On Reputation and Data-centric Misbehavior Detection Mechanisms for VANET

by

Zhen Huang

Thesis submitted to the
Faculty of Graduate and Postgraduate Studies
In partial fulfillment of the requirements
For Masters of Applied Science degree in
Electrical Engineering

School of Electrical Engineering & Computer Science
Faculty of Engineering
University of Ottawa

Acknowledgement

I would give my sincerest gratitude to Prof. Amiya Nayak, my supervisor, for his exceptional guidance and support not only in this thesis but also in my study and research. I am grateful to my co-supervisor Prof. Ivan Stojmenovic for his help in my study. Special thanks to Dr. Sushmita Ruj who gave me valuable advices, suggestions and encouragements. Without her instructions, I could not have learned so much and complete this thesis.

I wish to express my cheers to my friends whom I learned from. We will be friends as we were before.

Finally I would like to thank my parents who always understand and support my ideals, and this work is dedicated to them.

Abstract

Vehicular ad hoc networks (VANET) is a class of ad hoc networks build to ensure the safety of traffic. This is important because accidents claim many lives. Trust and security remain a major concern in VANET since a simple mistake can have catastrophic consequence. A crucial point in VANET is how to trust the information transmitted when the neighboring vehicles are rapidly changing and moving in and out of range. Current reputation systems for VANET try to establish trust between entities, which might not be required for practical scenarios. Due to the ephemeral nature of VANET, reputation schemes for mobile ad hoc networks (MANETs) cannot be applied to VANET. In this thesis, we point out several limitations of reputation trust management schemes for VANET. In particular, we identify the problem of information cascading and oversampling, which commonly arise in social networks. Oversampling is a situation in which a node observing two or more nodes, takes into consideration both their opinions equally without knowing that they might have influenced each other in decision making. We show that simple voting for decision making, leads to oversampling. We propose a solution to overcome this problem in VANET. We also suggest new ways to merge reputation schemes with misbehavior detection schemes to establish a trustworthy VANET.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Vehicular ad hoc network (VANET) is a mobile ad hoc network that provides wireless inter-communication among nearby vehicles or the communication between vehicles and fixed roadside infrastructures. The main aim of this technology is to give drivers more comfortable and more secure driving experience. Based on automatic information exchange between cars and infrastructures, the drivers could know the road conditions or the information about the parking lots immediately. VANET makes Intelligent Transport System (ITS) become reality.

1.1 Background

1.1.1 Why VANET?

Vehicles no doubt make our life easy to move around, but on other side they increase safety risk. According to the National Highway Traffic Safety Administration (NHTSA) report, a total of 37,261 people got killed in traffic accidents [4] in 2008. In the same year, 73,484 people died and 304,919 were injured because of traffic accidents in China [5]. Some of these accidents could be avoided if the driver could

be given warnings in advance or the vehicles had the ability to choose the lane to divert from the crash cars in advance. Hence, the concept of *Vehicular Ad Hoc Network* (VANET) was introduced, in which vehicles could exchange messages using their equipped communication devices. Vehicles broadcast and receive messages, and through this way, drivers are aware of their surrounding road conditions. As a result, they can react to some events in advance, before they actually see the event. If a vehicle disseminates congestion information message in its communication range, it can help other vehicles choose alternative route and path.

Apart from traffic safety, VANET supports various applications for on-line toll payment, Internet access and even entertainments to improve passengers' comfort. This is called Intelligent Transportation System (ITS) which combines communications technology and information applications with vehicles to support more safe transportation and reduce the congestion time and fuel consumption.

1.1.2 Architecture of VANET

As shown in Figure 1.1, VANET consists of vehicles (also referred to as nodes), *road side units* (RSUs) and *trusted authority* (TA) or *certification authorities* (CAs), whose goal is to ensure road safety and help in secure transfer of message and data. Each vehicle has an *on board unit* (OBU), which transmits messages about the position, speed, acceleration/deceleration, alert signals, etc. OBU also has authentication capabilities, to verify that an incoming message has been broadcasted by a valid entity. RSU is a physical stationary communication device and responsible for collecting and disseminating critical information such as the nearest parking lots and gas price. RSU is equipped with at least a network device for short range wireless communication and Internet. RSU can be deployed in intersection, it acts as a gateway between In-

ternet and OBU which enables vehicles to establish connections with Internet. RSUs also help in coordinating and collecting information about vehicle activities (e.g. red light violations). TA is actually a fully trusted party, it could be the department of government and stores the information of all vehicles. As usual, TA cannot be compromised by adversary and has sufficient storage.

Communication in VANET can either be *vehicle-to-vehicle* (V2V) (e.g. relaying alert information) or *vehicle-to-infrastructure* (V2I) (e.g. when the vehicle needs to report some event to the CA). V2V communication allows vehicle send and receive valuable messages; vehicles communicate with each other directly without the help of other infrastructures. V2I indicates one way for OBU to connect with RSU; vehicle can establish connections with Internet through RSU. The message sent by one vehicle might have important security implications such as accident prevention.

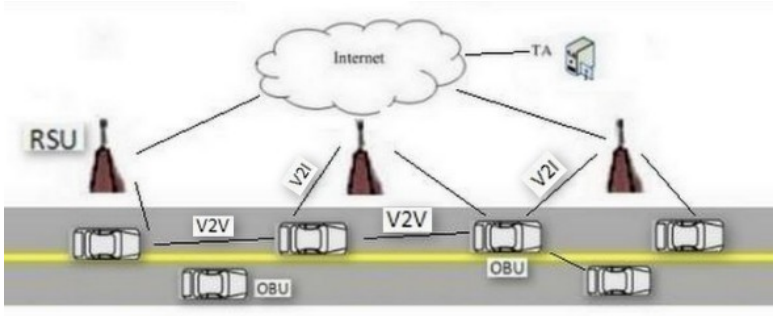

Figure 1.1: Architecture of VANETs

1.1.3 Difference between VANET & MANET

Apparently, VANET is a specific type of MENET [56]. RSU and OBU constitute of the ad hoc domain in which RSU is fixed node while OBU is mobile node. They are both based on 802.11p protocol. But actually, VANET and MANET have very different characteristics as follows:

1. In traditional MANET, energy is actually a bottleneck for the system. Some nodes are fixed in severe remote place, it is hard work to charge or change the battery for these nodes. By contrast, the nodes (OBU and RSU) in VANET have plenty of energy because they are attached to the vehicles. If the vehicle could work, they also have enough power to work.

2. VANETs are ephemeral networks [57]. The speed and direction of vehicles are unpredictable. In some extreme cases, when two cars are moving in opposite directions, the connection link between these two vehicles is very short. Hence, the real-time requirements should be considered when developing applications. In delay of a few seconds, a message which includes crucial safety-related information may become meaningless.

3. Since vehicles are fast-moving, the topology of network changes constantly. This is the most significant difference between VANET and MANET. In MANET, the nodes are fixed but they are highly mobile in VANET. Sometimes, the speed of vehicles is up to 150 kilometers per hour on the highway. On the other hand, the vehicle trajectories are on the roads which are already defined. The dissemination of messages could take this advantage. Another consequence is that the node density changes throughout a day: lower at night and higher during peak hours of the day.

4. The computation system of OBU is more complex and could implement more onerous calculations. Therefore, the distributed computation device should be designed to deal with more complex affairs regardless of the energy issue since these devices always have enough energy to work.

5. The size of VANET is much larger than that of MANET. The number of vehicles is in hundreds of thousands while the communication is mainly local. Therefore, we divide this network into small partition and design protocols and applications which are suitable for local area.

6. The mobility of a vehicle in VANET more or less depends on the layout of the road and the road condition. Vehicles cannot move around arbitrarily as in MANET, but use predefined roads. A vehicle's direction is unpredictable only at the road intersections.

All the above characteristics of VANET are different from traditional ad hoc networks. Consequently, the existing algorithms now-a-days applied in MANET are not suitable to VANET. Vehicular ad hoc networks have considerable challenges and require some new protocols.

1.1.4 Applications

The aim of VANET is to improve driving experience and the safety of transportation. By frequently broadcasting and receiving messages, vehicles are aware of their surrounding road situations. The drivers can react to events happen on the road by these messages in advance. Even in some urgent cases, the OBU can make the decision(stop or change the lane) automatically. There are many applications used in VANET, which are mainly classified in two parts: *safety-related* and *user application*.

Safety-related applications: This type of application can significantly decrease the number of accidents. Normally, vehicles travel at very high speed especially in the high ways. Drivers have very short time to react to the car crashes in front. If an accident occurs, the vehicles behind often crash before they stop on the highway.

Safety applications should give warning to drivers in advance, so that they can change the lanes or stop, thus avoiding accidents.

The application should supply driver the road situation and choose the best path for driver. This could prevent road congestion, saving people's time and fuel consumption. It can be noticed that the security for this type of application is mandatory; even a tiny mistake can cause serious consequences.

User applications: User application provides drivers with information (e.g., gas station), online-payment, entertainments and advertisements. For instance, if a driver wants to know the location of cheapest price gas station, he/she can get the information by sending request to the nearby RSU. After receiving this request, RSU checks from Internet and echoes it to the vehicle. Security is also required here but not as much important as in safety-related application.

1.2 Motivation

VANETs are a class of ephemeral networks [57], where the connection between vehicles (nodes) is short lived. The network topology changes frequently, as nodes move in and out of range of each other. The density of the network also changes over time, e.g. increases during rush hours. These characteristics make VANET very challenging for dealing with security issues. Some challenges are listed here.

Human behavioral tendencies are reflected in the movement of the vehicles (rational behavior). Vehicles can either start malfunctioning due to some internal failures and give out false alerts, false location and speed intentionally for selfish reasons. Malicious vehicles may also attempt to gather sensitive information about other users e.g. credit card number while interpreting RFID signals at an electronic toll station.

To protect privacy, vehicles do not use their unique identity (for example, the electronic license plate), but *pseudonyms*, when broadcasting data. Pseudonyms are generated and assigned in such a way that a vehicle's unique identity cannot be derived by observing two or more pseudonyms. Users can also authenticate themselves using pseudonyms. Current research on security in VANET has been focused on location privacy, maintaining authenticity of data and revocation of certificates and secret credentials. Surveys on the security challenges in VANETs can be found in [53, 39, 56, 16]. Most papers on location privacy deal with how to assign pseudonyms [57], when to change pseudonyms [57, 66, 27, 19], and how to assign signatures using pseudonyms [70]. Authentication techniques rely on signatures, such that a message is signed with a private key which can be verified if a user has the corresponding public key. A certificate is also issued which verifies the validity of the public key. Signature schemes for VANETs have been studied extensively, e.g. ECMV [76] and PASS [70].

Revocation of malicious nodes is another issue that has received a lot of attention. The issue is whether to maintain a list of all revoked certificates and keys or some seed of the revoked vehicles [70]. Revocation of certificates and secret credentials has a significant disadvantage: the certificate revocation list (CRL) containing all the certificates of revoked vehicles, has to be sent to all the nodes in the network. This approach requires large bandwidth, if the number of revoked nodes is high. However, revocation may not be necessary if a vehicle misbehaved only once for some selfish reason. In our work, we assume that nodes misbehave mostly because of selfish reasons, such as reaching their destinations faster and driving on a free lane. Vehicle might send false reports on congestion, accident or road block. It is conceivable to believe that a vehicle does not have malicious intentions of causing accidents, but normally sends valid and useful information.

If all the vehicle's certificates are revoked, then useful information sent will be ignored by other vehicles. We can argue that there is no need to classify vehicles according to their overall behavior, but instead one should be able to distinguish between correct and false information received from a vehicle. It is important to timely identify false data and the sender, because a delay of even one second might cause accidents. This problem is termed as *data-centric misbehavior detection* in contrast to entity-centric misbehavior detection, where the main goal is to find out and penalize a misbehaving node. The idea of data-centric misbehavior detection stems from Raya's work [57] on data-centric trust, where the author considers trust on information rather than on the source of information. Nodes which misbehave do not have to be revoked; instead, the misbehaving node can receive a fine depending upon its action. It can keep sending further information which might not necessarily be malicious. The payment of fines would hopefully discourage nodes from sending further false messages. Thus, it is important to detect false alert messages and false location information sent out by a node. Alerts like emergency breaking, approaching emergency vehicles, road feature notifications, change of lanes, etc, can still be detected.

Intrusion detection has been studied extensively in the context of wireless ad hoc networks. Existing solutions for detecting malicious behavior have limitations and cannot be applied to VANET. In a nutshell, the detection itself is application and scenario dependent. Trust management based solutions are not feasible because neighborhood may change rapidly and therefore trust relationships could be short lived and difficult to even establish in the first place. In relatively static neighborhood graphs (e.g. in congested areas) neighbors may not have history of misbehavior so the first violation cannot be automatically detected. Central authority may not be available to facilitate misbehavior detection and penalize accordingly. Some solutions

[81, 55] are based on countering Sybil attacks [23] in VANET. They assume that the precise location of nodes are known, and cannot detect false alerts raised by nodes. In Sybil attacks, nodes pose as separate identities and influence the decision of revocation and MDS, which rely on majority votes. Since our scheme does not rely on voting schemes and group associations, they are automatically resistant to Sybil attacks.

1.3 Thesis Objectives & Contributions

The objectives of this thesis are: (i) to understand limitations of VANET, (ii) to investigate why existing reputation and revocation schemes suggested for MANET do not work for VANET, and (iii) to propose solutions to improve the misbehavior detection scheme for VANET.

The contributions are the following:

1. We review the existing reputation and misbehavior detection schemes and point out their limitations. The existing trust management schemes for VANET are based on voting on the decision received by several vehicles which might not be required for practical scenarios. Most detection schemes are concerned with detection of malicious vehicles, the revocation requires revoke all certificates of malicious vehicles. Hence, huge bandwidth is needed to broadcast the revocation results to the whole network.

2. To improve voting efficiency in trust management, we introduce the concept of information cascading and oversampling to VANET. Information cascading and oversampling is the situation that later vehicles' decisions sometimes go along with the decision of early vehicle. We propose a novel mechanism to decrease

the effect of information cascading. We validate our voting mechanism using simulations and compare the performance with other existing schemes. The simulation results show that our scheme improves the correctness of voting. This result has been accepted for publication at IEEE PIMRC 2011 [36].

3. We consider that in most situations vehicles would send wrong information for selfish reasons. It is more important to detect false information than to identify malicious vehicles. A new data-centric detection model of VANET is presented in this thesis. Our scheme detects false messages and misbehaving vehicles by observing their actions after sending out alert messages. Meanwhile, we impose some fines to the detected vehicles instead of revoking the certificates of them. No voting and majority decisions is required in our misbehavior detection scheme, making it resilient to Sybil attack. The simulation results show that our model is very effective in misbehavior detection in a straight road. This part of work has been accepted by at the VTC conference 2011 [63]. The expanded version of this work [64] has been submitted to IEEE Transactions on Vehicular Technology.

1.4 Organization

This thesis is organized as follows:

- Chapter 2 is a background on the vehicular ad hoc network technology. This chapter presents the related work which has been done, it provides a comprehensive study on security, reputation and detection. A brief description of current issues and the corresponding solutions are described in this chapter. We discuss the limitations of reputation and detection systems.

- Chapter 3 introduces the concept of information cascading and oversampling, which commonly arise in social networks. A novel and simple algorithm is presented in this chapter to decrease the impact of information cascading. The simulation results and protocol comparison illustrate that our scheme is effective in VANET.

- In Chapter 4, we discuss that it is more important to identify whether the message is malign or not. Then we propose a new data-centric misbehavior detection model of VANET. The corresponding simulation results, analysis and limitation are in the remaining part of this chapter.

- Chapter 5 discusses conclusion and future work.

Chapter 2

Literature Review

Three aspects of VANET will be discussed in this chapter: security, reputation and detection. Some existing schemes are introduced and compared.

2.1 Security

Like other communication systems, the most important thing in VANET is security which is now attracting attention of many researchers. Security here has a few aspects: vehicles should have the ability to ensure the messages are not tampered by other vehicles; the vehicles cannot pretend to be someone else when they are misbehaving; the malicious vehicles and false messages must be detected and revoked by VANET automatically.

2.1.1 Attackers and Adversaries

In fact, we believe the majority of vehicle or driver is trusted. However, some malicious vehicles misbehave for selfish reasons. Understanding these reasons is responsible for researchers to design corresponding solutions to secure the network.

2.1.1.1 Adversaries

The nature of adversary will largely determine the scope of the defense required to secure a vehicular network [56]. The realistic evaluation for the vehicular environment defines two classes of adversaries as follows:

- Selfish drivers. Most drivers misbehave for selfish reasons. They do not want to share the lanes with other vehicles and try to maximize their driving profit. The common case is that a vehicle can tell vehicles behind it that "there is congestion ahead". So, if vehicles trust this message, they will choose another route and allow the greedy driver to use the clear lane to reach its destination.

- Malicious attackers. This kind of attacker is more harmful to the system and could bring more jeopardy to other drivers. These attackers could tamper the messages and give the wrong information deliberately. Meanwhile they could cheat the system to obtain more resources. In the worst case, malicious attackers attempt to sabotage the network by compromising the RSU.

2.1.1.2 Attacks

An adversary can sabotage the network in many ways.

- Message integrity: Making sure that the message cannot be tampered is significant. Sometimes the attacker tends to modify the message to benefit himself. This kind of action is not allowed in some safety-related cases, the wrong information against the actual event let the driver make wrong decision and consequently cause serious accidents.

- Cheating with position information: Position plays an important role in VANET; attackers in this case could transmit false position information. Even

sometimes the attacker can pretend to be another person by cloning the other's location.

- Denial of service(DoS) attack: The malicious attacker does not supply the service deliberately by jamming the channel. This behavior would cause serious consequences if the service is safety-related application.

- Non-repudiation: This happens in the following case: if a malicious vehicle sends out a wrong message to the others, after the system finding this malicious car and punishing, the driver could deny this message is sent from him. VANETs should ensure vehicle cannot refuse to acknowledge its message.

- Sybil attack: When a malicious vehicle creates a large number of identities and then act as a few vehicles, this is called Sybil attack. This malicious driver could force the others to believe that there are several vehicles. The other vehicle may understand there are several cars in front of him, while actually there is just one car that duplicates itself.

- Bogus message: The adversary could send the message whose contents do not correspond to the real road condition. For instance, one vehicle may send a fake congestion message to others to make use of the lane alone.

- Compromised RSU: RSUs are semi-trusted entities and could be compromised for a few reasons. In fact, the attackers would find it more difficult to compromise the RSU than OBU.

2.1.2 Security Requirements

Current research on security in VANET has been focused on location privacy, maintaining authenticity of data and revocation of certificates and secret credentials. Sur-

veys on the security challenges in VANET can be found in [39, 53, 56].

Authentication techniques rely on signatures, such that a message is signed with a private key which can be verified if a user has the corresponding public key. A certificate is also issued which verifies the validity of the public key. Revocation of malicious nodes is another issue that has received a lot of attention. The issue is whether to maintain a list of all revoked certificates and keys or revoked vehicles or some seed of the revoked vehicles [70]. Revocation of certificates and secret credentials has the following disadvantages. The certificate revocation list (CRL) containing all the certificates of revoked vehicles, has to be sent to all the nodes in the network. This approach requires a huge bandwidth, if the number of revoked nodes is high. To obtain a secure vehicular ad hoc network, we should satisfy the following requirements:

- Privacy: RSU and OBU should not be able to trace a vehicle's trajectory at a long time. The character of VANETs is each vehicle send out beacons or alert messages at certain intervals. This gives a chance to malicious attackers, such that they could trace the source of the messages through these beacons. This is really a big concern to the user's privacy.

- Authentication: In our life, we all validate a file or contract by our signature. In VANET, we borrow this idea to ensure the integrity and validation of messages. Authentication is that vehicle use its own private key sign the message before sending it. The other vehicles, after receiving this message, should check the key and certificate. Vehicle authenticates other vehicles and the messages send out from other vehicles. Messages from invalid vehicles will no longer be trusted by other vehicles.

- Availability: This requires that the network is available all the time. As we discussed above, in DoS attack, vehicles refuse to provide service. This is ab-

solutely not allowed in safety-related applications, which may need very fast response. Only a few seconds delay can render the information useless. We must have alternative scenario to ensure the availability.

- Efficiency: Vehicular ad hoc network is large scale but most of applications are local. In our real world, an big city may have a lot of vehicles but the alert message is only valuable in the vicinity of the event. Therefore, VANET is distributed system. Each vehicle runs the same application and just pays attention to its local environment.

- Robustness: The VANET system should continuously operate even in the situation when some parts of vehicles are out of work. The decentralized scheme makes sure each vehicle could work independently.

- Key distribution: Cryptography is commonly used in the communication system to ensure the integrity of the messages. A fundamental building block is key distribution which has several challenges in practical. First of all, the OBUs are manufactured by different companies, hence these companies should make a coordination at first. Secondly, when and how to change a key is the core problem in pseudonymous authentication system. At last, anonymous service needs a large number of identities, managing these identities is also not an easy task.

- Detection and revocation: When a vehicle misbehaves, the VANET system and other vehicles should have the abilities to detect the malign vehicle and then revoke the valid certificates of that vehicle. After this process, the revoked vehicle is invalid and other vehicles would not accept any message from this vehicle. There is a certificate revocation list(CRL),which includes all invalid

certificates and identities. This CRL is stored in each vehicle and everyone knows the malign vehicles. The drawback of this method is the size of CRL increases linearly when the number of vehicle is huge.

- Real-time constraints [58]: Vehicular ad hoc network is class of ephemeral network, at very high velocity in VANETs, strict time constraints should be considered. Messages should be received, verified and sent when required.

2.2 Authentication versus Privacy

In vehicular networks, we would like to distribute single identity to each vehicle at the beginning. This identity is unique and cannot be changed by drivers. Vehicle signs the message before they send out by this identity. A vehicle needs to authenticate other legitimate vehicles and messages sent out by other legitimate cars. Meanwhile vehicles only accept valid messages from valid cars. By this way, vehicle cannot deny the message which is sending by itself because the message includes its identity; the system could detect and revoke misbehavior vehicles through their unique identity. Also, this method prevents Sybil attack since each vehicle only has one identity.

However, on the other side, this method really has a big drawback for attacker. The messages that are send out by vehicles, contain information about position, time and event as well as the identity of source vehicle. For the attackers who want to trace somebody, the only thing they need to do is to gather the messages from the target after filtering other messages by the target'os identity. For instance, suppose one vehicle presents the message at time T_1, location L_1, and presents another message at next interval T_2,location L_2, the malicious vehicle would easily learn the target vehicle moves from L_1 to L_2. Therefore, the trajectory of target vehicle is tracked. This is really a big offense to user's privacy.

To address the tension between authentication and privacy, researchers came up with an idea of anonymous service [56]. This concept allows vehicle use anonymous identity when they supply applications, this more or less resolve some tension between authentication and privacy. Vehicle in anonymous service network is given an unique identity and several pseudonyms, it would never use its real identity to sign the message, instead of that, each vehicle use pseudonym to sign the message. Meanwhile, each vehicle changes its pseudonym identity over time. Since vehicle has no fixed identity to sign the message, therefore for the attacker, it's hard to track someone. For instance, suppose one vehicle presents the message at time T_1,location L_1, and presents another message at next interval T_2,location L_2, the identities this vehicle used for these two message are different, for malicious attacker, it is impossible to know these two identities are from one same car when the number of vehicle is huge and identities-changing speed is high.

In recent days, two methods always been used for anonymous service: *pseudonymous authentication scheme* and *group signature*. Both of them can address the problem of authentication and privacy. In pseudonymous scheme, each vehicle stores a set of identities while in group signature scheme, vehicle is issued a group private key, with which it signs the message.

2.2.1 Pseudonymous Authentication

The basic idea of pseudonymous authentication scheme is that each vehicle stores a large number of pseudonymous certificates at first and then chooses one of these certificates to act as its identity at one time randomly. As we know from above, TA (trusted authority) has sufficient storage and cannot be compromised; therefore, it is safe and feasible for TA to store the pseudonymous certificates. When vehicles first

register, TA distributes enough pseudonymous certificates to each vehicle, and at the same time, each vehicle also gets an unique permanent identity.

For privacy consideration, vehicles never use this permanent identity to sign the message; they randomly choose one of pseudonymous certificates TA has issued for digital signature. By this way, the temporary identity of each vehicle changes over time, and malicious attacker can hardly trace a specific vehicle because after altering the certificate the attacker cannot link this new certificate with the old certificate, which means that the attacker has lost the target. However, this method still has some problems. The cost of revocation is expensive. When a vehicle is revoked, the number of pseudonymous certificates which will be added to the CRL could be 43,800 [70], and the size of certificate revocation list(CRL) increases quickly if the size of network is huge.

Hubaux et al.[37] first introduced pseudonymous to VANET. This method since then has been used by many researchers [45, 38, 77]. Most papers on location privacy deal with how to assign pseudonyms [57], when to change pseudonyms [19, 27], and how to assign signatures using pseudonyms [70]. Lu et al. in [45] proposed ECPP protocol to address the issue of anonymous authentication for safety messages. Based on on-the-fly short time key generation between OBU and RSU, ECPP increases the authentication speed and minimizes the required storage for keys. Jiang et al.[38] proposed a scheme called *Binary Authentication Tree*. In their scheme, a binary tree of the received signatures can be built to efficiently eliminate the performance bottleneck due to the significantly reduced computational overhead. In [77], Wasef et al. proposd an efficient distributed certificate service (DCS) scheme to ensure OBU could update its certificate from any RSU, no matter whether this OBU is in the same domain where it is registered. The DCS scheme can rapidly certify a large number of certificates and signatures.

Sun et al.[70] propose Pseudonymous Authentication Scheme with Strong privacy preservation (PASS) which allows vehicles to update their certificate set from RSU on the road. Compared with the previous pseudonymous schemes, PASS can decrease the revocation overhead and certificate updating overhead.

Pseudonyms are changed from time to time to preserve location privacy. In [28], the authors point out that the pseudonyms should be changed only in the mix zones. Mix zones are areas where nodes cannot be observed, either by another node or by a RSU. The problem with this approach is that if there is only one node in a mix-zone and it changes its pseudonym, then it is clear that the two pseudonyms belong to the same node. However, if there are more than one vehicle in the mix zone and they change their pseudonym, then it cannot be easily predicted which pseudonym corresponds to which node. Buttyan et al.[19] show by simulation, how the privacy level is changed using the above approach. Frequent change of pseudonyms ensure higher privacy, but pseudonyms are expensive and often obtained from a central authority.

Freudiger et al. [27] give a detailed study about the age of pseudonyms and discussed different parameters, on which the age of pseudonym depends. Sampigethaya et al. [65, 66] used random silent period between the change of pseudonyms. They assumed that vehicles move in a group with similar speed. When a new vehicle joins the network with some pseudonym, it waits for a random time before changing its pseudonym. For example, if two nodes enter the network at the same time and change their pseudonym after a random time interval, then the new and old pseudonyms of a vehicle cannot be linked.

2.2.2 Group signature

Another alternative for providing anonymity in vehicle networks is the use of group signature [20, 68, 33]. A group signature scheme allows every member of the group to sign a message without revealing the exact identity, while other members of the group have the ability to verify that the message originated from a group member. Group signatures were introduced by Chaum and van Heyst [21] to provide anonymity to the signers. Boneh et al.[18] suggested the use of group signatures in vehicular networks.

A group signature has two aspects: *Group manager* and *Group member*. Each group has one group manager which is in charge of the key distribution, adding group member and detecting and revoking group member. Each group member only signs the message by group user key which is issued by manager, and others never know the exact identity of the sender. At first, group manager issues different group user key to each group member, then issues group public key to all members of the group. The members use their own group user key to sign the messages and use group public key to verify that the message is from a group member. Only the group manager knows the real identity of each member, thus it could detect and revoke the group members. Group signatures can be applied to sign messages in VANET; when another vehicle receives the message, it can only check the authenticity of the message, with no means to track the node who sent it. Although these schemes provide authentication, conditional anonymity and non-repudiation, they result in large revocation cost. Since groups can change very frequently in a city network, vehicles can join and leave groups very rapidly; therefore this scheme is not so practical.

Studer et al.[68] proposed a scheme based on group signature. They examine a VANET key management scheme based on Temporary Anonymous Certificate Keys(TACKs). They give some valid assumptions and discuss the efficiency of their

scheme. However, some issues in VANET still remain in their TACKs; the detection and revocation of temporary key are restricted by the expiration scheme. Also, the correlation attack happens in the following situation: if there is only one single OBU changing the keys at a time, an adversary can associate the new key with the old key of this OBU, and by this way the adversary can compute the exact identity of OBU.

Calandriello et al.[20] introduced a hybrid scheme of the combination of pseudonym with group signature. Each vehicle is equipped with a set of pseudonyms and a secret group signing key, that it can issue itself a certificate by group key. The authors shows by this way, the computational overhead of authentication decreases.

2.3 Reputation

In VANET, an important issue is that how to trust the specific vehicle or message. For instance, if a car sends the message that there is congestion at location X, should other vehicles believe this car as well as this message and then make a corresponding action? Researchers come up with the method which called reputation system to solve this issue. We will discuss the work related to reputation system.

2.3.1 What is Reputation?

In ad hoc networks, nodes are both terminals and routers for the lack of routing infrastructure, they have to cooperate with each to exchange information. Misbehavior means deviation from regular action, node could misbehave for selfish reasons and consequently impact the system. The aim of reputation system is to construct trust value for each node in this network, and based on these values, the other nodes decide whom to trust thus encourage trustworthy behavior. Resnick and Zeckhauser [62] list three goals for reputation systems:

- To provide information to distinguish between a trustworthy peer and an untrustworthy peer.
- To encourage peers to act in a trustworthy manner.
- To discourage untrustworthy peers from participating in the service, the reputation mechanism is presented.

In recent years, reputation-based system is abundantly studied. It provides input for computational trust as predictions on future behavior based on the node's past actions. The node who have transacted with the specific node could get information about this node's actions and then construct the reputation value for this node. Usually the range from 0 to 1 are introduced to imply the degree of trustworthiness, 1 represents the node is trustful while 0 means the peer is unreliable.

2.3.2 Reputation-based System Model

Reputation system is highly similar with our credit in society. The work on reputation modeling in mobile ad hoc networks is mainly in three sets: 1) *reputation establishment*, which gathers trust information of nodes based on their past behaviors; 2) *trust calculation*, in which some mathematic functions are used to calculate the trust values of nodes based on the input of gathered reputation information; 3) *decision making*, in this step, node decides whether to trust the specific entity or not.

In *reputation establishment* phase, node gathers enough information for a specific target node through past interactions between them. If the past behaviors of this specific node are malicious then its reputation is low. However there is a possibility that interactions between node and the target node never happened, in this situation, the reputation information of the target node will be gathered from the nodes who

has communicated with it in the whole network. During the second phase *trust calculation*, each node uses existing mechanism to compute the trust value by input of reputation information obtained in first phase. This computation function depends upon the application.

At last, based on the trust value, nodes determine whether to believe the target node or not. This step requires a threshold to make the decision. Trust value overtaking threshold means this node is trustful while a lower value represents unreliable.

2.3.3 Reputation System in MANET

Reputation was firstly used in Internet and then spread to mobile ad hoc network. Due to the characteristics of MANET, [12, 26, 69] give some main features of trust:

- The trusted centralized certification authority cannot be assumed, hence the decision method should be distributed. In other words, decentralized mechanism is required.
- The nodes in MANET are not always cooperative. In restricted environment, nodes may refuse to cooperate in order to save the energy or for some selfish reasons.
- Trust is not necessarily transitive. A trusts B and B trusts does not means A trusts C. Each node makes its decision by itself.
- Gathering reputation information from past relationship is high computational, researchers should minimize the excessive calculation to decrease the workload of the network.

Researchers have done large amount work on reputation systems in MANET. To learn these methods we discuss a part of existing reputation systems. Wang [72]

proposed a trust model which could be used in routing for MANET. In their system, each node is assigned an initial trust value and updates trust level by the reports from detection tools. These reports are generated from the neighbors of the target node by monitoring target's behavior. The source node uses these trust levels to select the best reliable path to destination.

Zouridaki et al.[82] introduced a trust scheme, called HERMES, with the objective of reliable delivery and routing. The authors apply Bayesian approach to compute the trustworthiness between two neighboring nodes based on the set of first-hand observation of packet forwarding. Source node chooses the best route by computing the weight of the path through a set of peer-to-peer trust between the peers along the path.

Marti et al.[46] proposed a reputation system for ad hoc networks where a node monitors the transmission of a neighbor to make sure that the neighbor forwards other traffic. In essence, one can consider such a reputation system as a repeated game whose objective is to stimulate cooperation. Michiardi et al. [47] proposed a mechanism, called CORE, to enforce node cooperation based on a collaborative monitoring technique. Each network entity in CORE keeps track of the other entity's collaboration using a reputation scheme.

2.3.4 Reputation in VANET

Vehicular networks ensure that the information received from any vehicle is promptly and correctly propagated to nearby vehicles. A crucial point is how to trust the information transmitted especially in life-critical situations because any tiny mistake can cause serious consequence. Therefore, an effective reputation system for VANET is urgently required.

However, due to the differences from MANET, the requirement for reputation system in VANET also differs from that in MANET. The following characteristics indicate that the existing reputation systems for MANET are not suitable for VANET:

- VANET is an ephemeral network in which cars are constantly roaming around and is highly dynamic. The velocity on a highway is up to 150 kilometers per hour. At this high speed, the contact and reaction time is too short to build trust among themselves.

- VANET is a large scale network where the number of nodes and the density is much higher than that in MANET. Network traffic can be very high. Hence, there should be intelligent vehicle communication systems that are scalable and can prevent data congestion by deciding which peers to interact [79] with.

- Almost all the existing reputation systems compute trust value based on the past interaction with target node. However, this assumption is not valid in VANET due to the dynamic and open environment. In fact, if a vehicle is communicating with another vehicle, it is not guaranteed that it will interact with the same vehicle in the future. Therefore, the existing algorithm which is based on long-term relationship is not suitable for VANET.

- Some of the reputation models [80] depend on a central entity to gather the information. However, the large number of nodes and high dynamic environment require a decentralized system in VANET. Aggregation should be a local action instead of a global action.

- Another important issue is pseudonymous authentication which is ignored by many researchers. As discussed above, for privacy consideration, each vehicle is issued a large number of identities, and only CA knows the relation between real

ID with pseudonymous identities. Therefore, it is infeasible to get meaningful reputation values because vehicles change identity over time. For instance, if vehicle A interacts with B then A has a reputation value for B, but at next moment if B changes it pseudonyms to C, the reputation value stored in A is no longer useful.

2.3.5 Trust Models in VANET

Only a few trust models have been proposed for honest information sharing in VANET. Based on the characteristics discussed above, there are two basic models: *entity-oriented trust model* and *information-oriented trust model*. Entity-oriented trust model focuses on the trust of vehicles that means construct trust values for vehicles and determine whether to believe the vehicles or not, whereas information-oriented trust model normally decides how to trust the message transmitted. Existing information-oriented models consider each vehicle is equal. As a result, majority voting is widely used to judge the correctness of the message.

2.3.5.1 Entity-oriented Model

As we discussed above, reputation of vehicles is not easily built in this ephemeral environment. Two typical entity-oriented trust models are proposed by Gerlach [29] and Minhaset et al.[48]. Gerlach in [29] proposed a sociological trust model based on the principle of trust and confidence tagging. Their trust establishment service is tagging the content of the database with confidence value which may be based on certification or self-tests of the system. Then, they describe how to choose a confidence value. Meanwhile, the authors also propose an architecture for communication and a model for privacy.

Minhas et al.[48] developed a multi-faceted trust modeling framework which incorporates role-based trust, experience-based trust and majority-based trust to receive the most effective reports. Meanwhile, they describe the algorithm that shows how to integrate various trusts. This model allows the vehicle to actively inquire an event by sending requests to other vehicles. The above two schemes are both based on the trust value of vehicles.

2.3.5.2 Information-oriented Model

In contrast to the traditional entity-oriented scheme, information-oriented trust is some different. As we know, VANET is ephemeral and highly dynamic ad hoc network, if a vehicle connecting with one another, it is not guaranteed to interact with the same vehicle in the future. More importantly, pseudonymous identities are used for privacy consideration, this aspect enforces vehicle change ID over time and consequently the previous relationship will no longer valuable. Furthermore, the relationship evolved slowly with time: they last a long interval and change after lengthy operations[61]. Therefore, entity-oriented model is not easily applied to vehicle ad hoc network.

To solve this issue, more focus has been placed on evaluating the quality of information sent by vehicles. In other words, we no longer construct our trust based on vehicles but on message itself. In these models, long-term relationships between vehicles are not required which is the basis in entity-oriented model. Hence, *information-oriented trust* is more efficient in this life-critical network because it decrease the delay of relationship processing and computing.

One of the first papers on reputation in VANET is a Vehicle Ad Hoc Reputation System (VARS) [24]. In that paper, the authors propose a modular reputation system architecture that strictly separates direct and indirect reputation handling from

opinion generation. VARS is not based on the behavior of nodes but on the opinion about distributed content. Receivers can evaluate the opinion of other nodes and use it as a basis for their own decision about the trustworthiness of a message. This scheme has some limitations pointed out by the authors. One possible problem is the overhead added to the package by appending the opinion of all intermediate nodes.

In [40], authors introduced VSRP, an algorithm based on reputation that uses trust values assigned to nodes to detect as well as eliminate malicious nodes from the network. In fact, this paper is mainly on the detection of malicious nodes. The simulation results are very effective and suitable for mobile ad hoc network. But it has some issues: 1) This algorithm is based on the detection range of sensors, but the range is only 50m which is inefficient. In the case of congestion, if the distance between one specific vehicle and congestion location is more than 50m, which means this vehicle may not believe there is congestion because its sensor cannot detect this situation. Therefore this car will not choose another path to avoid the congestion, instead, it joins the waiting queue. The driving efficiency is not improved; 2) Also, this algorithm is too much to rely on detection sensors so that if vehicle's sensor stops working, this vehicle may reject all messages from other vehicles.

In [44], Nai-Wei Lo et al. introduced a dynamic reputation system to prevent the spread of false warning messages based on a new mechanism. In this paper, the authors list four functions to compute the confidence threshold and trust threshold. However, some issues still remain: 1) Vehicle changes ID over time. The authors did not mention how to guarantee that the ID of a vehicle does not change during an event; 2) The result is highly influenced by the speed of vehicles. For instance, faster the vehicle speed, less time is given to detect a specific event. If threshold is constant, higher speed will decrease the accuracy.

Raya et al.[61] indicated that *information-oriented* trust maybe more appropriate in ephemeral ad hoc network. In this paper, author firstly develop the notion of data-centric trust: data trustworthiness should be primarily to data itself rather than being merely a reflection of the trust of data reporting-entities (vehicles). They apply Bayesian and Dempster-Shafer[22] theory to address ephemeral networks that very demanding in terms of processing speed. They also evaluate the data reports with corresponding trust values by weighted voting,Bayesian inference and Dempster-Shafer theory. The simulation results show us that these methods are suitable and effective enough for a life-critical vehicular ad hoc. The shortcoming of their model is the trust is established on the specific data, it needs to construct trust value again and again for each event.

Timo Kosch [41] introduced the concept of local danger warning prototype based on an ad hoc communication network. Ostermaier et al.[52] extended Timo's concept and then they introduced a special active safety application. A security mechanism based on information-centric evaluation of the message is proposed. The author has developed four decision methods which are based on voting scheme. The simulation result shows the performance of these voting schemes for local danger warning message in VANET are simple and also effective. But there are still some shortcomings: This method cannot detect malicious nodes which mean attacker can misbehave over time. Also Sybil attack is exists in the system and we cannot prevent it. Some methods are based on threshold, but in case of sparse vehicles (less than threshold), these schemes are ineffective and increase the incorrect decisions.

2.4 Detection and Revocation

Another issue in vehicular ad hoc network security is detection and revocation of misbehaving vehicles which has recently received a lot of attention. For example, a crashed vehicle may send a Post Crash Notification (PCN) message when a car crash happening, while a malicious vehicle may raise the PCN alert even in the absence of a crash, or report the false position. Note that the authentication mechanism itself only guarantees the message integrity but cannot ensure the content of message is correct. Therefore when a vehicle misbehaves, like modification of messages, giving bogus information to others, attacking the network by pretending to be another one, the network should have the ability to detect these false messages and malicious vehicles and then revoke them out through some schemes.

Tiranuch Anantvalee et al.[13] introduced the definition of intrusion detection. They say the computer and network monitor activities in the system, collect activity information and then analyze it to judge whether or not an activity violates the rules. Once an Intrusion Detection System finds an activity is unusual or is known to be an attacker, it then generates alert messages and informs other nodes. Meanwhile, network revokes misbehavior nodes out of the system. The detection and revocation procedure is the following:

1. Vehicles send and receive messages as usual.

2. If there is an event, malicious vehicle A sends a message about this event. Other vehicles who receive this message will detect whether it is true or false by existing detecting and reputation methods.

3. If A's message is detected as bogus, other vehicles will report A's present pseudonym to RSU or CA.

4. RSU or CA knows vehicle A's real identity as well as all its pseudonyms. As a result, CA creates a revocation list which includes all A's pseudonyms and broadcast it to the network.

5. All vehicles store this revocation list and then they know A's message is unreliable. Consequently, vehicles will not accept all messages from A. This means A is revoked out of the system even if it still traveling on the road.

The problems in this procedure are mainly the two issues: *detecting the misbehavior* and *distributing the revocation list*. Vehicle runs some misbehavior detection scheme (MDS) to detect misbehavior and report the malicious vehicle to CA. How to improve the efficiency of the MDS has attracted a lot of attention. Meanwhile, the communication overhead is a big issue when distributing the revocation list. Extensive research has been done to decrease the size of revocation list. In some revocation schemes, revocation list is no longer used.

2.4.1 Detection of Misbehavior

Intrusion detection is studied extensively in MANET. Generally speaking, detection is related with reputation system. If vehicle wants to judge a message is correct or not, it should first gather enough information from others or by its own detection device, and then make the decision to determine the validity of the message. From this point of view, misbehavior detection is similar to the reputation system we have discussed in the previous chapter.

There are several detection schemes for MANET in [13]. Tiranuch classifies an intrusion detection system (IDS) into many subclasses according to their architectures and detection techniques. IDSs are of different types: distributed and cooperative

IDS, local IDS, distributed IDS system using multiple sensors and dynamic hierarchical IDS.

Golle et al.[31] proposed a general approach to evaluating the validity of VANET data. In their approach, each vehicle maintains a model containing all the knowledge of possible events and actions in VANET. Each node equips sensor, the sensor data is collected by vehicles and propagated to the neighboring region. The node tests the validity of data it received from other nodes by checking it with the model, if the data agrees with the model, then it is considered as a valid message. Meanwhile, each vehicle has an adversarial model, which is used to search for explanations of errors, ranking explanations and using the best explanation to correct the errors when an invalid message is detected. The main problem of this scheme is in such a large network as VANET, maintaining a model is expensive. Also we hardly construct the whole adversarial model at beginning, new adversary might join. No mechanism to update this model has been mentioned in the paper. At last, the authors mention that there is a trade off between privacy and the ability to detect and correct the malicious data: frequently changing keys increases privacy but decrease information to detect and correct malicious data. Hence, for their scheme, privacy is sacrificed for the ability of detection and correction.

One big issue requires us to solve which is called Sybil Attack [51]. In Sybil attack, a malicious vehicle can easily create several false identities and pose as multiple vehicles. False information reported by a Sybil node will be convincing to the rest of the network because this vehicle appears as several vehicles who agree with this information. For instance, if a vehicle finds traffic congestion and reduces its speed significantly, this vehicle will broadcast alert message to the following cars. Recipients will relay the message to vehicles further behind. However, the Sybil recipients could create some illusions and convince following vehicles that there's no congestion. In

this way, the malicious Sybil vehicle can cause a massive pileup, potentially leading to serious consequences. Identity authentication does not prevent Sybil attacks in VANET. There have been other papers that deal with Sybil attacks in VANET. Guette et al.[32] have proposed some methods to deal with this issue: use of public key cryptography, assuming a given propagation model and secure positioning.

Park et al.[55] proposed an approach called timestamp series to defend the Sybil attack by the user or RSU. A vehicle gets the timestamps signed by RSU when it passes the RSU, and the message sent out by vehicle should include a series of recently obtained timestamps. The authors consider that due to the differences of dynamics among vehicles, it is impossible for two vehicles passing by RSUs at the same time. The Sybil attack could be detected when vehicles receiving messages with same timestamps. This scheme is not efficient and has a flaw in the sense that a malicious node might change the timestamps and send different timestamps for different identities that it has faked and avoid being detected as a Sybil node.

The authors in [81] defeat Sybil attack by public key cryptography. They present a Sybil detection scheme called privacy-preserving detection of abuses of pseudonyms. This scheme has two steps: system initialization and attack detection. During system initialization, CA generates a sufficient number of pseudonyms for each vehicle. There are two hash functions which are called coarse grained and fine grained respectively. Each pseudonym has a fine grained hash value and a coarse grained value. The fine grained hash value of all pseudonyms of a vehicle are the same while any two fine grained hash value of pseudonym belonging to different vehicles are different. In detection phase, after receiving two pseudonyms, RSU computes the coarse grained hash value and sends these values to CA, which checks the fine grained values to see if the same message was send by a vehicle with multiple entities or from different

vehicles. By this way, Sybil attack is detected. However, the flaw of their scheme is the majority computation is finished by RSU which requires RSU cover the entire roads, this is not an infeasible assumption. Also, the authors did not mention how to deal with compromised RSU which will cause a serious result.

Guette et al.[32] proposed a precise method to detect Sybil attacks by taking into account transmission signal attenuation and the kind of receiving antenna (omni- or bi-directional). They evaluate four cases: no attenuation and standard transmission power, no attenuation and low transmission power, no attenuation and high transmission power and propagation with attenuation. The last case is a real propagation environment where they consider attenuation factor α as 1. But as authors have pointed out, such attenuation depends on different parameters. The signal loss in sparse place and urban site where standing lots of skyscrapers is fiercely different. Therefore, their scheme could only deduce the bound but not exact position of the area that Sybil nodes existing.

Bin Xiao et al.[78] present a lightweight scheme for Sybil attack detection by signal strength distribution. Authors divide vehicles playing three roles: claimer, witness and verifier. Claimer is the vehicle which periodically broadcasts a beacon to neighbors; witnesses are vehicles in the communication range of the claimer; verifier computes the estimated position of claimer. Each vehicle would periodically play all these roles. The overall detection process includes three phases: Phase 1: witness vehicles save the corresponding signal strength measurements for each received beacons in its memory. Phase 2: when verifier gathers enough signal strength measurements for a neighboring vehicle, it computes the position of claimer. Phase 3: if claimer is detected as a Sybil vehicle, verifier uses some schemes to find all potential Sybil vehicles originating from the same malicious physical vehicle. Their scheme also operates

even malicious vehicle increase or decrease the signal power. However, the unsolved issue is how densely RSU can be deployed for better performance since this method is based on the help of RSU.

One issue of Bin Xiao's scheme [78] is that the system relies on gathering information from neighbors, and therefore the reporting delay is inevitable. Ghosh et al. [30] proposed a detection scheme to determine the root cause of a misbehavior. They address post-crash notification (PCN), where vehicles send false information about a PCN alert after an accident has taken place. The malicious vehicle might either send a crash alert even if there is no crash and not send a crash alert even if there actually exists a crash. The authors consider two situations: false alert (alert is raised even there is no crash) and true alert (there is a crash) with false crash position. Their analysis is based on the deviation of the actual trajectory from the expected trajectory. The expected trajectory is calculated and compared against the sensed trajectory. Depending upon the deviation from the expected values, the type of misbehavior can be detected. However, this scheme has a problem which is not solved that the change of pseudonyms during the estimation of the deviations might affect the results, but if the pseudonyms are not changed regularly then the privacy cannot be guaranteed.

Raya et al.[60] proposed a scheme to detect and revoke malicious vehicles. Each vehicle has a few pseudonyms, and corresponding to each pseudonym there is one public/private key pair and a certificate issued by CA. Their detection scheme has two parts: misbehaving detection scheme (MDS) and local eviction of attackers by voting evaluators ($LEAVE$). In MDS phase, each vehicle uses its own sensory inputs, messages from neighbors to classify safety message from a given vehicle as faulty or correct. Entropy method is used by author to measure the deviation between expected

values and actual metric values. The given vehicle is detected as malicious node when entropy is over the threshold. If vehicle detects an attacker, it broadcasts warning messages to all vehicles in range. The accusations to the attacker are collected by the $LEAVE$ protocol. If the number of accusations is above a certain threshold, $LEAVE$ protocol will evict the certificates of this vehicle. However, the above two schemes operate based on the assumption that the majority of neighbor vehicles is honest, which is feasible but sometimes cannot be achieved in the situation of sparse density of vehicles.

Vulimiri et al.[71] proposed a scheme to detect the misbehaviors by use of secondary information. For example, if a vehicle triggers a Post Crash Notification (PCN) alert, it may cause other vehicles nearby to slow down or stop. Consequently, these vehicles will trigger a Slow or Stopped Vehicle Advisory (SVA) message to be transmitted. Therefore the receipt of a certain number of SVA messages following the receipt of PCN could strengthen a vehicle's belief to the trust of event. However, there are two issues the authors did not consider. Firstly, this scheme will not work in the situation of no vehicle following the first car which transmits the PCN alert. Secondly, if there are indeed some vehicles nearby the first car, they could be malicious vehicles who would not transmit SVA deliberately, and this situation is highly likely to happen.

2.4.2 Revocation of Vehicle

Revocation of malicious nodes is another issue that has received lots of attention. Revocation of vehicles means revocation of their certificates. The use of pseudonyms implies that the malicious vehicles' certificates must all be revoked. Revocation of certificates has the following disadvantages: the certificate revocation list (CRL) con-

taining all the certificates of revoked vehicles, has to be sent to all the nodes in the network. This approach requires a huge bandwidth if the number of revoked vehicles is high. However, this distribution process takes time to all remaining vehicles in the whole network. During this interval, the attacks could still jeopardize other driver's safety. The existing revocation schemes are mainly in two types: *local revocation* and *global revocation*.

Local revocation uses local voting mechanism to identify and revoke a malicious vehicle. Two requirements must hold: *the majority is honest* and *vehicles are able to detect*. The viewpoint of Liu [43] is that these two requirements are demanding. Most of the honest nodes may be unable to vote due to lack of detection ability, e.g. out of detection range. Also, there exists Sybil attack, which can affect the voting result. In *global revocation*, CA identifies the accused vehicle and determines whether to revoke it by the use of trust management. If one vehicle is judged as a misbehaving node, all its certificates are invalidated in the entire network. However, the issues for global revocation scheme are that CA is not always available and the latency may be unacceptable, which is crucial in VANET.

One of the most difficult challenges of revocation is balancing security and privacy. On one hand, considering the privacy, pseudonym is applied; on the other hand, if each vehicle has a large number of certificates, detecting and excluding a node is much more complicated since the revoked node can easily switch to another pseudonym.

Raya et al.[60] proposed two revocation schemes, called revocation of the trusted component (RTC) and revocation using compressed certificate revocation lists (RC^2RL) respectively. RTC means that if CA determines which vehicle must be revoked, CA instructs the vehicle to erase all cryptographic materials it stores and halts its operation upon completion of the protocol. However, as the authors pointed

out, RTC is not robust enough because the adversary could control the communication link between CA and vehicle. In terms of decreasing the size of CRL, they proposed the RC^2RL protocol by the use of Bloom filter [17] compression. Then, they proposed a LEAVE protocol which evicts the certificates of a node if the number of accusations is above certain threshold.

Moore et al.[49] proposed several strategies to remove compromised devices from ad hoc networks. The authors modified the existing suicide scheme and called it *Stinger* which deviates from suicide in the following aspects: both the accused and accuser still receive safety instructions from other vehicles in Stinger, and the good accuser could still accuse other bad vehicles next time. Stinger works as follows: when a benign node G detects a bad node B, it broadcasts a sting (G, B). All nodes near G blacklist both the nodes B and G. Any message from B and G is disregarded. However, they can still receive and forward messages. When the nodes are within the reach of RSU, they are evicted altogether.

Moore et al. [50] compared the security and performance properties of $LEAVE$ and Stinger by varying attacker capabilities and traffic conditions. Stinger is faster than $LEAVE$ and scales better with growing traffic density. However, $LEAVE$ has a lower false positive rate. This means that it revokes fewer benign nodes than Stinger. At last, author devised a hybrid scheme allowing vehicles choose Stinger or $LEAVE$ that best suits the circumstances.

Raya et al. [59] also use game theory to analyze vehicle actions and corresponding cost. In terms of two existing revocation schemes: voting and suicide, Raya et al. point out that most of the users will prefer avoiding the contribution to revoke the misbehaving nodes notably due to the potential cost of the revocation procedure. They define three possible strategies for the revocation of the attacker: abstain

from the local revocation, participate in a local revocation by voting procedure and play self-suicide to declare the invalidity of both their identities and their attacker's identities. Then they proposed a game-theoretic approach called *RevoGame* which considers the cost when making a revocation decision to choose the best strategy for each individual vehicle. Raya's work is to decrease the total cost of the network. Their simulation results illustrate that self-suicide strategy is more efficient but more costly, while voting mechanism avoids the revocation of benign vehicles when the percentage of attackers is small. However in [43], Liu discussed the limitation of Raya's scheme. In Raya's model, each vehicle's action is based on the previous player's action and the anticipated actions of the following players. This means that each vehicle has the correct knowledge about the eligible voters around which is infeasible to achieve.

Based on Raya's work, Bilogrevic et al. [15] proposed an improved and extended local certificate revocation framework for ephemeral network which includes VANET. They provide incentives to guarantee the revocation of malicious nodes when they collude or when the parameter estimations are difficult. Then, they allow for personalized and dynamic costs based on the past behavior reputation. Since each node could have a different reputation, they establish on-the-fly Nash equilibrium to minimize the social cost of the revocation.

Meanwhile, some work has been done to decrease the size of CRL in order to reduce the network traffic during the distribution phase. Papadimitratos in [54] described a novel scheme that divides the CRL into several pieces and then broadcast it to vehicles by the use of Fountain or Erasure codes, after receiving a certain number of pieces each vehicle could construct the CRL. Also in their scheme, the collaboration between regional CAs make CRLs only contains regional revocation information and consequently their size is kept low. The simulation results show all vehicles can

obtain the latest CRL very fast. Laberteaux et al.[42] distribute the CRL by using vehicles in an epidemic fashion instead of broadcasting from RSUs. Haas et al.[34] present mechanisms that achieve the goal to reduce the size of CRL, and they suggest optimizations for organizing, storing and exchanging CRL. They group together all of one vehicle's certificate identifiers through the use of a single key so that a vehicle's multiple certificates could be used by a single value. Bloom filter [17] is used to store the certificate identifiers, and then they broadcast CRL updates instead of the whole CRL. The sender only sends the revocation information which the receiver does not have. For instance, if the total number of identities in CRL is 120 while the receiver has 90, the sender will only send the remaining 30 identities to the receiver. Recently, Sun et al.[70] proposed an authentication scheme in which the CRL size is in the order of the revoked vehicles and does not depend on the number of pseudonyms that the revoked vehicles have.

Chapter 3

Information Cascading and Oversampling

In this chapter, we introduce the concept of information cascading and oversampling and then identify the negative impact on voting mechanism. A comparison of three different voting schemes are given to show the improvement of our simple voting scheme.

3.1 Concept of Cascading and Oversampling

Due to safety concerns, it is more important to know the correctness of the data rather than the authenticity of the nodes. As we discussed in Chapter 2, most trust management schemes in MANETs rely on voting which is similar in VANET. Simple voting decisions lead to two major problems in ad hoc networks, which to the best of our knowledge has not been addressed in literature. These two problems arise in social networks and are known as *information cascading* [25, Chapter 16] and *oversampling*[11].

Information cascading is that there is a decision to be made and people decide sequentially after observing the behavior of others. Then, their decision is highly influenced by the previous decisions and might overrule its own observation. The instance for information cascading is: sometimes the customer gives some comments after they take a dinner in a restaurant. It is highly possible that the other customers will be influenced by these comments, and later responders sometimes go along with the decision of early responders[2].

In ad hoc network, information cascading and oversampling occur in the following situation: Node A receives the opinion of nodes B and C. It might be possible that B's opinion is influenced by C. So we say that the opinion of C has been oversampled [3]. In such cases, there is a need to discount the opinion of B, so that C's decision is not oversampled.

We observe that this situation arises commonly in VANET. Suppose a node A receives the information of an event, "congestion" (say) from a node B and node C (where C's decision is obtained from B) and "no congestion" from node D and F. If the node decide what to do, and do a majority voting, then it receives two votes in both favor of and against congestion. However, the votes in favor of congestion has been over-weighed because C's decision is obtained from B. In such cases, the opinion of C should be discounted, by using a weighing factor α (< 1). So instead of considering C's vote as 1, it is considered as α. Such a weighing mechanism will counter oversampling, as we show later by analysis and simulation. This is the first time to study these two phenomenon in ad hoc networks.

We also show experimentally, that considering only the first hand information gives better results than voting based on the opinions of all neighboring nodes.

3.2 Information Cascading and Oversampling in VANET

Due to the rational aspects of VANET, the decisions taken by nodes in the network influences the decisions taken by other nodes. In certain situations, the opinions about events reported by nodes can be so overwhelming, that the opinion of one node is suppressed. It occurs mainly where decisions are made sequentially [25]. Consider the situation shown in Figure 3.1.

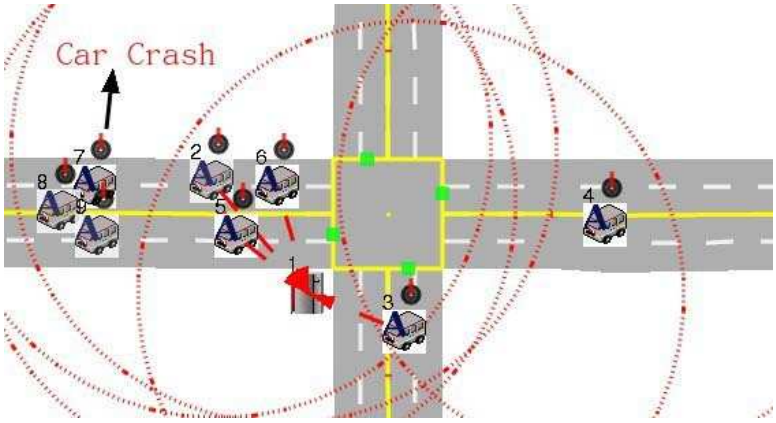

Figure 3.1: A network situation

There are five vehicles in the vicinity of an intersection (vehicles 2, 3, 4, 5 and 6), 1 is an RSU. Vehicles 2, 5 and 6 consist the first-observation set. Vehicle 3 is in the communication range of vehicles 2, 5 and 6. Vehicle 4 is in the range of vehicle 3. Let us assume some events (like a "Car Crash") occurs on the left of first-observation set (2, 5 and 6) (as shown in the Figure 3.1), then these vehicles will send out alert messages. After receiving these alert messages, vehicle 3 will make decision and re-broadcast these messages to vehicle 4. Vehicle 4 receives four messages from four different vehicles (messages from vehicles 2, 5 and 6 are first-hand and retransmitted by vehicle 3, and message from vehicle 3, which it calculates from the first-hand

decisions). If the majority of vehicles 2, 5 and 6 are incorrect (means two false information and one correct information), all the vehicles behind will receive wrong information, there is no way to prevent this.

Assume that vehicles 2 and 5 are good, while vehicles 6 and 3 are malicious/selfish. So, the majority of first-observation set is correct and vehicle 3 makes an incorrect decision deliberately. Vehicle 4 will receive two correct and two incorrect messages. This situation is called information cascading and oversampling, where the decisions of other nodes influences one node to take a decision contrary to its observation. When node 4 makes a decision, it uses the opinions of nodes 2, 3, 5 and 6. However, the opinion of node 3 is based on the opinions of nodes 2, 5 and 6. So, the observations of nodes 2, 5 and 6 are being oversampled. Even if vehicle 4 observes that vehicle 2 and 5 have acted as if there is a congestion, its decision might be overridden by that of 6 and 3. So a wrong decision will cascade through the entire network. There should be a way to decrease the importance of the message sent by vehicle 3.

One way to overcome this, is to give weight to the decisions made by nodes. For example, nodes which observe an event are considered with weight 1. However, less weight is given to nodes which are at two or more hops from the direct observers.

3.3 Reducing the Impact of Oversampling

Consider a network containing N nodes. Emergency events can be either congested road, hazardous road condition, accidents, etc. On observing such events, nodes transmit alert signals. Some nodes can generate false alert messages either for malicious or selfish reasons. A node which generates an alert message is known as a first-hand observer.

A node can receive contradictory messages about events, for example “accident”

and "no accident". In such situations, it has to decide which of these information is correct and transmits that information. Nodes can receive several messages from other nodes and make a decision about which one to accept. A node can receive messages from direct observers or through multi-hop paths.

3.3.1 Network Model

It is considered that nodes move in the same direction. If a node observes an event (for example, an accident) at the front of it, it transmits the information to the nodes behind it. Let n_i be a node. Node n_j is said to be in the neighborhood of n_i, if n_j is in the communication range of n_i and n_i is either at the distance greater than δ behind n_j or vice versa. The neighborhood of n_i is denoted by $nbd(i)$. This parameter δ is considered so that nodes n_i and n_j moving side by side do not consider each others message. We consider δ to be 5 meters.

A message M_i consists of a number c which denotes the number of hops from the source of the alert. For example, the first hand observers have $c = 0$, second hand observers have $c = 1$, and so on. This number is denoted by $c(M_i)$. M_i also contains a decision d_i. d_i can be either +1 or -1. Let F be a set of nodes which observe an event. They report either an accident C (correct) or no accident I (incorrect). The decision of any node n_i is denoted by d_i. A node n_i receives messages from its neighbors and has to decide whether there is an accident or not. It considers all the neighbors $n_j \in nbd(i)$ that are in front of n_i. A simple voting algorithm works as follows:

Let v_i = number of nodes which report C - number of nodes which report I. If $v_i \geq 0$, then $d_i = 1$ the node n_i says "there is an accident" and if $v_i < 0$, then $d_i = -1$ and the node says "there is no accident". We should notice that the bad vehicle gives the

opposite decision of its real opinion.

3.3.2 Our Algorithm

To deal with information oversampling we do not consider all the opinions received with equal weight. We give more weight to nodes which are closer to the event (that lead to the alert), than to nodes far away. The opinion of a node, which observes an event directly is given a weight of one, whereas a node which receives second hand information is given an weight α. The opinion of the node at two hops from the direct observer is given a wight α^2 and, so on. The pseudocode of our approach is given in Algorithm 1.

Algorithm 1 This algorithm decides the opinion of node n_i, based upon the messages it received from it neighbors $nbd(i)$

ALGO **Input**: Node n_i which has to make a decision and messages M_j, where $n_j \in nbd(n_i)$
Output: Decision taken by each node n_i "accident" or "No accident"

$v_i = 0$
for $n_j \in nbd(i)$ and in front of n_i **do**
    $w_j = \alpha^{c(M_j)}$
    $v_i = v_i + w_j d_j$
end for
if $v_i \geq 0$ **then**
    if n_i is a good node **then**
        $d_i = 1$
        Opinion of n_i is "there is accident"
    else
        $d_i = -1$
        Opinion of n_i is "there is no accident"
    end if
else
    if n_i is a good node **then**
        $d_i = -1$
        Opinion of n_i is "there is no accident"
    else
        $d_i = 1$
        Opinion of n_i is "there is accident"
    end if
end if
$c(M_i) = 1 + \min_{n_j \in nbd(i) \text{and in front}} \{c(M_j)\}$

Though the solution is described in the situation for VANET, this can be applied to MANET to deal with information oversampling. Instead of taking a majority decision, the information received from different nodes are weighed according to their distance from the actual observation. The weighted majority is then considered to determine the decision of the node.

If a vehicle A receives n messages, the set of vehicles R_1 sends first hand messages, the set of vehicles R_2 sends second hand messages and the set R_n sends n-th hand messages, then the decision of A is taken as $\sum_{i \in R_1} d_i + \alpha \sum_{i \in R_2} d_i + \cdots + \alpha^{n-1} \sum_{i \in R_n} d_i$, where d_i is the decision of node i. If this sum is over 0, A's decision is 1; otherwise it is -1.

3.4 NCTUns Simulator

To evaluate VANET protocols and applications, the important step is to perform an outdoor experiment. Several outdoor wireless communication technologies have been proposed in vehicular environment, such as GPRS, IEEE 802.11p and IEEE 802.16. Before the technology can meet the expectations, some experiments should be repeatedly done to test it to verify whether this protocol or application is feasible in the real-life environment. But, the large scale experiment is very costly in terms of time and money. For one specific experiment, many equipments need to be purchased. Besides, the experiments may face the potential dangers such as collisions due to its immature consideration. Therefore, software simulations play a vital role before these protocols and applications can be used in practice.

Software simulators for VANET can be roughly classified into two parts, namely traffic simulator and network simulator[35]. In general, network simulator can be used to evaluate the network protocols and applications in different situations whereas

traffic simulator can be used to generate traffic transportation and predict a driver's misbehavior. The network simulator is only for the network protocols and applications but a traffic simulator is only for the study of traffic[75]. These simulators work independently. In order to satisfy the requirements of VANET, the integrated simulation platform is needed.

Several network simulators[6, 8, 1] and traffic simulators[7, 10, 9] have been proposed. However, for our simulation, we choose NCTUns[74] simulator which is a highly integrated simulation platform. It supports interactions between a road network and communication network, hence it is suitable for the study of ITS. NCTUns(National Chiao Tung University Network Simulator) was proposed in 2002 by S.Y.Wang based on Harvard simulator. NCTUns is fully open source before version 6.0 and written in C++. With a powerful GUI support, the user need not worry about the complex coding.

3.4.1 Major Components

NCTUns has four main components in its architecture: Graphical User Interface (GUI), Simulation Engine (SE), Car Agent (CA) and Signal Agent (SA)[74]. Figure 3.2 shows the architecture and the relationship between these four components in NCTUns.

Graphic User Interface (GUI) The GUI provides users with an environment where they can easily construct desired road network. The road deployment can be finished by a few operations of mouse clicks instead of specification script file. Additionally, the network protocol selection can also be done in a few operations. After user finishes the desired road network, the GUI will automatically export the configuration files for other components. The GUI can play back the animations

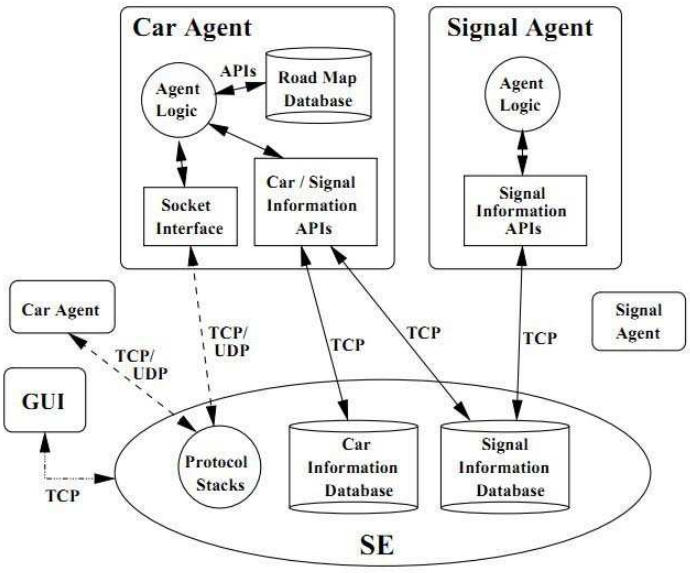

Figure 3.2: Architecture of NCTUns [75]

of packet transmissions and vehicle movement after simulation. This visual display helps users check the correctness of designs and simplify the operations.

Simulation Engine (SE) Before the SE begins the simulation, all required files have to be exported by the GUI for the SE to read. It stores some vehicle and signal traffic light information for servicing the request required by Car Agent and Signal Agent. The SE is responsible for network-layer and transport-layer protocols.

Car Agent (CA) Each car runs a Car Agent which has four components: agent logic, road map database, socket interface, and car/signal information API. The agent logic generates the driving behavior of each vehicle automatically while the road map database stores the direction and the roads. The socket interface supports TCP/UDP connections and exchanges messages on the road. The car/signal information API is the interface between SE and CA.

Signal Agent (SA) The SA is responsible for controlling the state of the traffic signal lights located at the junction. It has two parts: the signal logic and the signal information API. The signal logic determines when to change the signal state; signal information API are called by the signal agent to update the signal information database.

Road Types NCTUns provides several road types, including single-lane roads, multi-lane roads, cross roads and lane-merging roads. With the powerful GUI, the user can draw a desired network topology only with a few clicks. Wang in [74] illustrates the detailed steps on how to construct a road topology.

Vehicle Movement Controls There are two approaches for vehicle movement control in NCTUns 5.0: pre-specified and autopilot[75]. The user needs to specify the moving path and speed of each vehicle before the simulation in pre-specified approach. In this situation, the vehicle will move according to its pre-specified moving path and the car agent running on each vehicle does not control the movement of vehicle. By the contrast, in terms of autopilot approach, the car agent controls the movement automatically during the simulation.

New Modules NCTUns 5.0 is an open source code software. It consists of a few network modules and protocol modules. For some specific experiments which needs new protocols, users could easily build a new model and add it to the software. The detailed steps are given in [73].

In summary, NCTUns 5.0 combines the network simulator with the traffic simulator. It can simulate the microscopic wireless communication networks for ITS research. The powerful GUI makes NCTUns easy to use for the users, and also the open source code allows users create new modules for specific experiments based on

their own requirements.

3.5 Experimentation

This section shows the traditional simple voting scheme can result in incorrect decision making, while our algorithm performs better by reducing the effect of oversampling. The best approach is to rely only on the information of the first hand observers and transmit only that information across to other nodes. There is no need to transmit the decision of the intermediate nodes to the other nodes. This is because it will result in oversampling and hence incorrect results, and the intermediate nodes might be malicious/selfish and change received decisions.

The following simulation environment is considered:

1. The simulations occur around a road intersection.
2. In every experiment, 35 vehicles are randomly deployed in the vicinity of the road intersection.
3. Every car has the communication range of 100m (the effective range is about 80m).
4. Each experiment runs 10 times.
5. In this simulation, to get the theoretical evaluations of the mechanisms, no obstacles(like buildings) are added.

3.5.1 Simulation Results

In Figure 3.3, we show how our algorithm performs for different values of α.

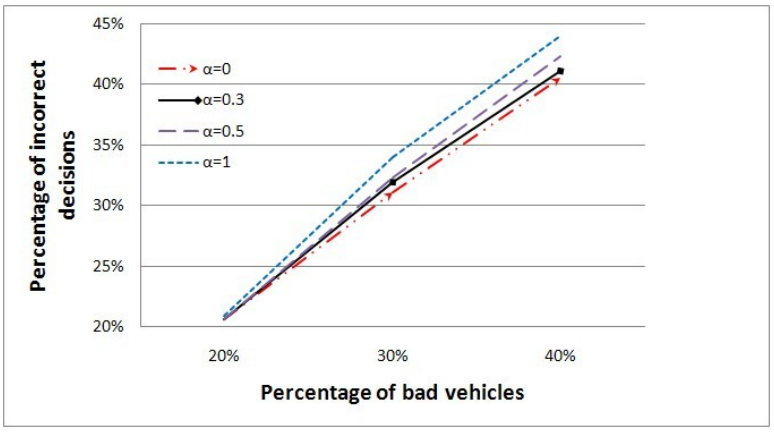

Figure 3.3: Experimental results with different values of alpha

We choose different values of α and show how the decision varies with changing values of α. According to Figure 3.3, we find when α=0, the performance of our algorithm is the best, the percentage of incorrect decision increase with the growth of α. This means the best performance is when we do not consider the decisions of vehicles which are not in first set.

Then we compare three different voting schemes in the following:

1. Mechanism 1: Each vehicle makes a decision based on the messages from the first observation set vehicles. This means every vehicle should obtain the same result for a specific event.

2. Mechanism 2: Simple voting mechanism - the messages vehicles use to vote are from neighbors. Neighbors are considered to be the vehicle who reports an event occurring in front of it.

3. Mechanism 3: This is a combination of Mechanism 1 and Mechanism 2. Vehicles make their own decisions based on their neighbor's decisions and the first

observation set.

Figure 3.4 shows that simple voting results in maximum incorrect decisions. It is shown that the Mechanism 3 performs better than simple voting for decision making. When only the first hand knowledge is considered, then the maximum correct decision is received.

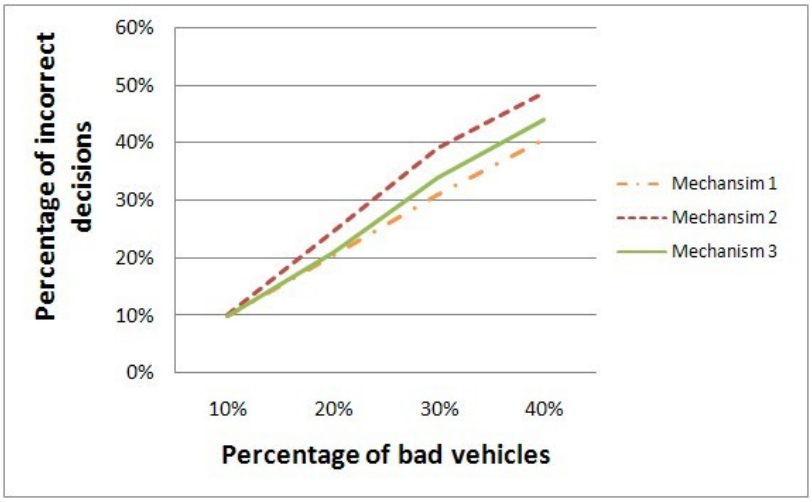

Figure 3.4: Experimental results comparing different decision-making mechanisms

The algorithm presented in Section 3.3.2 is considered as a way to handle information oversampling. One can note that if $\alpha = 1$, then it is Mechanism 3 and if $\alpha = 0$, then only the first hand information is considered, that is Mechanism 1. The results obtained with the simulations are:

- Majority voting for decision making results in a higher percentage of incorrect decisions in VANET because of the impact of information cascading and oversampling.

- Considering only the opinions of vehicles, which have directly observed the events have higher chance of correct decision making.

- Majority voting results in incorrect decision making, which can be corrected using our algorithm.

3.6 Conclusions

In this chapter, we discussed the concept of information cascading and oversampling as well as their impact to vehicular ad hoc networks. We then gave our simple and novel voting scheme by adding a weight factor α to each vehicle's opinion. The simulation results show the performance improvement of our scheme comparing with the traditional voting scheme. Note that our scheme could not only apply to VANET, but also can be used into MANET to decrease the influence of information oversampling.

Chapter 4

A Misbehavior Detection Model for VANET

From the previous chapters we know that the voting scheme is actually influenced by information cascading and Sybil attack. In order to deal with the deficiencies in trust management in VANET, we introduce, in this chapter, a new model for VANET and a data-centric misbehavior detection scheme which detects false alert messages and misbehaving nodes by observing their actions after sending out the alert messages. With the data-centric MDS, each node can independently decide whether information received is correct or false. The decision is based on the consistency of recent messages and new alert with reported and estimated vehicle positions.

As we discussed in Chapter 2, the revocation of certificates requires a huge bandwidth if the number of revoked vehicles is high. But, we argue that the revocation may not be necessary if a vehicle misbehaved only once for some selfish reasons. Examples of these reasons might be to reach their destination faster and/or to drive on a free lane. Vehicle might send false reports on congestion, accident or road block. It is conceivable to believe that a vehicle does not have malicious intentions of causing

accidents.

Each vehicle normally sends valid and useful information. If all the vehicle's certificates are revoked then useful information sent will be ignored by other vehicles. We believe that there is no need to classify vehicles according to their overall behavior, but instead one should be able to distinguish between correct and false information received from a vehicle. It is important to identify false data and the sender efficiently, because a delay of even one second might cause accidents. This problem is termed as *data-centric misbehavior detection* in contrast to entity-centric misbehavior detection where the main goal is to find out and penalize a misbehaving node.

The idea of data-centric misbehavior detection stems from Raya's work [57] on data-centric trust, where the author considers trust on information rather than on the source of information. In our approach, we do not revoke nodes which misbehave. Instead, the misbehaving node receives a fine depending upon its action. It can keep on sending further information which might not necessarily be malicious. The payment of fines would hopefully discourage nodes from sending further false messages.

We will concentrate on detecting false alert messages and false location information sent out by a node. Vehicles send periodic beacon messages, so that the positions of neighbors is monitored over time. Our method borrows some ideas from [30]. If reported position is not consistent with the real position then the receiving node declares the message as incorrect and discards it.

4.1 Model and Assumptions

The network consists of a set of nodes, $\mathcal{N}$, a set of RSUs, $\mathcal{R}$, and a set of CAs, $\mathcal{C}$. Vehicles are denoted by n_i (also called nodes), road side units (RSU) by R_i and Certification Authorities (CA) by C_i and a Master Authority (MA). We assume a

one way traffic network. We assume that vehicles are moving in a straight line and there are no curves, $U-turns$ or loops (as in over-bridges).

MA is headed by the government of a State or Province. Its authority is divided into several smaller regions each having a local authority named CA. The CAs are government agencies that maintain records of vehicles and their owners, and issue unique identities as license plates and secret credentials like pseudonyms, public/private keys and certificates. We use the authentication scheme ECMV [76] which completely suits our purpose. It has an hierarchical structure with several CAs. We assume that the CAs to be trustworthy and has authority over all vehicles registered locally.

We assume that RSUs are much more difficult to compromise than the vehicles. Although RSUs are sometimes in isolated places, their hardware may present some tampering proof capabilities, making it difficult for a regular human being to compromise it. This is not true in the case of vehicles, because its owner can drive to an expert just to have the vehicle tampered.

Nodes misbehave by sending out false information, mainly out of selfish motives like getting faster and easier access of a road. Depending on their intentions when sending a false information, nodes can either be faulty (damaged), selfish or malicious. Since most of the undesirable behaviors will be caused due to selfish motives, the classification of "benign" node and "malign" node is not so important as the distinction between correct information and false information. We will be more concerned with finding out if the message or alert signal generated by a node is correct or false, than if the node is "benign" node and "malign". This can be termed as *data-centric misbehavior detection*. This is different from entity-centric MDS.

A node can send several types of messages when on the road. We mainly deal with two types of messages. *Beacons* specify the location of the vehicles. *Alert messages* which ensure safety of vehicles on the road [14]. These include:

1. Emergency Electronic Brake lights (EEBL),
2. Post Crash Notification (PCN),
3. Road Hazard Condition Notification (RHCN),
4. Road Feature Notification (RFN),
5. Stopped/Slow Vehicle Advisor (SVA),
6. Cooperative Collision Warning (CCW),
7. Cooperative Violation Warning (CVW),
8. Congested Road Notification (CRN),
9. Change of Lanes (CL),
10. Emergency Vehicle approaching (EVA).

EEBL alerts that a vehicle is decelerating rapidly, so that the rear vehicles can prevent rear-end collisions. PCN alerts are sent by vehicles warning other vehicles of an accident which has already occurred. RHCN reports of road conditions like "slippery road" or "ice" or unwanted debris on the road. SVA alerts that a vehicle is moving slowly. RFN alerts of speed limits near schools and hospitals or a sudden bend or steep slope. CCW sends information about possible collisions that should be avoided. CVW warns vehicles about possible violations of traffic signals. CL notifies when a node is changing its lane. These conditions are sent by nodes to nodes behind them. The alert EVA might also be sent by nodes to other nodes ahead.

Alerts can be either observed or self generated. For example, a vehicle might observe the road hazardous condition (PCN, RHCN) or "school ahead" sign (RHN)

or a slow moving vehicle (SVA). Alerts can also be generated by the node itself, when it is decelerating rapidly (EEBL) or changing lanes (CL).

CAs know the mapping between the pseudonym used by each node and the unique id relating the pseudonym to the node. If there is evidence from the RSU and other nodes, that the node has misbehaved in a given situation then this is noted and a penalty is imposed in the form of a fine. Each false alert has an associated amount of fine, depending on the impact of misbehavior. The difference is that the decision is no longer taken by an authority (police or speed cameras), but taken by fellow nodes or RSUs. We will assume that a faulty vehicle will be condemned in a similar way. Therefore, it is up to the owner to maintain all vehicle's sensors at good conditions and always be aware of what messages are being sent out.

We define *freshness interval* as the time period during which a message is considered as valid and potentially effective. This will vary for different types of alerts. We denote it by F.

4.2 Notations

Table 4.1 gives the notations that we follow throughout the rest of the this chapter. We use nodes and vehicles interchangeably.

4.3 Proposed Misbehavior Detection Scheme

We first give a sketch of our approach and then work out the details.

Table 4.1: Table of Notations

Notation	Meaning
N	Number of nodes in the network
n_i	ith node
R_i	ith RSU
E	Event
T_E	Type of alert caused by event E
L_E	Location of event E
F	Period of freshness
p_{it}	Pseudonym of node i at time t
l_{it}	Location of n_i at time t
M_A	Alert message
M_R	Replay message
M_B	Beacon
$dist(l_{it}, l_{jt})$	Distance between l_{it} and l_{jt}

4.3.1 Sketch of our Misbehavior Detection System

Suppose a node n_j having a pseudonym p_{jt} sends out an alert message M_A at time t. Once a node n_i receives some alert signal from a node n_j, it finds out from the alert message, the type of alert and the location of the event E for which the alert was generated. For example, the alert might be "There is road block in location X". In this case E would be "road block" and L_E would be X. The type T_E is RHCN.

When n_i later receives a beacon from node n_j after an elapse of time Δt, it checks the current location of n_j, and checks if it can be a valid location for n_j. For example if node n_j first sends a message "There is a road block in location X" and after time Δt it is close to location X, then it implies two contradictory statements and the alert message cannot be trusted. In this case n_j might send a false alert, n_j sends to divert the traffic away from the location X.

For this reason, each node maintains a list called the *list of invalid events, (LIE)*, for short. LIE contains a list of events and corresponding invalid actions. For example, if there is an emergency breaking, then the distance between the old and new positions

of n_j cannot be more than 100 meters. In Table 4.2, we present event and invalid action pair.

Table 4.2: Events and invalid actions

Event	Expected action	Invalid Action
EEBL	Car must slow down	$D > d$ meters
PCN	Car stops/Changes Lane	$D > d$ meters and No Lane Change
RHCN	Car stops/ Changes rout	$D > d$ meters and Same Route
RFN	Decrease speed	$D > d$ meters
SVA	Change lane/decrease speed	$D > d$ meters and Same Lane
CCW	Slow down	$D > d$ meters
CVW	Slow down	$D > d$ meters
CL	Lane change	Same Lane
EVA	Change Lane/slow down	$D > d$ meters and Same Lane as Vehicle

Where $D \equiv dist(l_{jt_1}, l_{jt_2})$. The first and third columns are used to build LIE.

LIE will contain the information from first and third column of the table. The change of lanes can be known from the position information and interpreting it using GPS. d is the safe distance. For example if a car is driving at 80kmph when it observes the alert and then reduces its speed to 20kmph as a consequence of the alert, then it will travel less than 100 meters in the next two seconds. Thus the positions sent in the beacons should be less than d =100 meters apart.

As soon as the node n_i receives an alert message it retransmits it. Later, after time $\hat{t}$, if n_i realizes that the message from node n_j is incorrect, then it sends out the negation of the alert message already sent. Node n_i checks for a certain time interval $\hat{t}$ and if it does not receive any beacon during this interval, then it assumes that the node n_j has changed its pseudonym. It also sends a message to the RSU, convicting n_j of sending false alert message. RSU checks with its own observation and sends a message to the CA stating the misbehavior and the pseudonym of the node n_j. CA knows the mapping of the pseudonym with the original id and update

its records against node n_j. It is to be noted here that the node transmitting the alert message must not change the pseudonym anytime between sending the alert message and beacon.

4.3.2 The Details

There is a pool of pseudonym $\mathcal{P}$. A node n_i is given a set of pseudonyms P_i from this pool, such that $\bigcup_{i=1}^{N} P_i = \mathcal{P}$. Each node n_i has an id i and pseudonym $p_{it} \in P_i$, at time t. Let τ denote time. Each vehicle has an on board unit (OBU), which is loaded with a public/private key pair, corresponding to each pseudonym, by a CA.

We use the certificate management scheme, ECMV of Wasef et al [76]. There is a master authority (MA) and several CAs. The MA generates public/private key pairs and two secret certificate signing keys, for each CA. The MA also generates public keys for verifying the certificates of RSUs and nodes. Each CA uses the certificate signing keys to sign a certificate set of each RSU. The certificates in a set are shared among all the RSUs in the set. The other secret key is used to generate a partial signing key for each RSU, to generate certificates for each node within its range. Public keys can be used by CAs, RSUs or nodes to verify the certificates of RSUs and nodes. ECMV has certificate update algorithm which suits our objective.

An event E can be like:

1. Emergency breaking,
2. Observation of unwanted debris on road or hazardous road conditions like “ice”, “slippery road” etc,
3. Observation of “Drive slow” sign in areas like school or hospitals or steep slope,
4. Crash Notification,

5. Approaching emergency vehicles.

A list of emergency alerts has been discussed . We denote the set of types of alerts by $\mathcal{T}$. The set of locations is given by $\mathcal{L}$. $\mathcal{M}$ denotes the message space.

An *alert message*, denoted by $M\_A \in \mathcal{M}$ is

$$M\_A = (p_{it}, E, T_E, L_E, t, l_{it}),$$

where, $p_{it} \in P_i$ is the pseudonym of the node n_i who generated the alert at time $t \in \tau$,
E is alert generated,
$T_E \in \mathcal{T}$ is the type of alert E,
$L_E \in \mathcal{L}$ is the location of the event E_j for which the alert was generated,
$t \in \tau$ is the time at which the alert message had been sent,
$l_{it} \in \mathcal{L}$ is the location of the node n_i which generated the alert at time t.

A node that receives an alert message from a neighboring node, relays it to other nodes and RSUs in its vicinity. A *relay alert*, denoted by M_R is

$$M\_R = (p_{it}, t, M\_A),$$

where, $p_{it} \in P_i$ is the pseudonym of the node n_i who sends the relay alert, $t \in \tau$ is the time at which M_R was sent and $M\_A \in \mathcal{M}$ is the alert message that it is relaying.

A *beacon* sent by a node is denoted by M_B and is

$$M\_B = (p_{it}, t, l_{it}),$$

where, $p_i \in P_i$ is the pseudonym of the node, $t \in \tau$ is the time at which the beacon was sent, and $l_{it} \in \mathcal{L}$ is the location of the node.

When a node n_i, having a pseudonym p_{it_2} at time t_2 receives an alert message M_A, from a node n_j (with pseudonym p_{jt_1}), it first checks if it has a valid signature. This can be done by ECMV [76]. If node p_{jt_1} has a valid signature, then n_i notes

the time t_1 from the message $M_A = (p_{jt_1}, E, T_E, L_E, t_1, l_{jt_1})$. L_E is the location of the event E for which the alert was generated. We define a threshold time F after which a message becomes stale. F is also known as *period of freshness*. If $t_2 - t_1 > F$, then the message originally sent by n_j become stale and n_i discards it. If the message is fresh then the position of the event L_E and the location of node l_{jt_1} is noted. If the positions are contradictory, then no action is taken for the alert and the message is discarded.

The positions are contradictory, if n_j is between n_i and E, assuming they are moving in a straight line. If the order of location is anything other than $n_i - n_j - E_x$ or $E_x - n_j - n_i$. The first condition arises when the event (for example, an accident at E and emergency breaking alert is raised by n_j, or there is road hazard like water or ice on road) has occurred in front of n_j and n_j is sending a message to the node n_i behind it. The second condition arises when the event has occurred behind n_j and n_j is sending a message to the node n_i, which is in front of it. For example, if there is an emergency vehicle approaching from behind, node n_j reports to n_i (who is in front of it) to make space for the vehicle approaching from behind.

If the positions are correct, then the node n_i analyzes the alert. We will see in the next section how to detect incorrect location information. Node n_i might receive more than one alert messages from different nodes. We do not, however, make a decision on the validity of the alert based on the number of vehicles that report the alert, because we do not rely on voting. This is main difference from other VANET MDS. For this reason, Sybil attack is not effective against our scheme.

After receiving an alert message, node n_i waits for beacons from n_j for a time period of $\hat{t}$. It verifies the position of the node n_j from all the beacons it receives during this time period. When the node n_i receives beacon message from the node n_j, it checks the position L_E in the alert message and the position l_{jt_3} in the beacon

message $M\_B = (p_{jt_3}, t_3, l_{jt_3})$. It checks the LIE to see if the alert type contradicts with the position. If it does, then n_i sends out an alert message which is the negation of the previous message. It also reports to the nearest RSU and convicts node n_j for sending a false message. If node n_i does not receive any beacon from n_j in time $\hat{t}$ after receiving an alert message from n_j, then it assumes that the pseudonym has changed. Changing pseudonym within a time $\hat{t}$ is considered to be a misbehavior and so n_i reports the RSU that n_j is misbehaving.

The RSU, upon receiving such conviction messages, compares with its own observation and reports to the CA the pseudonym of the misbehaving node along with the reason for accusing it. Only the CA can match the pseudonym with the original identity of the node. The CA then issues negative points to the node, which has to pay it as a fine, depending on the number of negative points received.

Our assumptions are based on the fact that, any misbehaving node does so, mainly due to selfish reasons and is most likely not to send false message all the time. A large number of misbehaviors can be interpreted as a malicious motive and such nodes can be revoked off their certificates and other secret credentials using the revocation scheme in PASS [70].

4.3.3 Detecting Incorrect Location Information

In the previous section, we assumed that the location information sent in the alert message or in the beacons is correct. However, a clever malicious node will also send incorrect location information, along with the false alert message. In this section we propose a technique to detect incorrect location information.

Studder et al.[67] have presented how to detect nodes moving in a straight line and transmitting false location information. The decision to convict a node depends

on the number of votes casted against it. If a node is surrounded by many corrupt nodes, then a node is not convicted. The authors also show that if the first two nodes in the straight line (convoy) send false messages, then these messages cannot be detected.

In our scheme, if a node transmits a false alert or beacon with wrong position information, we do not use the limited incorrect location detection approach of Studder et al.[67], instead we use the following approach: suppose a node n_j sends a beacon at time t_1, and n_i (in communication range of n_j) receives the message at time t_2. Then t_2 is given by:

$$t_2 = t_1 + \frac{dist(l_{it_2}, l_{jt_1})}{c} \qquad \textit{(where c is the speed of light)} \tag{4.1}$$

Suppose that node n_j wants to fake its location as l'_{jt_1}, so it sends a beacon with the information $(p_{jt_1}, t_1, l'_{jt_1})$. Node n_i receives it at time t_2. Node n_i finds out that n_j is misleading because Eq(4.1) does not hold. To convince n_i, that it is a valid location, node n_j should also change the time stamp when the beacon is sent. The time t'_1 at which it should have sent the message, so that n_j receives it at t_2, is given by

$$t_2 = t'_1 + \frac{dist(l_{it_2}, l'_{jt_1})}{c} \tag{4.2}$$

Therefore,

$$t'_1 = t_1 + \frac{dist(l_{it_2}, l_{jt_1})}{c} - \frac{dist(l_{it_2}, l'_{jt_1})}{c} \tag{4.3}$$

Therefore, node n_j sends a beacon $(p_{jt_1}, t'_1, l'_{jt_1})$ to convince n_i that it is sending the correct message. However, since node n_j does not know the distance between itself and the node n_i accurately, it cannot accurately calculate t'_1. So, when node n_i observes the time stamp t_1 and the false location l'_{jt_1}, then any node can calculate the

expected position and verify it according to equation (4.1). We will show in Section 4.5 that there is an expection to this.

We will include this observation in designing our algorithm. On receiving the message M_A, a node n_j first checks the authenticity of the message using ECMV [76]. If the message is not stale ($t_2 - t_1 < F$, Step 5) and the order of the vehicles is either $n_i - n_j - E$ or $E - n_j - n_i$ (checked by the condition $dist(l_{jt_1}, l_{it_2}) < dist(l_{it_2,L_E})$), then node n_i waits for beacons from n_j during time $\hat{t}$. Suppose n_i receives a beacon message $M\_B = (p_{jt_3}, t_3, l_{jt_3})$ from n_j at time t_3. n_i first checks the validity of the location using Eq(4.1). It then looks up in LIE for event E. If the action is invalid, then it reports misbehavior and broadcast the negation of the message. If action is correct, then it broadcasts the message M_B. The variable count, keeps track of the number of beacons received during the interval $\hat{t}$. If no beacon is received, then it implies that the node n_j has changed its pseudonym and this is reported and misbehavior reported. We present the pseudo-code of MDS algorithm as Algorithm 2.

The procedure *check_action_function* takes the inputs LIE, T_E, t_1, t_3, l_{jt_1}, l_{jt_3} and outputs 0 or 1 depending on malign or benign behavior. First it finds out the event E that cause the alert. Then it looks up in the table LIE to check if the conditions corresponding to E hold. If the conditions match then r is set to 0, meaning that there is a misbehavior. If the conditions in the table LIE do not match, then $r = 1$. The function *report_misbehavior* takes as input the pseudonym of the reporting node and the false alert message and send it to the RSU. *Message_negation* creates the negation of the message. For example if the alert was "There is ice on road X", then the negation of the message is "There is no ice on road X". If node n_i retransmits a false message, then it will be found in either of the two ways:

Algorithm 2 MDS algorithm operated by a node n_i

Input: Alert message M_A, beacons M_B, Table LIE
Output: "Valid Alert" or "Invalid Alert"

$Tag = 1$, $count = 0$
Node n_i receives $M\_A = (p_{jt_1}, E, T_E, L_E, t_1, l_{jt_1})$ at time t_2
Check authenticity of M_A using ECMV
if M_A is authentic and Eq(4.1) holds **then**
  if ($t_1 < t_2$ and $t_2 - t_1 < F$) and ($dist(l_{jt_1}, l_{it_2}) < dist(l_{it_2,L_E})$) **then**
    while $t < t_2 + \hat{t}$ **do**
      while n_i receives beacon from n_j **do**
        Node n_i receives $M\_B = (p_{jt_3}, t_3, l_{jt_3})$ at time t_4
        $count = count + 1$
        if t_3 and t_4 satisfy equation(4.1) **then**
          Look up in LIE for type T_E of event E
          $r = check\_action\_function$(LIE, E, T_E, t_1, t_3, l_{jt_1}, l_{jt_3})
          $Tag = 0$
          if $r = 1$ **then**
            Broadcast $M_R = (p_{it_5}, t_5, M_A)$
            Take action against E
            Print "Valid Alert"
          else
            $M\_A' = Message\_negation(M\_A)$ {creates the negation of the message}
            Broadcast $M\_R = (p_{it_5}, t_5, M\_A')$
            $Tag = 1$
          end if
        else
          $Tag = 1$
        end if
      end while
    end while
    if $Count = 0$ **then**
      "Pseudonym Change"
      $Tag = 1$
    end if
  else
    $Tag = 1$
  end if
else
  $Tag = 1$
end if
if $Tag = 1$ **then**
  $report\_misbehavior(p_{it_5}, M\_A)$
  Discard M_A
  Print "Invalid Alert"
end if

1. By node n_j (if within communication range) by observing that M_R does not contain the message M_A that it had sent.

2. By other nodes which receive M_R and subsequent location information from beacons of n_j. They will find out if node n_j is malicious by an algorithm similar to Algorithm 2.

However it is possible that node n_i is falsely accusing node n_j. In this case the RSU will reject the accusation and convict n_i, based upon its observation. The other cars will be warned that n_i is lying. Since the results are also verified by the RSU before sending to the CA, even if the observing vehicles are misbehaving, the misbehavior can be detected. The messages are signed using ECMV by n_i. If the RSU changes the message, then it has to sign the message with n_i's private key, which it cannot access. So RSU cannot modify the message. The above approach does not require any voting or majority, so a Sybil attack does not have any effect in misbehavior detection.

We recall LEAVE protocol [60] and sting protocol [50]. As we had pointed out that misbehaving node can convict honest node, but will go undetected. Our detection mechanism ensures that misbehaving nodes are punished.

4.3.4 Dealing with Compromised RSUs

RSUs are prone to be compromised. However, compromising RSU is much difficult that compromising nodes. Compromised RSUs can either transmit false messages or convict benign nodes. A RSU which transmits false messages can be noticed by other nodes, and compared with their own observations. Nodes then report these to the next RSU. If it receives a large number of such reports over a long time, then the RSU is considered to be compromised If benign nodes are convicted, they are imposed fines by the CA. If the message sent by a node has been modified by the RSU, then

the convicted node can prove it using its signature. We note here that in ECMV certificates are not created by the RSUs. So RSUs cannot fake the signature on the messages, sent by the nodes. The nodes can then prove their message authenticity, using the signatures on the messages.

Once it is known that an RSU has been compromised, its certificate is revoked using the techniques in ECMV scheme. We note that there are fewer RSUs, compared to the nodes. So broadcasting a CRL for RSUs will not be expensive. ECMV protocol can be run and a set of keys and certificated is given to the RSU.

4.4 Performance Analysis and Comparison

This scheme is mainly on the data-centric detection but not pseudonym .In this section we concentrate on the efficiency of detection scheme by presenting simulation results and compare our scheme with existing misbehavior detection schemes.

4.4.1 Simulation Results

We use the network simulator NCTUns to show the validity of our MDS. The results illustrates that our algorithm (Algorithm 2) detects misbehaviors with a very high accuracy. The false positive rate (rate of detecting honest nodes as misbehaving nodes) is very low. Also we calculate the distance d in LIE table.

4.4.1.1 Signal Speed in NCTUns

Signal is transmitted in air in the speed of light. As we discussed above, the speed of light c is one of the key parameters in the detection procedure. In general, $c = 3 \times 10^8$ meters per second in vacuum. However, in our simulator NCTUns, we find the transmission speed C is not equal to 3×10^8 m/s.

For our purpose of detection, we should first know the exactly signal transmission speed in NCTUns. In order to obtain speed C, we design an experiment: the source vehicle is fixed in the junction of the roads, while other 20 vehicles are fixed on the road around the source vehicle and in its communication range. Source vehicle broadcasts a message at time t_1, and we calculate the receiving time of this message by other vehicles. Meanwhile we know the distance between them with source vehicle. Therefore we can obtain the transmission speed of the signal by equation:

$$C = Distance/time \tag{4.4}$$

For statistically significant results, we repeat the experiment 15 times. Table 4.3 shows the experiment parameters. The simulation was carried out on a Intel Core i7 processor with speed of 2.93 GHz.

Table 4.3: Experiment parameters for speed calculation

Frequency	2400MHz
Channel Model	Two Ray Ground
Antenna Height(m)	1.5
Transmission Power(dbm)	15
Transmission Range(m)	250
Type of Antenna(degree)	360
TxGain of Antenna	1
RxGain of Antenna	1

From the experiments, we obtain that the signal transmission speed C in NCTUns is 33,931,686 m/s.

4.4.1.2 Correctness of Equation(4.1)

We will now evaluate the performance of our novel and simple detection mechanism.

The validity of our misbehavior detection scheme depends on the correctness of

Equation(4.1). We will assume that pseudonymous authentication using ECMV is already available and we concentrate on the position detection using Equation(4.1).

As we discussed , if a vehicle A who wants to fake its location it must change the message time stamp at the same time, otherwise Equation(4.1) will no longer be valid. However, the vehicle A does not know the exact distance between itself and the vehicle n_i, it is infeasible to accurately fake the time stamp in order to meet this equation. Therefore theoretically speaking, Equation(4.1) makes sure that each vehicle could detect any misbehavior in location information by calculating the expected position and verify it with the position in message.

To verify the validity of Equation(4.1), we consider two parameters: 1) *false position detection rate* and 2)*false positive rate*. False position detection rate is the probability that this equation detects the position misbehavior. False positive rate is the probability that a result is actually correct but tested as incorrect.

For our simulation, false position detection rate is the probability one vehicle's misbehavior can be detected when this vehicle is lying on its position, false positive rate is the probability that one vehicle is detected as lying on its position but actually it does not.

4.4.1.3 False Position Detection Rate

We have discussed that Equation(4.1) detects any misbehavior on location information. In the following, we obtain the false position detection rate by simulation to verify whether its performance satisfies the theoretical value 100%.

The simulation is on the straight line road segment. We consider one particular misbehavior situation, where a vehicle reports of a crash. Let vehicle A detects a "car crash" in front of it and then broadcast the alert message to the vehicles behind it at time t_1. Vehicle n_i receives this alert message at time t_i ($t_i > t_1$) and

detects the estimated position of vehicle A using Equation(4.1). Then we verify A's position included in the alert message with the detected position $L\_estimated$ we have obtained. If they are different then we consider a misbehavior is detected.

In the simulation, we assume that there are ten vehicles behind A. They are all in the communication range of A which means they are one-hop neighbors of A, and the location of these ten vehicles are randomly distributed. In order to get the false position detection rate, we introduce these notations in Table 4.4.

Table 4.4: Notations for experiments

L_real	A's real location
L_false	Fake position of A in the alert message
T_1	Sending time of A's alert message
T_n	Alert message Receiving time by n_{th} vehicle
L_n	Position of n_{th} vehicle at time T_n
C	Signal transmission speed

Let $D\_estimated$ donates the estimated distance between A (as calculated using the time stamp of the alert message) and vehicle n_i. We obtain,

$$D\_estimated = (t_i - t_1) * C \tag{4.5}$$

Vehicle n_i can calculate this value. This is compared with $dist(L\_false, L_i)$. Let μ be the tolerable error in distance estimation. We will later specify this value in certain cases. If

$$|D\_estimated - dist(L\_false, L_i)| \leq \mu, \tag{4.6}$$

then A is not lying about its position. If Equation(4.6), vehicles behind A will consider A is lying on its location information in the alert message and report nearby RSU. To obtain the accurate results, this experiment is repeated 30 times. One point to be

noted here is that $|D\_estimated - dist(L\_false, L_i)| \leq \mu$, does not necessarily mean that A is not lying (as we will see in Figure 4.6). However, such occurrences are rare and can be neglected in practice. Table 4.5 shows the parameters and protocols we choose in the simulation.

Table 4.5: Experiment parameters for false position detection

Frequency	2400MHz
Path loss model	Two Ray Ground
Antenna Height(m)	1.5
Transmission Power(dbm)	15
Transmission Range(m)	250
Type of Antenna(degree)	90
TxGain of Antenna	1
RxGain of Antenna	1
Vehicle speed	30 Km/hour
MAC protocol	802.11p
Fading channel	Ricean
Length of vehicle(m)	4
Width of road(m)	30

If A wants to convince other vehicles its fake location it should change t_1 to satisfy Equation(4.1) in the same alert message which is infeasible because A cannot calculate t_1 in advance since it does not know the accurate distance between itself and vehicle n_i. In this simulation, first vehicle A randomly generates false position L_false within the range of 30 meters from its real position and adds this false position into alert message. Since the experiment is performed 30 times and there are 10 cars behind A we have 300 results. We observe that out of these 300 results 19 results satisfy the following condition:

$$|D\_estimated - dist(L\_false, L_i)| \leq 4 \tag{4.7}$$

This means that the deviation between false position and real position is less than

4 meters which is the length of vehicle we have set in Table 4.5. We note here that $\mu = 4$ which is the length of the vehicle. In this situation, we consider that the estimated position is actually equal to the A's false position and equal to the real position. Hence we have:

$$L\_estimated \approx L\_false \approx L\_real \tag{4.8}$$

Although this false position will not affects others, it is still a misbehavior, but our Equation(4.1) cannot detect this misbehavior. So, out of 300 results our simulation results show that false position is detected only for $(300 - 19)/300 = 93.7\%$ cases. This is when A picks a false position within 30 meters of its real position.

We define δ as radius of the circular region around the vehicle, where the vehicle can be falsely located.

We define false position detection rate as,

$$FPD(\delta, \mu) = \frac{\text{Total number of observations–Number of observations where Equation(4.6) holds}}{\text{Total number of observations}}$$

We obtain the false position detection rate $FPD(\delta, 4)$ of Equation(4.1) which is shown in Figure 4.1. According to the diagram, with the increase of δ, the detection rate also increases. When δ=200m, the false position detection rate is approximately 99% which is very close to the theoretical value 100%, therefore Equation (4.1) is very effective in detecting the target vehicle's false position in a straight line road segment.

4.4.1.4 False Positive Rate

Another important efficiency norm of our scheme is false positive which means a result is actually correct but tested as incorrect. This means that vehicle A broadcasts its

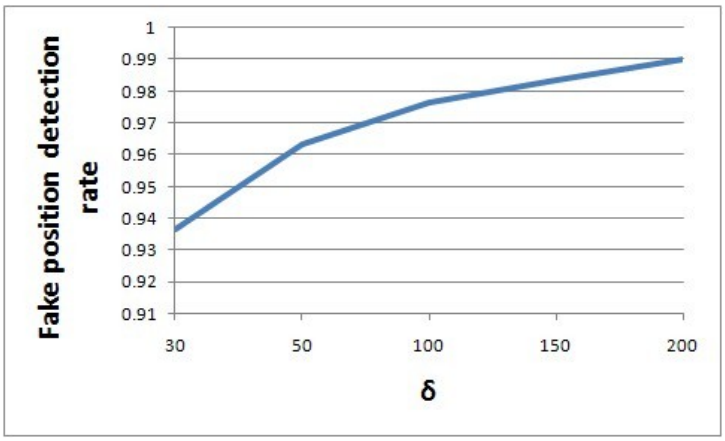

Figure 4.1: False position detection rate varies with δ(in meter)

location information correctly, but is detected to be misbehaving. To get the exactly false positive rate, we designed a similar experiment. Vehicle A is fixed on a straight road and 10 vehicles are randomly distributed behind A, all in the communication range of A. At time t_1, A sends out alert message with position L_real to the following ten vehicles, at time t_i vehicle n_i receives this alert message and then estimates the distance of A by Equation(4.5). If

$$|D\_estimated - dist(L\_real, L_i)| \leq \nu \tag{4.9}$$

we call it a false positive. ν is the tolerable error in estimation.

For the simulation experiment, the parameters are the same as in Table 4.5. To make the results more accurate, this step is repeated for 30 times that means we obtain 10×30 different results. Figure 4.2 is the distribution of estimated position of A by the following vehicles. We have set vehicle A's real position is at coordinate 0, $L\_real = 0$, n_i vehicle's position $l_i > 0$. We can see all these 300 results of estimated position are in 20 meters range of A's real position 0. Let us donate by ϵ, the range of estimated position from coordinate 0 and then count the number of points that lie

between $-\epsilon$ and ϵ. For instance, $\epsilon = 2$ means the area from range -2 to +2 in Figure 4.2, we count the number of points in this area. Table 4.6 shows the different value ϵ and its corresponding number of estimated positions.

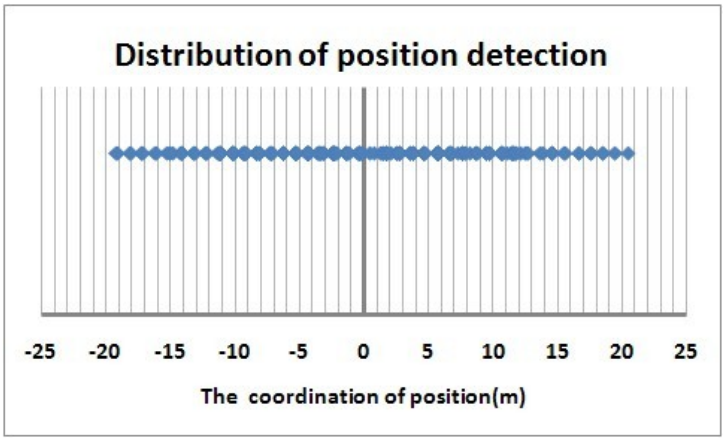

Figure 4.2: The distribution of 300 estimated position of A while A's real position is located at 0

Table 4.6: Numerical values corresponding to Figure 4.2

Value of $\epsilon(m)$	Number of estimated positions
2	38
4	95
6	136
8	179
10	213
12	257
14	269
16	284
18	293

We define false positive rate as

$$FP(\epsilon, \nu) = \frac{\text{Total number of observations–Number of observations where Equation(4.9) holds}}{\text{Total number of observations}}$$

We obtain the false positive rate $FP(\delta, 4)$ of Equation(4.1) which is shown in Figure 4.3.

We now discuss the meaning of Table 4.6. From the first row, we see that there are 38 results which estimate the position of A as lying between 2 meters around its real position. However, since the length of a vehicle is 4 meters (by Table 4.5), this means that the 38 results correctly estimate A's position. So, when $\epsilon = 2$ false positive rate is $(300 - 38)/300 = 0.873$. Similarly, a total of 95 results estimate the location of A, 4 meters within its real location. If we make the assumption that there is only one vehicle A in this area, we can say these 95 estimates positions approximately accurately detect A's real position, hence we have:

$$L_{estimated} \approx L_{real} \tag{4.10}$$

While 293 results are in the area of $\epsilon = 18$ means 293 results are approximately equal to A's real position under the assumption that there is only A in this 36 meters area. Figure 4.3 shows the false positive rate vary from different ϵ.

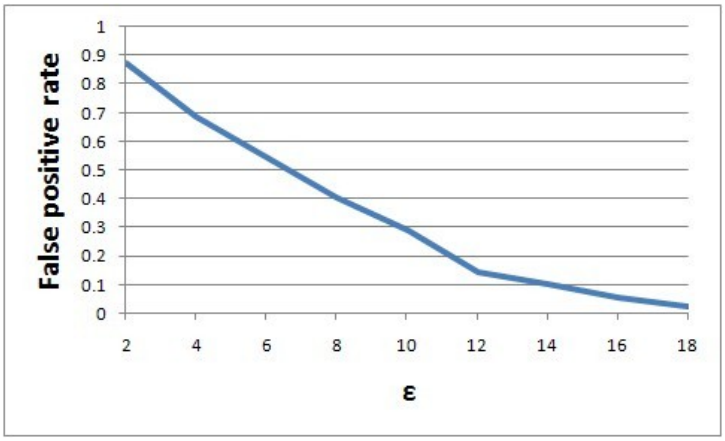

Figure 4.3: False positive rate vary with ϵ(in meter)

In practice, every two vehicles should have some gap between them, and this

gap cannot be very short since short gap decreases the reaction time of driver and adversely affects traffic safety. The distance between two cars on the highway is always over 50 meters because the very fast moving leaves drivers little time to react to the accidents. While in urban site the distance between vehicles could be a little shorter but it is also over 10 meters. From Figure 4.3, when $\epsilon = 2$ meters the false positive is approximate 90% but when $\epsilon = 18$ meters, the false positive is only 2%. The false positive rate decreases with the growth of ϵ. Therefore if we give the valid assumption that on highway the distance between two vehicles is 50 meters while in urban site the distance is 20 meters. We will get the false positive rate approximate 0% and 30% for highway and urban respectively.

4.4.1.5 Calculation of Distances d and D LIE Table 4.2

After checking the validity by Equation(4.1), our algorithm needs to check the LIE table to see whether the distance D satisfies the LIE distance d. D is obtained by the following equation:

$$D = Vt - Dec \times t^2/2, \tag{4.11}$$

where V is the speed of vehicle A at time t_1, t is the time interval between alert message and beacon message, Dec donates as the deceleration of vehicle A. We set $t = 1$s, and then obtain the theoretical distance D as shown in Table 4.7.

Table 4.7: Theoretical distance D(m) ($Dec \equiv$ deceleration in m/s^2)

Speed (Km/h)	$Dec = 2$	$Dec = 3$	$Dec = 4$	$Dec = 5$	$Dec = 0$
20	4.5	4	3.5	3	5.5
40	10	9.6	9.1	8.6	11.1
60	15.6	15.1	14.6	14.1	16.7
80	21.2	20.7	20.2	19.7	22.2
100	26.8	26.2	25.8	25.3	27.8

We realize that when $Dec = 0m/s^2$ which means the vehicle does not decelerate after sending the alert message, the distance of vehicle pass during this 1 second is maximum. When fixing D in LIE, we assume $Dec = 0$ and fix different D's corresponding to different speeds as given in Table 4.7. This is because of the following reason: If we set $d = 27.8m$ to all different speed, we cannot detect many misbehaviors. For example, if A's speed is only 20Km/h, and it does not decelerate, the distance D vehicle A has go through is 5.5m which is far less than 27.8m. A is actually misbehaving because A should decelerate after sending the alert message. In this case, if d=27.8m, we can not detect A's misbehavior. Hence we should set different value of d according to different speed. To simplify the simulation and make detection more accurate, we set d is equal to the distance of $Dec = 0$ which is the maximum distance, we have Table 4.8.

Table 4.8: LIE distance d with corresponding vehicle speed

LIE Distance d(m)	Speed (Km/h)
5.5	20
11.1	40
16.7	60
22.2	80
27.8	100

We apply these values to the simulation, and see the false positive of these values. In our experiment, vehicle A sends one alert message and one beacon message 1 second later. A does not misbehavior on its position. A randomly chooses a speed and the following vehicles check whether the difference of location in alert message and beacon message is less than the corresponding d. This experiment is repeated for 10 times. From all these 100 results, we find they are all satisfy the LIE table and no false positive happens. Therefore, this d value is effective for our scheme. Figure 4.4

shows how d value varies with the speed of vehicle.

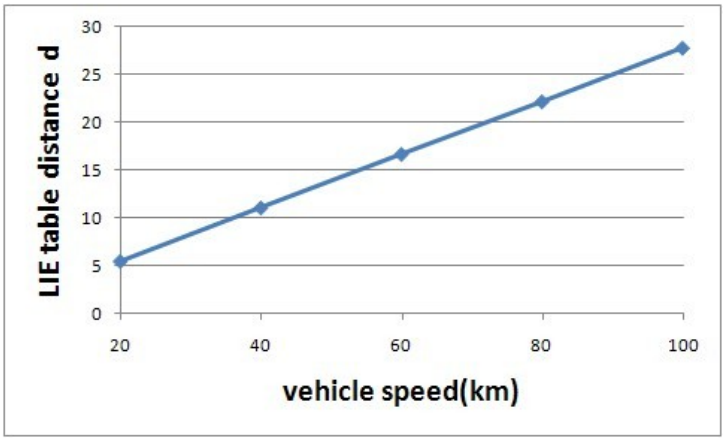

Figure 4.4: LIE table distance for different speed

4.4.2 Comparison with other MDSs

We compare our scheme with existing MDS schemes in Table 4.9. Our scheme has all the desirable properties like location privacy, ability to detect false location information and immunity against Sybil attacks.

Table 4.9: Comparison with other misbehavior detection schemes

Scheme	Location Privacy	False Location Info.	Immune to Sybil Attacks
LEAVE [60]	Yes	Yes	No
Golle et al [31]	No	Yes	No
Zhou et al [81]	No	No	Yes
Ghosh et al [30]	No	No	No
Ours	Yes	Yes	Yes

In our scheme, we have used ECMV [76] for authentication which fits into the hierarchical structure of our network. The transmission delay is only 6.47ms (as stated in [76]). The time taken for certificate verification is 14.7ms and for signature

verification is $9.6ms$. ($3T_{par} + 2T_{mul}$ and $2T_{par} + T_{mul}$ where $T_{par} = 4.5ms$ and $T_{mul} = 0.6$ms).

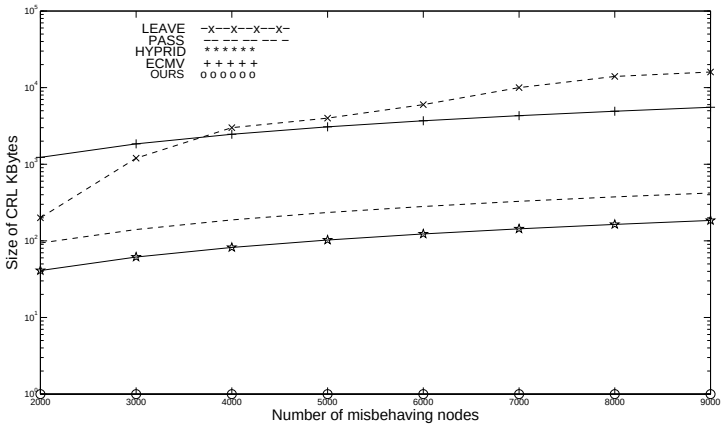

Figure 4.5: Comparison of communication overhead required for to send the CRL

According to our algorithm, there is no extra communication overhead for revocation list, because only the identity of the misbehaving node is sent to the nearest RSU. The other nodes need not check any CRL list before then send their message. The message transmitted is simply the negation of the transmitted message which involves the same cost. Since CRL is not needed in our scheme, communication overhead is greatly reduced as compared to other schemes. In Fig 4.5, we show the communication overhead incurred by different schemes. We consider the Hybrid scheme by Calandriello et al [20], ECMV scheme [76], PASS [70] and LEAVE [60]. We see that LEAVE incurs a high communication overhead, compared with [20], [76] and [70].

4.5 Limitations and Countermeasures

In this section we discuss limitations of our scheme and provide possible solutions to the problems along that line.

4.5.1 Limitations

Our scheme is based on calculating distance between nodes, in order to detect false location information. An exception arises as shown in Figure 4.6. In case the node sends a location information l'_{jt_3} at time t_3, when its current location is l_{jt_3}, then n_i will not be able to conclude that node n_j is lying because it is equidistant from both the locations l'_{jt_3} and l_{jt_3}. Similar is the case for all vehicles which lie on the line Z. However, if there is another node n_r (with pseudonym p_{rt_3}, as shown), then it will be able to conclude that n_j is lying about its position. If the RSU is not on Z, then it is able to understand that n_j is lying and it will take action against it. In the rare case if the RSU is also on the line Z, it cannot predict correctly based on one node's correct observation.

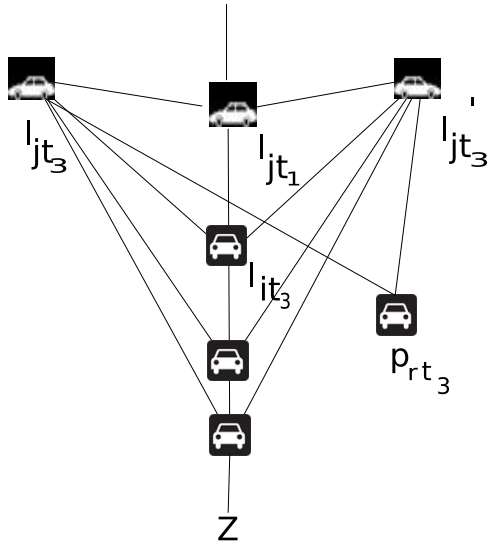

Figure 4.6: Exception when many cars are equidistant from the true and correct positions of n_j

A situation where our scheme will fail to give correct results is when a node takes a U-turn. In such scenario, it might seem that a node is giving false location information, because its location is far behind the expected location. It might seem

that the vehicle has moved backwards, which means its location information is faulty. The other situation is when a node is moving on a flyover with loops, the actual distance might be quite different from the Euclidean distance $dist(u, v)$, between two vehicles u and v. We leave it as an open problem.

If nodes are moving in a group. One node might aid the other by sending a wrong alert information. Suppose two nodes n_i and n_j are moving together as a group. Node n_i should take a right turn and node n_j is supposed to go left. Node n_i takes the right turn as per its requirement. Node n_j can then aid node n_i by sending an alert like "hazardous condition on right road" and take the left one as desired. Nodes behind n_j notice the alert and also the subsequent beacon messages and conclude that the alert sent by n_j is correct, following our algorithm. Misbehavior of node n_j will go undetected. They might then take the straight road and benefit node n_i. Such issues are left as future work.

Another limitation is that, since each node retransmits the alert message to the others, the same alert message will be retransmitted by several nodes, using more bandwidth than necessary. How to efficiently handle this problem is left as a future work.

4.5.2 Incentivizing Nodes

We assume that nodes which sense or receive alerts, immediately transmit them to other nodes. However transmitting such alerts leads to power consumption. So, it is natural to ask why would a node be motivated to send out such alerts, when its only goal is to save itself. One way to overcome this problem is to give incentives to nodes which co-operate in transmitting useful messages so that every transmitted message M_A and M_R is noted. The node p_{jt} is given an incentive, in terms of positive points

is it sends out correct information and given negative points, if it sends out wrong information. The points can be suitably adjusted as providing free service for points above a certain threshold.

4.5.3 Change of Direction

We have assumed that nodes move in the same direction. However, a node moving in the opposite direction can send out false alert message. Such nodes might not do so for selfish reason, but have malicious intentions. We leave this problem open for future research.

4.6 Conclusions

We proposed a new scheme of misbehavior detection for VANET and introduced the concept of data-centric MDS, where we are more interested in finding out false information rather than classifying nodes as "malign" and "benign". The main reason for this is that nodes misbehave mainly because of selfish reasons and need not be classified as "malign" or "benign" nodes. Our scheme provides location privacy by the use of pseudonyms. Any node can detect false alert information by observing the location of the node after issuing an alert. There is no need of voting and majority decisions. This makes our scheme resilient to Sybil attacks. We do not revoke nodes completely, but simply impose fines depending false message sent out. The simulation results show that our MDS is effective in the situation of straight roads.

Chapter 5

Conclusions and Future Work

In this thesis, we discussed the existing reputation and misbehavior detection schemes in VANET. We also show the drawbacks of these schemes.

We then introduced the concept of information cascading and oversampling and then a novel voting scheme to decrease the impact of these two phenomenon. Our scheme adds weight factor into each vehicle's opinion based on the distance to the event. The first set of vehicles which can observe the event directly have the highest weight factor value. Our simulations show that if we just consider the first set vehicle's opinions and ignore the decisions of other vehicles, the incorrect decision is minimum which means the impact of information cascading is reduced.

Although we addressed the problem of information cascading, the following questions still remain: When do the vehicles make the decision? Do they make a decision immediately after receiving the messages? Or do they wait for a little interval to collect the opinions of other vehicles? In real life situations, there are delays in transmissions. If we make decisions immediately, we might lose important messages to support the voting. However, if we wait for some time before voting, there is also a problem: during this interval, vehicles might receive some incorrect messages. This

will adversely affect the decision. Our future work will try to find a trade off of the time when to make the decision for each vehicle.

In voting scheme, the Sybil attack is hard to avoid since the pseudonym is applied. We presented a new VANET model which is based on data-centric misbehavior detection. The main characteristic of this new scheme is that we can focus on finding the false information rather than classifying the vehicles as good or bad. We detect the misbehavior by our simple location detection scheme, and there is no need of voting and majority decisions. This makes our scheme resilient to Sybil attacks. Also, we do not revoke the misbehavior vehicle; instead, some fines are applied to the malicious car drivers to restrain them from misbehaving for selfish reasons. The simulation results illustrate that our scheme is effective on the straight road.

As we discussed, there are still some limitations in our scheme. More effort is required to make the scheme more effective and feasible. This will be part of the future work.

References

[1] The cnet network simulator, available at http://www.csse.uwa.edu.au/cnet/.

[2] http://en.wikipedia.org/wiki/information_cascade/.

[3] http://web.mit.edu/newsoffice/2010/crowd-wisdom-1115.html.

[4] http://www-fars.nhtsa.dot.gov/main/index.aspx.

[5] http://www.mps.gov.cn/n16/n85753/n 85870/ index.html.

[6] The network simulator - ns-2,available at http://www.isi.edu/nsnam/ns.

[7] The ptv simulation-vissim,available at http://www.english.ptv.de/cgi-bin/traffic/traf vissim.pl.

[8] The qualnet software, available at http://www.scalable-networks.com/.

[9] The sumo trafc simulation package, available at http://sumo.sourceforge.net/index.shtml.

[10] The transmodeler trafc simulator, available at http://www.caliper.com/transmodeler/.

[11] Daron Acemoglu, Munther A. Dahleh, Ilan Lobel, and Asuman Ozdaglar. Bayesian learning in social networks, 2010. Available at http://web.mit.edu/dahleh/www/pubs/socialnetworks.pdf.

[12] William Joseph Adams and Nathaniel J. Davis. Toward a decentralized trust-based access control system for dynamic collaboration. In *IEEE Workshop on Information Assurance*, pages 317–324, 2005.

[13] Tiranuch Anantvalee and Jie Wu. *Wireless/Mobile Network Security*, chapter A Survey on Intrusion Detection in Mobile Ad Hoc Networks. Springer-Verlag, 2008.

[14] Fan Bai, Hariharan Krishnan, Varsha Sadekar, Gavin Holl, and Tamer Elbatt. Towards characterizing and classifying communication-based automotive applications from a wireless networking perspective. In *Proceedings of IEEE Workshop on Automotive Networking and Applications (AutoNet)*, 2006.

[15] Igor Bilogrevic, Mohammad Hossein Manshaei, Maxim Raya, and Jean-Pierre Hubaux. Optimal revocations in ephemeral networks: A game-theoretic framework. In *8th International Symposium on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks, University of Avignon, Avignon, France*, pages 21–30, 2010.

[16] Subir Biswas, Md. Mahbubul Haque, and Jelena V. Misic. Privacy and anonymity in vanets: A contemporary study. *Ad Hoc & Sensor Wireless Networks*, 10(2-3):177–192, 2010.

[17] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[18] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology,24th Annual International Cryptology Conference, Santa Barbara, California, USA*, pages 41–55, 2004.

[19] Levente Buttyán, Tamás Holczer, and István Vajda. On the effectiveness of changing pseudonyms to provide location privacy in vanets. In *Security and Privacy in Ad-Hoc and Sensor Networks, 4th European Workshop, ESAS*, pages 129–141, 2007.

[20] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and robust pseudonymous authentication in vanet. In *Vehicular Ad Hoc Networks*, pages 19–28, 2007.

[21] David Chaum and Eugène van Heyst. Group signatures. In *EUROCRYPT*, pages 257–265, 1991.

[22] Thomas M. Chen and Varadharajan Venkataramanan. Dempster-shafer theory for intrusion detection in ad hoc networks. *IEEE Internet Computing*, 9(6):35–41, 2005.

[23] John R. Douceur. The sybil attack. In *Peer-to-Peer Systems,First International Workshop,Cambridge,MA,USA,Revised Papers*, volume 2429, pages 251–260, 2002.

[24] Florian Dtzer, Lars Fischer, and Przemyslaw Magiera. Vars: A vehicle ad-hoc network reputation system. In *Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2005.

[25] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[26] Laurent Eschenauer, Virgil D. Gligor, and John Baras. On trust establishment in mobile ad-hoc networks. In *Proceedings of the Security Protocols Workshop*, pages 47–66. Springer-Verlag, 2002.

[27] Julien Freudiger, Mohammad Hossein Manshaei, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. On the age of pseudonyms in mobile ad hoc networks. In *29th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies,San Diego,CA,USA*, pages 1577 – 1585, 2010.

[28] Julien Freudiger, Maxim Raya, Mrk Felegyhazi, Panos Papadimitratos, and Jean-Pierre Hubeax. Mix-zones for location privacy in vehicular networks. In *Proceedings of WiN-ITS*, 2007.

[29] Matthias Gerlach. Trust for vehicular applications. *International Symposium on Autonomous Decentralized Systems*, pages 295–304, 2007.

[30] Mainak Ghosh, Anitha Varghese, Arobinda Gupta, Arzad Alam Kherani, and Skanda N. Muthaiah. Detecting misbehaviors in vanet with integrated root-cause analysis. *Ad Hoc Networks*, 8(7):778–790, 2010.

[31] Philippe Golle, Daniel H. Greene, and Jessica Staddon. Detecting and correcting malicious data in vanets. In Kenneth P. Laberteaux, Raja Sengupta, Chen-Nee Chuah, and Daniel Jiang, editors, *Vehicular Ad Hoc Networks*, pages 29–37. ACM, 2004.

[32] Gilles Guette and Bertrand Ducourthial. On the sybil attack detection in vanet. *IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–6, 2007.

[33] Jinhua Guo, J.P Baugh, and Shengquan Wang. A group signature based secure and privacy-preserving vehicular communication framework. *Mobile Networking for Vehicular Environments*, pages 103–108, 2007.

[34] Jason J. Haas, Yih chun Hu, and Kenneth P. Laberteaux. Design and analysis of a lightweight certificate revocation mechanism for vanet. In *Proceedings of the sixth ACM international workshop on VehiculAr InterNETworking*, pages 89–98, 2009.

[35] Aamir Hassan. *VANET Simulation Master Thesis*. Halmstad University, 2009.

[36] Zhen Huang, Sushmita Ruj, Marcos Antonio Cavenaghi, and Amiya Nayak. Limitations of trust management schemes in vanets and countermeasures. *22nd IEEE PIMRC, Toronto, Canada*, 2011.

[37] Jean-Pierre Hubaux, Srdjan Capkun, and Jun Luo. The security and privacy of smart vehicles. *IEEE Security & Privacy*, 2(3):49–55, 2004.

[38] Yixin Jiang, Minghui Shi, Xuemin Shen, and Chuang Lin. Bat: A robust signature scheme for vehicular networks using binary authentication tree. *IEEE Trans. Wireless Communications*, 8(4):1974 – 1983, 2009.

[39] F. Kargl, P. Papadimitratos, L. Buttyan, M. Mter, E. Schoch, B. Wiedersheim, T. v. Thong, G. Cal, A. Held, A. Kung, and J. p. Hubaux. Secure vehicular communication systems: Implementation, performance, and research challenges. In *IEEE Wireless Communication Magazine*, pages 110–118, 2008.

[40] Sanjay K.Dhurandher, Mohammad S.Obaidat, Amrit Jaiswal, Akanksha Tiwari, and Ankur Tyagi. Securing vehicular networks a reputation and plausibility checks-based approach. *IEEE GLOBECOM Workshops*, pages 1550–1554, 2010.

[41] Timo Kosch. Local danger warning based on vehicle ad-hoc networks: Prototype and simulation. *International Workshop on Intelligent Transportation*, 2004.

[42] Kenneth P. Laberteaux, Jason J. Haas, and Yih-Chun Hu. Security certificate revocation list distribution for vanet. In *Proceedings of the Fifth International Workshop on Vehicular Ad Hoc Networks,San Francisco, California,USA*, pages 88–89, 2008.

[43] Bisheng Liu, Jerry T. Chiang, and Yih-Chun Hu. Limits on revocation in vanets. *8th International Conference on Applied Cryptography and Network Security*, pages 38–52, 2010.

[44] Nai-Wei Lo and Hsiao-Chien Tsai. A reputation system for traffic safety event on vehicular ad hoc networks. *EURASIP J. Wireless Comm. and Networking. Data of Issue:10.1155/2009/125348*, 2009.

[45] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Pin-Han Ho, and Xuemin Shen. Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications. In *27th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies,Phoenix,AZ, USA*, pages 1229–1237, 2008.

[46] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *International Conference on Mobile Computing and Networking*, pages 255–265, 2000.

[47] Pietro Michiardi and Refik Molva. Core: A collaborative reputation mechanism to enforce node cooperation. In *Mobile Ad Hoc Networks. Communication and Multimedia Security*, pages 107–121, 2002.

[48] Umar Minhas, Jie Zhang, Thomas Tran, and Robin Cohen. *Intelligent Agents in Mobile Vehicular Ad-Hoc Networks: Leveraging Trust Modeling Based on Direct Experience with Incentives for Honesty*, pages 243–247, 2010.

[49] Tyler Moore, Jolyon Clulow, Shishir Nagaraja, and Ross Anderson. New strategies for revocation in ad-hoc networks. In *Security and Privacy in Ad-hoc and Sensor Networks, 4th European Workshop*, pages 232–246, 2007.

[50] Tyler Moore, Maxim Raya, Jolyon Clulow, Panagiotis Papadimitratos, Ross Anderson, and Jean-Pierre Hubaux. Fast exclusion of errant devices from vehicular networks. In *Proceedings of the Fifth Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks,Crowne Plaza, San Francisco, California, USA*, pages 135–143, 2008.

[51] James Newsome, Elaine Shi, Dawn Song, and Adrian Perrig. The sybil attack in sensor networks: analysis & defenses. In *Proceedings of the Third International Symposium on Information Processing in Sensor Networks, Berkeley, California,USA*, pages 259–268, 2004.

[52] Benedikt Ostermaier, Florian Dtzer, and Markus Strassberger. Enhancing the security of local danger warnings in vanets - a simulative analysis of voting schemes. In *Proceedings of the Second International Conference on Availability, Reliability and Security*, pages 422–431, 2007.

[53] P. Papadimitratos, L. Buttyan, T. Holczer, E. Schoch, J. Freudiger, M. Raya, Z. Ma, F. Kargl, A. Kung, and J. p. Hubaux. Secure vehicular communication systems: design and architecture. In *IEEE Wireless Communication Magazine*, pages 100–109, 2008.

[54] Panagiotis Papadimitratos, Ghita Mezzour, and Jean-Pierre Hubaux. Certificate revocation list distribution in vehicular communication systems. In *Proceedings of the Fifth International Workshop on Vehicular Ad Hoc Networks, San Francisco, California, USA*, pages 86–87, 2008.

[55] Soyoung Park, Baber Aslam, Damla Turgut, and Cliff C. Zou. Defense against sybil attack in vehicular ad hoc network based on roadside unit support. In *MILCOM*, pages 1–7, 2009.

[56] Bryan Parno and Adrian Perrig. Challenges in security vehicular networks. In *HotNets-IV*, 2005.

[57] Maxim Raya. *Data-Centric Trust in Ephemeral Networks. Ph D Thesis*. EPFL, Lausanne, 2009.

[58] Maxim Raya and Jean-Pierre Hubaux. Securing vehicular ad hoc networks. *Journal of Computer Security*, 15(1):39–68, 2007.

[59] Maxim Raya, Mohammad Hossein Manshaei, Márk Félegyházi, and Jean-Pierre Hubaux. Revocation games in ephemeral networks. In *Proceedings of the 2008 ACM Conference on Computer and Communications Security, Alexandria, Virginia, USA*, pages 199–210, 2008.

[60] Maxim Raya, Panagiotis Papadimitratos, Imad Aad, Daniel Jungels, and Jean-Pierre Hubaux. Eviction of misbehaving and faulty nodes in vehicular networks. *IEEE Journal on Selected Areas in Communications*, 25(8):1557–1568, 2007.

[61] Maxim Raya, Panagiotis Papadimitratos, Virgil D. Gligor, and Jean pierre Hubaux. On datacentric trust establishment in ephemeral ad hoc networks. In *IEEE INFOCOM*, 2008.

[62] Paul Resnick and Richard Zeckhauser. Trust among strangers in Internet transactions: Empirical analysis of eBay's reputation system. In Michael R. Baye, editor, *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics*, pages 127–157. Elsevier Science, 2002.

[63] Sushmita Ruj, Marcos Antonio Cavenaghi, Zhen Huang, Amiya Nayak, and Ivan Stojmenovic. Data-centric misbehavior detection in vanets. *IEEE Vehicular Technology Society, San Francisco, USA*, 2011.

[64] Sushmita Ruj, Zhen Huang, Marcos Antonio Cavenaghi, Amiya Nayak, and Ivan Stojmenovic. Mcmds: Data-centric misbehavior detection scheme in vanets. In *Submitted to IEEE Trans. on Vehicular Technology*, 2011.

[65] Krishna Sampigethaya, Leping Huang, Mingyan Li, Radha Poovendran, Kanta Matsuura, and Kaoru Sezaki. Caravan: Providing location privacy for vanet. In *Proc. of the Workshop on Embedded Security in Cars (ESCAR)*, 2005.

[66] Krishna Sampigethaya, Mingyan Li, Leping Huang, and Radha Poovendran. Amoeba: Robust location privacy scheme for vanet. *IEEE Journal on Selected Areas in Communications*, 25(8):1569–1589, 2007.

[67] Ahren Studer, Mark Luk, and Adrian Perrig. Efficient mechanisms to provide convoy member and vehicle sequence authentication in vanets. In *SecureComm*, pages 422–432, 2007.

[68] Ahren Studer, Elaine Shi, Fan Bai, and Adrian Perrig. Efficient mechanisms to provide convoy tacking together efficient authentication revocation, and privacy in vanets. In *SECON 2009*, pages 1–9, 2009.

[69] Yan Lindsay Sun, Zhu Han, and K. J. Ray Liu. Information theoretic framework of trust modeling and evaluation for ad hoc networks. *IEEE Journal on Selected Area in Communications*, 24(2):305–317, 2006.

[70] Yipin Sun, Rongxing Lu, Xiaodong Lin, Xuemin Shen, and Jinshu Su. An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications. *IEEE Trans. on Vehicular Technology*, 59(7):3589–3603, 2010.

[71] Ashish Vulimiri, Arobinda Gupta, Pramit Roy, Skanda N. Muthaiah, and Arzad Alam Kherani. Application of secondary information for misbehavior detection in vanets. In *Networking*, pages 385–396, 2010.

[72] Bo Wang, Sohraab Soltani, Jonathan K. Shapiro, and Pang ning Tan. Local detection of selfish routing behavior in ad hoc networks. In *International Symposium on Parallel Architectures, Algorithms and Networks (I-SPAN), Las Vegas*, pages 392–399, 2005.

[73] Shie-Yuan Wang, Chih-Liang Chou, Chih-Che Lin, and Chih-Hua Huang. The protocol developer manual for the nctuns 5.0 network simulator and emulator. avaliable at http://nsl.csie.nctu.edu.tw/nctuns.html.

[74] S.Y. Wang and C.L. Chou. Nctuns simulator for wireless vehicular ad hoc network research. available at http://nsl.csie.nctu.edu.tw/nctuns.html.

[75] S.Y. Wang, C.L. Chou, Y.H. Chiu, Y.S. Tseng, M.S. Hsu, Y.W. Cheng, W.L. Liu, and T.W. Ho. Nctuns 4.0: An integrated simulation platform for vehicular traffic, communication, and network researches. *IEEE International Symposium on Wireless Vehicular Communications*, 2007.

[76] Albert Wasef, Yixin Jiang, and Xuemin Shen. Ecmv: Efficient certificate management scheme for vehicular networks. In *Proceedings of the Global Communications Conference, New Orleans, LA, USA*, pages 639–643, 2008.

[77] Albert Wasef, Yixin Jiang, and Xuemin Shen. Dcs: An efcient distributed certicate service scheme for vehicular networks. *IEEE Trans. Vehicular Technology*, 59(2):533–549, 2010.

[78] Bin Xiao, Bo Yu, and Chuanshan Gao. Detection and localization of sybil nodes in vanets. In *Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks*, pages 1–8, 2006.

[79] Jie Zhang. A survey on trust management for vanets. In *25th IEEE International Conference on Advanced Information Networking and Applications, AINA 2011, Biopolis, Singapore*, pages 105–112, 2011.

[80] Jie Zhang and Robin Cohen. Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings. In *8th International Conference on Electronic Commerce: The new e-commerce - Innovations for Conquering Current Barriers, Obstacles and Limitations to Conducting Successful Business on the Internet*, pages 225–234, 2006.

[81] Tong Zhou, Romit Roy Choudhury, Peng Ning, and Krishnendu Chakrabarty. Privacy-preserving detection of sybil attacks in vehicular ad hoc networks. In *4th Annual International Conference on Mobile and Ubiquitous Systems, Philadelphia, PA, USA*, pages 1–8, 2007.

[82] C. Zouridaki, B. L. Mark, and M. Hejmo. A quantitative trust establishment framework for reliable data packet delivery in manets. In *Proceedings of the*

Third ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN), pages 1–10. ACM, 2005.