from the network. Fig.4 shows the average ratio of corrupted and false sent data in a network with different percentages of malicious nodes: 16% and 25%. We compare the performance of VIME with a network not using any solution, and network using MEB_Trust, a majority based and experience based trust algorithm adapted from [17], where a node always asks its neighbors about the validity of a received message before accepting it. Feature of our solution decreases this ratio until

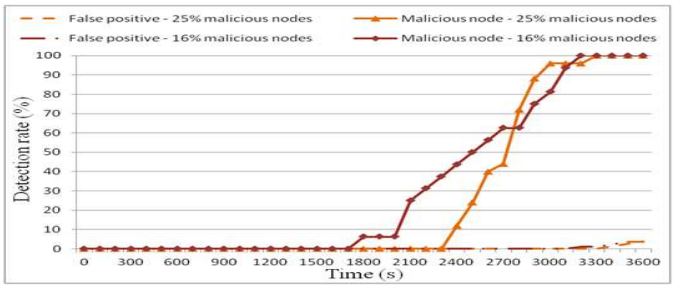

Fig. 3. Percentage of detected nodes, malicious and false positives for a network with 16% and 25% of malicious node.

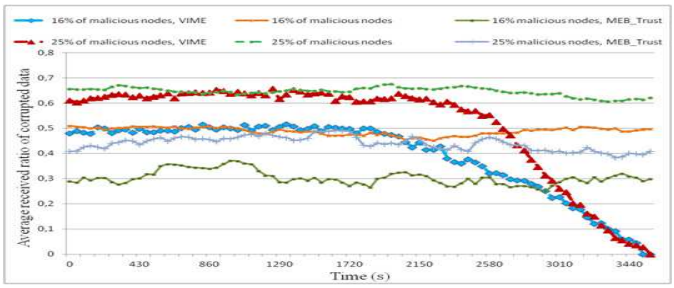

Fig. 4. Average ratio of corrupted data for a network with 16% and 25% of malicious node.

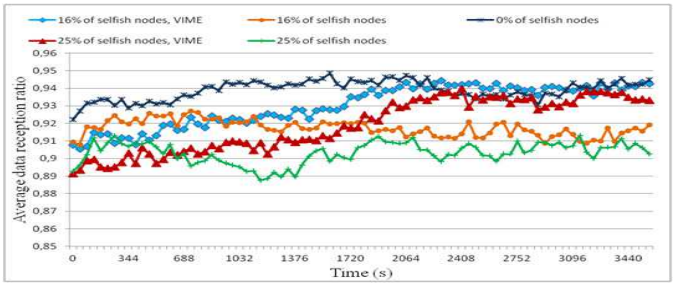

Fig. 5. Average ratio of received data for a network with 0%, 16% and 25% of selfish nodes.

reaching 0 after $3600s$ in both studied cases. The decrease of the received ratio of corrupted data is due to the detection and eradication of the malicious nodes. Unlike MEB_Trust, which does not eradicate malicious nodes from the network. The third metric, represented in Fig.5, shows the effectiveness of inciting selfish nodes to cooperate in the network. Different compositions of networks are made, with 0%, 16% and 25% of selfish nodes. The case where there are 0% of selfish nodes represents the ideal case. Note that after $2000s$ of simulation time, the ratio of received data in networks with 16% and 25% of selfish nodes using our incentive model is close to the ideal case, which proves the effectiveness of our approach in creating the need for selfish nodes of having credit, and thus increasing their cooperation in the network.

Furthermore, the generated messages using our solution do not interfere on the scalability of the network. Indeed, as our solution aims primarily at chiefly ensuring safety information, which generally occurs at sparse times and areas, the generated traffic is not continuous.

V. CONCLUSION

In this work, we propose VIME, an *incentive model with exclusion for malicious nodes*, inspired from *signaling theory* from economics. It is based on the management of an initial credit, that each node receives at the initialization of the application. VIME is an effective solution to cope with *both malicious and selfish nodes* in VANETs, without requiring the deployment of any infrastructure.It *detects and evicts* malicious nodes from the network when they have no more credits left. Furthermore, it *incites* selfish nodes to cooperate more by proposing interesting rewards. We showed via simulations that VIME detects gradually all malicious nodes in the network, while at the same time decreasing the ratio of corrupted and false sent data. Our solution also increases the participation of selfish nodes in a network, as to equal the percentage of the ideal case, with no selfish nodes in the network.

REFERENCES

[1] N. Haddadou, A. Rachedi, and Y. Ghamri-Doudane, "Modeling and performance evaluation of advanced diffusion with classified data in vehicular sensor networks," *Wireless Communications and Mobile Computing*, vol. 11, no. 12, pp. 1689–1701, Oct. 2011.

[2] C. Wu, S. Ohzahata, and T. Kato, "A broadcast path diversity mechanism for delay sensitive vanet safety applications," in *VNC*, Amsterdam, The Netherlands, 2011.

[3] K. Govindan and P. Mohapatra, "Trust computations and trust dynamics in mobile ad hoc networks: A survey," *IEEE Communications Surveys and Tutorials*, vol. Preprint, pp. 1–20, 2011.

[4] L. Buttyán and J. Hubaux, "Enforcing service availability in mobile ad-hoc wans," in *MobiHoc*, Boston, USA, Aug 2000.

[5] ——, "Stimulating cooperation in self-organizing mobile ad hoc networks," *ACM/Kluwer Mobile Networks and Applications (MONET)*, vol. 8, pp. 579–592, 2001.

[6] S. Zhong, J. Chen, and Y. Yang, "Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks," in *INFOCOM*, San Francisco, USA, Apr. 2003.

[7] M.Spence, "Signaling in retrospect and the informational structure of markets," *The American Economic Review*, vol. 92, no. 3, pp. 434–459, 2002.

[8] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, "An identity-based security system for user privacy in vehicular ad hoc networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 9, pp. 1227–1239, 2010.

[9] F. Li and J. Wu, "Frame: An innovative incentive scheme in vehicular networks," in *ICC*, Dresden, Germany, June 2009.

[10] "Trusted computing group: Tpm main specification. main specification version 1.2 rev. 116," March 2011.

[11] C. Zouridaki, B. Mark, M. Hejmo, and R. Thomas, "E-hermes: A robust cooperative trust establishment scheme for mobile ad hoc networks," *Ad Hoc Networks*, vol. 7, no. 6, pp. 1156–1168, 2009.

[12] "Ieee 1609.2-standard for wireless access in vehicular environments (wave) - security services for applications and management messages, available from its standards program."

[13] G. Guette and C. Bryce, "Using tpms to secure vehicular ad-hoc networks (vanets)," *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, vol. 5019, pp. 106–116, 2008.

[14] N. Lo and H. Tsai, "Illusion attack on vanet applications - a message plausibility problem," in *IEEE Globecom Workshops*, Washington, USA, 2007.

[15] The NS-2 website. [Online]. Available: http://www.isi.edu/nsnam/ns/

[16] M. Fiore, J. Härri, F. Fethi, and C. Bonnet, "Vehicular mobility simulation for vanets," in *ANSS*, Norfolk, USA, Mar. 2007.

[17] U. F. Minhas, J. Zhang, T. Tran, and R. Cohen, "Towards expended trust management for agents in vehicular ad-hoc networks," *International Journal of Computational Intelligence: Theory and Practice(IJCITP)*, vol. 5, no. 1, pp. 03–15, Jun. 2010.