Providing VANET Security Through Active Position Detection

Gongjun Yan, Stephan Olariu, Michele C. Weigle

Department of Computer Science, Old Dominion University, Norfolk, VA 23529-0162, USA

Abstract

Vehicle position is one of the most valuable pieces of information in a Vehicular Ad-hoc NETwork (VANET). The main contribution of this work is a novel approach to enhancing position security in VANETs. We achieve local security by enlisting the help of on-board radar to detect neighboring vehicles and to confirm their announced coordinates. Local security is extended to achieve global security by using preset position-based groups to create a communication network and by using a dynamic challenging mechanism to confirm remote position information. Our solution is predicated on the widely accepted assumption that the vast majority of vehicles are honest and behave responsively. Extensive simulations confirm the quality of the proposed solution by measuring how fast compromised vehicles can be detected under various conditions.

Key words: security, position, Sybil attack, radar, GPS, VANET

1 Introduction

The past decade has witnessed the emergence of Vehicular Ad-hoc Networks (VANETs), specializing the well-known Mobile Ad Hoc Networks (MANETs) to Vehicle-to-Vehicle and Vehicle-to-Roadside wireless communications. The importance and potential societal impact of VANETs has been confirmed by the rapid proliferation of consortia involving car manufacturers, government agencies and academia. Examples include the Car-2-Car Communication Consortium [1], the Vehicle Safety Communications Consortium [2], and Honda's

Email address: (ygongjun, olariu, mweigle)@cs.odu.edu (Gongjun Yan, Stephan Olariu, Michele C. Weigle).

URL: www.cs.odu.edu/∼(ygongjun, olariu, mweigle) (Gongjun Yan, Stephan Olariu, Michele C. Weigle).

Advanced Safety Vehicle Program [3], among others. While the original motivation for VANETs was to promote traffic safety, more recently privacy and security concerns have received much attention in the literature [4,5]. In addition, it is increasingly obvious that VANETs open new vistas for Internet access, distributed gaming, and the fast-growing mobile entertainment industry.

The introduction of new vehicular devices can greatly enhance security in VANETs. Indeed, the provision of on-board Global Positioning System (GPS) devices has revolutionized driving. Similarly, the recent introduction of short-range radar on some top-of-the-line models promises to reduce the number of fender-benders and other accidents. Interestingly, on-board radar is also used in advanced cruise control systems [6]. It is natural, therefore, to enlist the help of these devices for the purpose of enhancing the security of the information flow in VANETs. A classic example of "anti-social" behavior in VANETs is for malicious cars to fake their true positions. Our main contribution is to show that by using GPS and radar-provided information, one can ensure the validity of position information in a VANET and can detect and isolate malicious cars.

Although all VANET applications require some form of security, we divide them into two categories: (1) Non-position related applications, such as online payment services, online shopping, and the like; these applications focus on network access, for example from an ad hoc wireless network to the Internet. (2) Position-related applications: including traffic condition reports, collision avoidance, emergency alert, cooperative driving, or resource availability. The key ingredient in this category is accurate position information. If position information is not protected, these applications may not work at all.

Since position information is very important, adversaries, such as pranksters and malicious attackers, could harm the VANET by perpetrating the following attacks [7,8,9,10,5,11,12]:

- *Dropping packets*: In the presence of an accident, an attacker may drop all of the alerts to prevent appropriate deceleration alerts from reaching other vehicles.
- *Modifying existing packets or inserting bogus packets*: A prankster may create the illusion of a traffic jam before selecting an alternate route to his advantage.
- *Replaying packets*: A malicious user may pretend to be at a fake position to create the illusion of an actual vehicle.

Another well-known attack is the Sybil attack [7] which is launched by forging multiple identities. These false identities will give the illusion that there are additional vehicles on the roadway, which may have a serious effect on a VANET. For example, a collision warning application given fake vehicle po-

sitions may be tricked into thinking that an accident is imminent, prompting the driver to brake quickly, possibly causing a real accident.

In this paper, we propose a novel solution to prevent most of these attacks. This work was motivated by the need to provide secure topology information in VANETs and to build a secure network for applications, such as a congestion alert system. Underlying our solution is the famous adage: "*Seeing is believing*". We use on-board radar as the virtual "eye" of a vehicle. Although the "eyesight" is limited due to a modest radar transmission range, a vehicle can "see" surrounding vehicles and "hear" reports of their GPS coordinates. By comparing what is seen to what has been heard, a vehicle can corroborate the real position of neighbors and isolate malicious vehicles to achieve local security. We expect the on-board radar device to provide useful corroboration of reported location information, except for during short transient periods. For example, the line-of-sight that radar needs may be temporarily obstructed by a large truck. Due to the dynamic nature of traffic, even if there are transient obstructions, the line of sight will be restored eventually.

We use preset position-based cells as a basis for our approach. Each vehicle in the cell can directly communicate with every other vehicle in the cell. To achieve local security, a vehicle may use its radar to verify its neighbors' positions or issue queries to verify the position of a specified vehicle in the cell. In this way, each vehicle in the cell knows the position of the other vehicles in the cell with high certainty.

Because the on-board radar is not strong enough to verify vehicles in remote cells, we propose a method to challenge and confirm the position of a vehicle in a remote cell by using the on-board radar in oncoming traffic. Once remote data has been verified, each vehicle has three types of data in hand: local radar-detected data, remote radar-detected data from oncoming traffic, and agreed-upon data from cell neighbors. To achieve global security, we apply cosine similarity [13] to these three types of data. If the similarity value is beyond a threshold, we accept the data, otherwise it is ignored. With these accepted data, we can construct a history of vehicle movement. The movement history can help determine whether newly received data is valid or not. We isolate the vehicles that send invalid data. This isolation can help to prevent a large number of position-related attacks (forging positions), Sybil attacks (forging identities), and most combinations of position and Sybil attacks.

The remainder of this paper is organized as follows. In Section 2, we provide a review of related work on position security in VANETs. We present our system model in Section 3, describe how local security can be achieved in Section 4, and discuss global security in Section 5. Section 6 describes how attacks can be prevented. Section 7 presents our method to isolate malicious vehicles. In Section 8, we give an example of how our method can thwart an example

attack. Finally, in Section 9, we present the results of our evaluation study, focusing on how quickly malicious nodes can be detected.

2 Related Work

Previous work on position security and Sybil attacks can be divided into three categories: cryptography-based, radio signal-based, and resource-based.

Cryptography-based methods ensure the reliability of the position and identities claimed by vehicles through encryption. Most of the published work on position security focuses on using Public Key Infrastructure (PKI) [14,15,11,16,17,18] and digital signatures [19,20,5,21,11]. While these solutions provide security, they add significant overhead to the system. The algorithms involved in encrypting and decrypting the messages along with the issue of distributing public keys and their certificates makes the system complex. In this paper, we take a different approach. We allow vehicles to send the information in plain-text and depend upon receivers to verify the information.

Hubaux et al. [14] addressed a novel vehicle model, called the "*smart vehicle*". Inspired by this smart vehicle model, we use a similar model. The smart vehicle is composed of an Event Data Recorder (EDR), a GPS receiver, a front-end radar for detecting obstacles at distances as far as 200 meters, an electronic license plate, a display and a computing platform. The authors focused on improving security and privacy and sketched two solutions to these problems by using PKI. As opposed to their approach, we focus on location security not general message/data security.

Leinmüller et al. [10] proposed a method to secure position information by using hard thresholds to detect false locations. Vehicles monitor data to verify the reported position. If the reported position lies beyond a threshold, the location is determined to be false. If the number of nodes is larger than a threshold maximum number of nodes, the honest nodes know that there are some fake nodes. Although the authors do not use any other devices or hardware, the accuracy and efficiency are difficult to guarantee. In addition, this method is not flexible because of uncertainties involved in VANETs.

Dai and Wu [22] proposed a hybrid mechanism to maintain the routing path both by proactive and reactive mechanisms. Though this proposal involves low overhead, there is no emphasis on securing the topology. Also, the topology-based routing algorithm is not as efficient as position-based routing for establishing connectivity between vehicles [12].

Leinmüller et al. [9,12] describe the effect of false position information in

VANETs. The authors show that the effect of malicious nodes is more severe in highway scenarios than in city scenarios. In their simulations, the delivery ratio decreases by 90% to 100% if malicious nodes simply drop messages.

Golle et al. [8] proposed a method to detect and correct malicious data. This method is based on building a model for the VANETs against which the data can be verified for consistency. Our approach, on the other hand, prevents malicious data from entering the system. It thwarts any attempt to inject wrong information at the origin and such spurious messages do not propagate further.

Radio signal-based methods [23,24,25] determine false claimed positions based on the received signal power. The basic idea of this method is that the distance between nodes can be computed from the received signal power. If there is a vehicle at a position which does not match the distance computed from the received radio power, this nodes is determined to be a fake. However, a malicious node can use the same method to compute the transmission signal power to fool other nodes. Besides, radio may bounce off of vehicles and other obstacles. Detection based on these bounced radio signals may not be accurate.

Resource-based methods test vehicles' resources, such as radio resources [26,8], computational resources (vehicles failing to solve a puzzle are identified as fakes) [7], and identification resources (vehicles whose MAC and IP addresses are not recorded in a profile are identified as fakes) [27]. Newsome et al. [26] claimed that the method for detecting Sybil attacks proposed by Douceur [7] is not applicable to ad hoc networks and proposed other prevention methods including: radio resource testing, registration, and position verification. Radio resource testing is based on the assumption that no device can send and receive on more than one channel at a time. But, attackers may have multiple channels. Registration does not apply to Sybil attacks, because the attackers can simply create multiple identities. Besides, registration creates a privacy concern. Position verification relies on roadside infrastructure, like base stations. Piro et al. [27] records vehicles' MAC and IP addresses, as a passive ID, to create a profile about neighbors. However, attackers may have multiple devices to defeat this method. Moreover, privacy is an issue if MAC and IP addresses are recorded and tracked.

We incorporate the location-based group formation proposed by Raya et al. [15] to reduce the overhead of dynamically forming the groups. Each vehicle knows its location via GPS and can map the coordinates to determine its appropriate group. Simulation results in that paper proved that having location-based groups are more efficient than using dynamic groups.

3 System Model

3.1 Vehicle Model

An important new concept in the automotive industry is neighborhood awareness. This allows a vehicle to know about the presence, location and even speed of neighboring vehicles. Today, new vehicles may have computer network devices, computing devices, storage devices, and an EDR. Specifically, vehicles represented in this paper are assumed to be endowed with the following features:

- A GPS navigation system, including a GPS receiver and GPS maps.
- A front and a rear radar [28,29], such as microwave front radar and a short-distance infrared or ultrasonic radar. We assume that the omni-directional front radar can detect neighboring cars within line of sight in a radius of 200 meters. Some cruise control systems already use this kind of radar. Although in some cases, the driver can visually confirm the objects detected by radar, our system does not require human interaction at all.
- A computer center, which will provide data processing, computing and storage.
- A wireless transceiver, using Dedicated Short Range Communications (DSRC) for fast communications; we assume that data can be changed here by attackers for position attacks.
- A unique ID, such as an electronic license plate [14], which is issued by a registration authority annually. We assume that the ID can be changed by the attacker to launch a Sybil attack, for example using a electronic license plate test unit to generate fake IDs.
- A virus checker. We will not discuss virus injection-based attacks.

We base our vehicle model on the smart vehicle proposed by Hubaux et al. [14]. Vehicles with some of these devices (GPS, radar) are already in production. For example, Toyota has developed a Pre-Crash Safety system [29] which uses millimeter-wave radar to sense vehicles and obstacles on the road. Sensor Technologies and Systems developed forward looking vehicle radar [28], which can detect obstacles with a 170 meter range. In August 2006, legislation [30] was passed under the recommendation from the National Highway Traffic Safety Administration that by the year 2011, all automobile manufacturers must openly disclose to the consumer the existence and use of EDR technology in their vehicles. As of October 2006, it has been estimated that 64% of all vehicles sold in the United States have EDRs [31]. Furthermore, GPS and a computing center are popular vehicle accessories today. Since these components are already being installed in vehicles, there is no additional cost required to deploy our position security techniques.

In accord with other works, we assume that the majority of cars (about 85%) are honest [32].

We use radar to detect the physical parameters of neighboring vehicles as a first priority data resource, use the oncoming traffic radar to detect physical parameters of non-neighboring vehicles as a second priority data resource and use neighbors' reports as the third priority data resource. We apply cosine similarity to these data to build a history of movements. Based on the history of movements, we can screen forged data from the real data to solve most of these attacks. Vehicles fall into one of three levels of trust: trusted, questioned and untrusted. We use tables to classify each vehicle's level of trust and use these tables to isolate malicious vehicles. We simulate a position-related solution which shows efficient results. Simulations for the Sybil attack and combinations of Sybil attacks and position attacks are planned as our future work.

3.2 Network Model

Network communication can be modeled as in a city scenario or a highway scenario. We use a highway scenario since Leinmüller et al. [9,12] showed that the effects of malicious nodes in a highway scenario are worse than in a city scenario.

3.2.1 Network Cells

There are two types of cells that have been proposed in the literature [15,16]: dynamic cells and position-based cells. Although dynamic cells are flexible, they are not efficient. Position-based cells, on the other hand, are created beforehand, and vehicles use their GPS coordinates to map to their respective cells. These preset cells avoid the need to undergo the complex process of forming a cell and electing a cell leader. In this paper, we use position-based cells to build a communication network. Cells are shown in Figure 1. The center of a cell is at the median between the two directions of traffic.

Our network maintenance is hybrid comprising both proactive [22] and a reactive [33,34] components. For intra-cell maintenance, vehicles proactively broadcast their GPS coordinates. The frequency of the broadcast, f, depends upon the vehicle's velocity (meters per second)

$$f = \alpha * \left| \nu - \frac{\nu_{max}}{2} \right| \tag{1}$$

where f is the frequency (the number of broadcast packets per second), α is

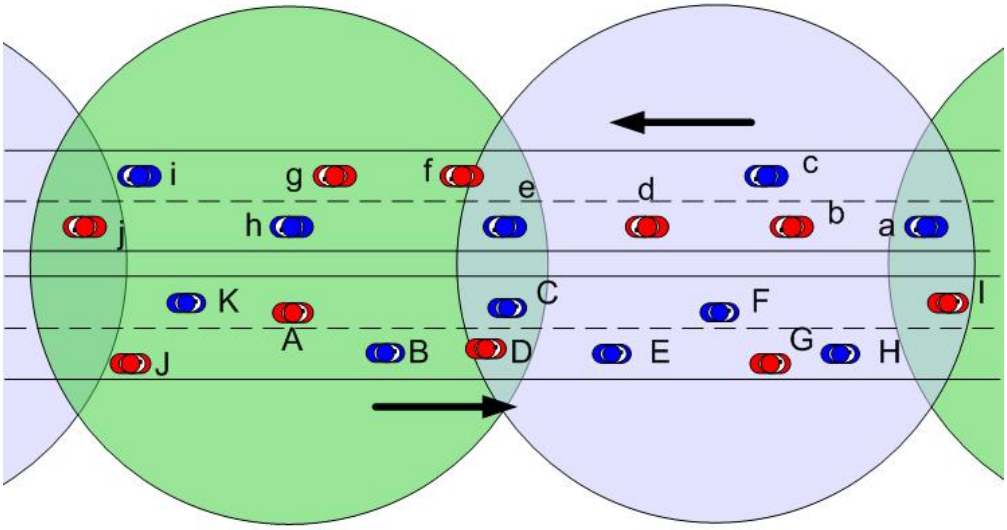

Fig. 1. System model. Shadowed areas are the preset cells on digital maps. Vehicles compare their GPS coordinates with these preset cells to identify their host cells.

a coefficient which is determined by the road characteristics(the number of lanes, the speed limit, etc.), ν is the current velocity of the vehicle, and ν_{max} is the maximum speed. If the vehicle velocity is high, the frequency is high because the position changes quickly and if the vehicle velocity is low, the frequency is low because vehicles do not change position very much.

We only propagate packets in the direction towards the destination. Therefore we need to maintain a rough topology of remote vehicles. Triggered by the events to be discussed in Section 4.1, the observer broadcasts a topology request. Cells receiving this request, usually cell leaders as discussed in Section 3.2.3, will send back topology packets which can be used by the observer to build a rough topology of the network cells.

3.2.2 Formation of Network Cells

We configure the road with virtual digital cells. For example, every 200 meters there is a cell on the road, i.e., the cells' radius is 100 meters. All vehicles inside a cell can receive packets from each other directly. The diameter of cells matches the transmission range of radar, so that all the neighbors inside can be directly detected by radar. The desired amount of overlap between two adjacent cells must be determined before the cells are formed. The size of overlap depends on the size of cells and the road characteristics. If the size of the cells is very large, the overlap may be a small portion of the cells. If the road is a highway, the overlap may be larger to contain more vehicles as potential routers. In this paper, we use a highway model and assume that the cells have a 100 meter radius; therefore we select a cell overlap of about 20-30 meters. When vehicles are close to the overlap area between two cells, they

may be chosen as routing vehicles as will be discussed in Section 3.2.4.

Service vendors first partition the digital map into cells, making sure they overlap and marking the coordinates of the cell center. Then each vehicle determine its cell based on its GPS coordinates and on-board preset digital maps.

For example, the service vendor partitions a highway into cells. The series of cell center coordinates are $(Xc_i, Yc_i), 0 < i < n$. The diameter is 200 meters, and the overlap is 30 meters. A vehicle is at coordinate (x_v, y_v) on the highway. The computer center in the vehicle will find the closest cell center coordinates (Xc_k, Yc_k) to the vehicle's coordinates. If the coordinates satisfy $(x_v - Xc_k)^2 + (y_v - Yc_k)^2 \leq 100^2$, the vehicle is in cell k.

When a vehicle enters a new cell, it provides its unique ID to the cell leader to become a new member of the group. If there is no cell leader, the newcomer takes over as cell leader. Details of this process are discussed in the next section. As with entering, any cell member exiting the cell notifies the cell leader about its departure as soon as it enters the overlap region.

3.2.3 Cell Leader

The main duty of the cell leader is to verify the GPS position of all the vehicles in its cell, aggregate these positions [35,15] and broadcast this data to other vehicles in the cell. In this way, a remote vehicle can know the position of other vehicles in its cell. Based on the knowledge of the whole road, position-related applications, such as collision avoidance, cooperative driving, traffic optimization, and resource service, can be facilitated. The cell leader collects the positions of other vehicles in its cell and exchanges this position information with other cells.

For security reasons, the cell leader is monitored by its neighbors. When the leader sends and receives aggregated position packets, all the members in the cell will compare the positions in the packets based on their knowledge. By remaining silent, they confirm that the packets have not been altered. Otherwise, they broadcast protest packets against the leader. The other neighbors will put the leader and the protestor vehicle into the question table after receiving the protest packet. Then, the opinion of the other neighbors is counted. If the majority of vehicles regard the leader as malicious, the record of the leader is moved to the distrust table as will be discussed in Section 7. Otherwise, the records sent by the leader are placed in trust table.

If a leader is determined to be malicious, or a leader moves away from the center of a cell, the leader can be challenged by another member of the cell. A vehicle challenging to become the new leader must meet one of the following

conditions: 1) be closest to cell center; 2) be approaching the cell leader. Condition 1 has precedence over condition 2. We can use the following formula to compute the score s for each leader candidate,

$$s = C_{dc} * D_c + C_{di} * D_i \tag{2}$$

where D_c is the distance to the cell center and D_i is the traveling direction. If the vehicle is moving towards the center of the cell $D_i = 1$, and if it is moving away from the center $D_i = -1$. We define $C_{dc} = \frac{R}{D_c+1}$ and $C_{di} = R$. The candidate with highest score wins. In cases where the traffic is dense and there is more than one vehicle with the same score, a simple binary countdown algorithm can select the vehicle with highest ID. Referring back to Figure 1, vehicle A is a cell leader because it is closest to the center of the cell. Assume vehicle F with ID 15 and vehicle G with ID 13 are close the center of the cell and both of them compete to be a cell leader. Based on binary countdown, vehicle F wins. Therefore vehicle F is determined to be the cell leader of that cell. This is similar to group leader determination described by Raya et al. [15], but Raya's method determines a group leader based on the position only without discussing the situation where multiple vehicles challenge the cell leader at the same time. A newcomer or vehicle without honest map history (to be discussed in Section 6.5) will not be determined as a cell leader just as a person without credit history can not obtain a loan. This will prevent attackers from changing their ID to become a cell leader using the Sybil attack. There is a possible case that an attacker may win a binary countdown, but using random checks may be a solution, and we will study this in future work. Besides, the cell leader and cell router, described in the next section, may be targets for selective attack, which we will study in the future.

3.2.4 Cell Router

Once a cell leader is determined, it will use cell routers to transmit the aggregated packets. Cell routers achieve inter-cell communication in a secure manner. When a cell router is sending or receiving packets, its neighbors will monitor the packets. If it modifies, inserts bogus information, or drops the packets, its neighbors will detect this by comparing the packet from the router and the original packet since the neighbors receive same packet as the router does. If the neighbors detect that the router is compromised, they will isolate the router in the same manner as cell leaders are isolated.

Determination of cell routers is similar to the determination of a cell leader. The difference is that two cell routers (upstream and downstream) will be determined instead of only one. Routers should be far from the center of the cell and close to the overlapping region. If more than one vehicle satisfies the criteria, a simple binary countdown algorithm can determine the vehicle

with highest ID. A newcomer or vehicle without good map history can not be determined as a cell router.

4 Local Security

Individual cells are the atomic entities that can be secured from most of the position attacks described in this paper. A vehicle in a cell can verify the GPS coordinates received from any other member using radar when radar has line of sight. If the broadcasted coordinates match radar findings, the message is accepted, otherwise the coordinates are removed from the trust table. This way of verifying the position is *local security*.

Similar to the idea behind greedy algorithms, local security is the basis of global security. In a greedy algorithm, we determine local optimal solutions and combine them to get the optimal solution in the global region. In this paper, we use radar to get local security and combine the local security in various regions to get global security.

4.1 Active Position Detection

We can obtain the relative velocity, angle and position to the target object from radar. There are two events that trigger radar detection. One is a timeout threshold. When an observer vehicle does not receive any packets from a observed vehicle after a certain amount of time, a timeout counter will increase by one. If the timeout counter increases beyond a threshold, the observer vehicle will transmit a radar signal to test the observed vehicle's position. Alternatively, radar detection will be triggered at a random moment during the on-going communication with a vehicle. The rational for this latter strategy is to ensure that a trusted vehicle remains trustworthy. To summarize, we combine a proactive and a reactive corroboration using radar, resulting in active position detection.

4.2 Determining Position

4.2.1 Determining GPS Position

In GPS, when satellite radio signals are transmitted, they are distorted by the troposphere and the ionosphere, therefore GPS coordinates have some tolerance. GPS data normally changes in the range of $\Delta x = \pm 10$ meter; $\Delta y = \pm 10$ meter [36]. In Figure 2, we assume that Δx and Δy are always

equal, marked as $\Delta x = \Delta y = \Delta\alpha$. The shadowed region is the set of possible real vehicle positions. We can use (3) to describe this region. We use (x, y) to represent the real position of vehicle in the GPS system.

$$(x - x_{gps})^2 + (y - y_{gps})^2 \leq (\Delta\alpha)^2 \tag{3}$$

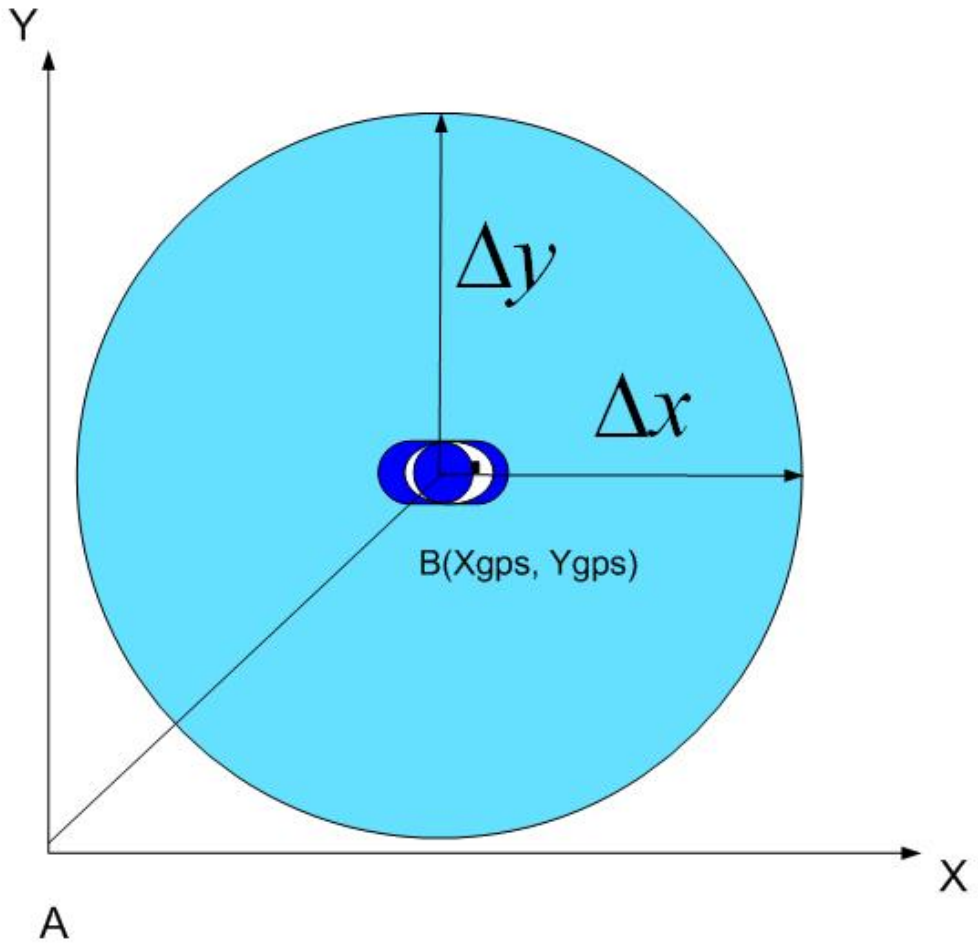

Fig. 2. GPS coordinates position. The GPS tolerance causes a set of real GPS position, shown as a shadow.

4.2.2 Determining Radar-Detected Position

Radar also has tolerance, and we assume that the radar's tolerance includes two parts: angle tolerance $\triangle\theta$ and radius tolerance $\triangle\gamma$, marked as $(\triangle\theta, \triangle\gamma)$. In Figure 3, the shaded region bounded by *HGQFEP* is the set of possible positions of the detected vehicle. We use (x, y) to represent the real position of vehicle and mark the radar readings as (θ, γ). We can use (6) and (7) to describe the two circles: *circle D* and *circle C* in Figure 3.

$$\alpha = \theta - \Delta\theta \tag{4}$$

$$\beta = \theta + \Delta\theta \tag{5}$$

$$(x - \gamma \times \sin\alpha)^2 + (y - \gamma \times \cos\alpha)^2 \leq (\Delta\gamma)^2 \tag{6}$$

$$(x - \gamma \times \sin\beta)^2 + (y - \gamma \times \cos\beta)^2 \leq (\Delta\gamma)^2 \tag{7}$$

Here, θ is the detected angle, starting from 0 degrees North, and γ is the detected radius in meters (distance between vehicle A and vehicle B).

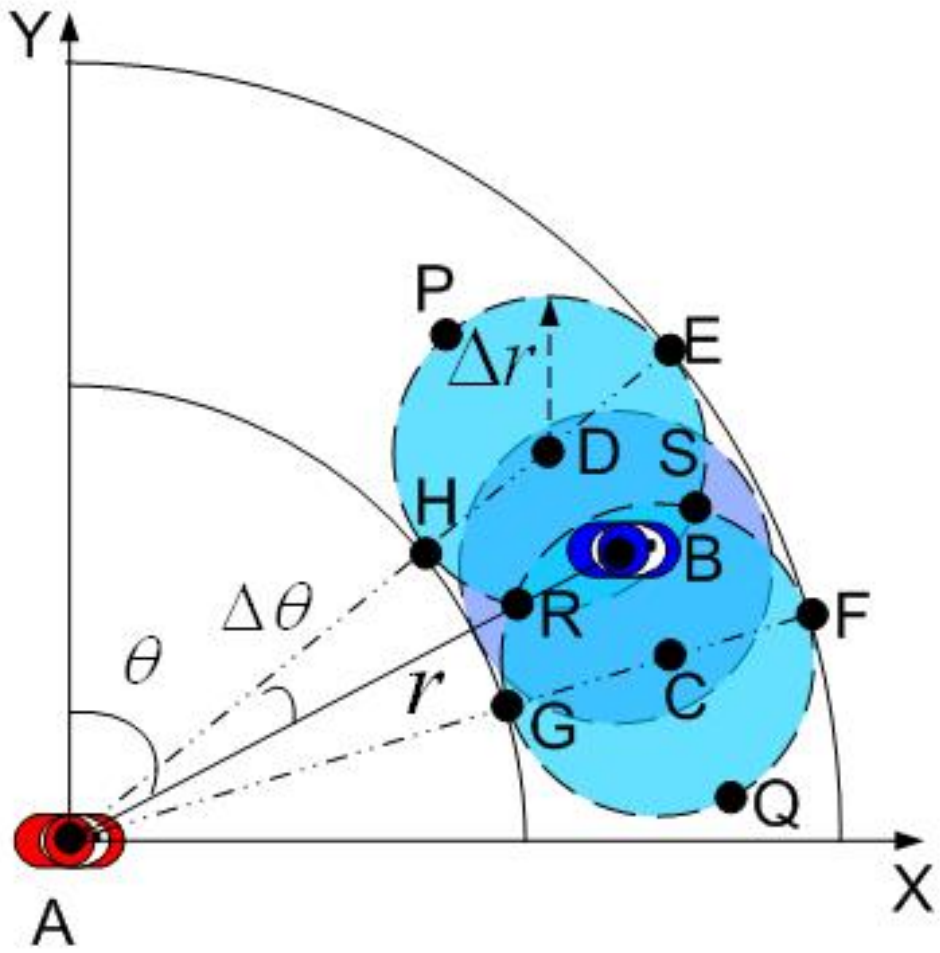

Fig. 3. Radar detected position.

We note that there are two small regions *HRG* and *EBF* where the vehicle could be located, but these regions are not described by (6) or (7). Therefore we use the following formula to describe the region *FCGHDE* in Figure 3:

$$\begin{cases} \gamma - \Delta\gamma \leq \sqrt{x^2 + y^2} \leq \gamma + \Delta\gamma \\ \theta - \Delta\theta \leq \arctan\frac{x}{y} \leq \theta + \Delta\theta \end{cases} \tag{8}$$

Although (8) includes some regions which are described by (6) and (7), for example the region *RGCFB*, this has no negative effect because we will find an intersection between the GPS position formula and radar position formula by using the technique to be described in the next section.

4.2.3 Combining GPS and Radar Coordinates

To draw a conclusion, such as "*my neighbor is lying to me about its position*", we have to find an overlap (solution) between the GPS position formula and the radar position formula.

Without loss of generality, we assume that the real vehicle is at the center of GPS position and radar position, shown as the lightly shaded area in Figure 4. If any of the following combinations has a solution, we can draw a conclusion that the detected vehicle is honest: (3) and (6), (3) and (7), or (3) and (8). Otherwise, the vehicle is determined to be compromised. The meaning of these combinations is shown in Figure 4. If the GPS real position intersects the radar real position region, i.e. if there is an intersection between the GPS position shadow and radar position shadow, this means the GPS real position is very close the value which is detected by radar system. Therefore, we claim that we can accept the GPS position.

5 Global Security

Generally, an adversary may launch a Sybil attack or two types of position attacks: 1) the compromised vehicle continually lies about its position; 2) the compromised vehicle occasionally lies about its position. We can successfully solve the first type of attack using the methods described here.

Locally secured position and speed information needs to be propagated so that other vehicles approaching the cell can benefit from it. We have chosen a cell router for each direction, which is responsible for forwarding this information along with the local traffic situation, to minimize collisions and bandwidth usage. The other approach could be to use the distance from the message source as a factor in determining the next broadcast time. The reason we opted not to use it is because of the fact that in a heavy traffic situations, there can be more than one car at approximately the same distance (taking precision error in account). This would increase the chance of collisions. The determination of cell routers does not add much overhead. Other vehicles in the cell have the responsibility of monitoring the responses of the cell router or cell leader. If either of them tries to change the records or inject wrong information, honest vehicles can notify other members about this compromised cell vehicle and

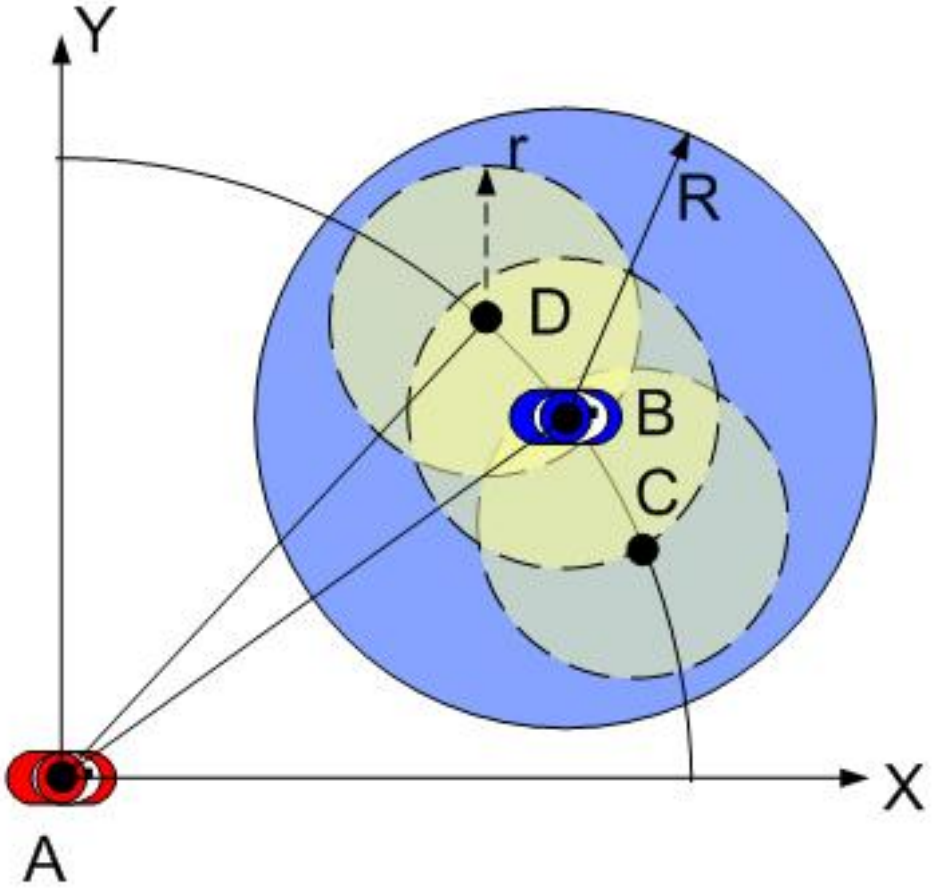

Fig. 4. Confirming GPS coordinates on GPS and radar position, if there is an intersection area between darker shadow (GPS position) and lighter shadow (radar position), we accept the GPS coordinates, otherwise discard it.

initiate the process to determine new cell router or cell leader and broadcast the correct record. There could be a case where the presence of many compromised vehicles might isolate honest vehicles. In order to avoid such cases, we should consider the following facts: (1) it is most likely that majority of the nodes are honest, (2) even if in one cell there are more compromised than honest nodes, it is difficult to maintain such a topology in a VANETs, and (3) to confirm whether the cell router is compromised, vehicles in other cells who have received records from this cell router can run a simple verification test as described below.

Each node uses the packet structure described in Figure 5 to communicate with other vehicles. The *version* field is included for backward compatibility. The *priority* field identifies the message type, for example for emergency requests that need to be propagated immediately. Implementations of priority and version are left for future development. The *hop limit* field prevents the

propagation of the packet past a certain distance. The *source address* includes the vehicles unique ID and cell ID. The *destination address* can be a broadcast address or an ID. The packet body consists of the vehicle's ID, location, direction, speed and time when reported. The timestamp helps to determine the relevance of the message.

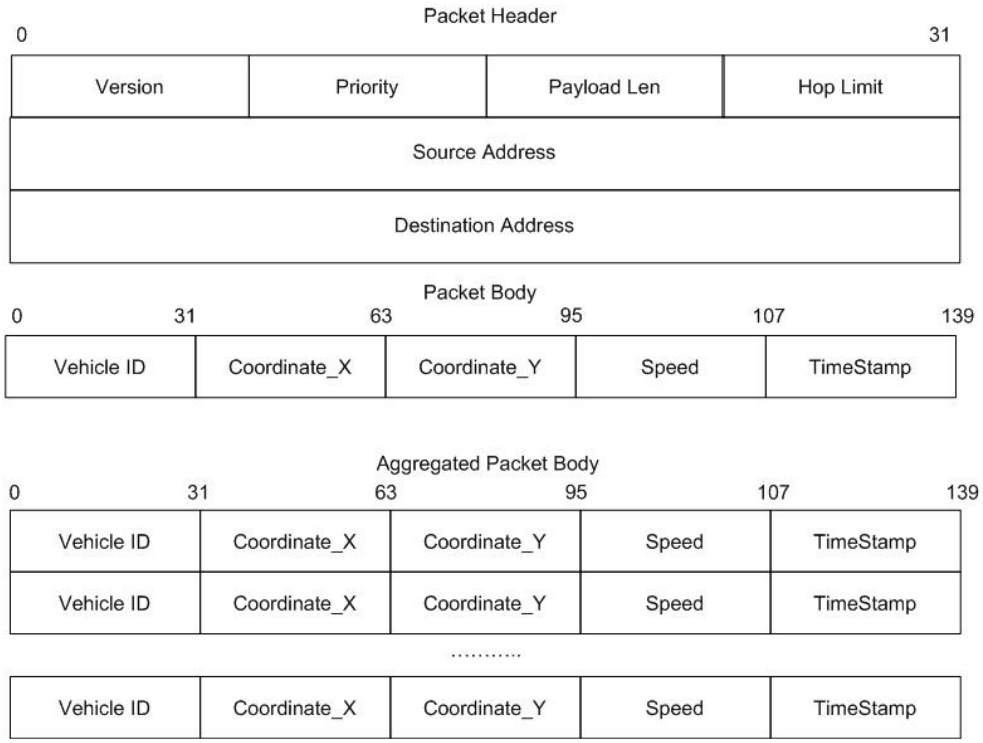

Fig. 5. Packet Structure

The message challenging method is used to verify records which are disputed. Whenever vehicles in a cell do not agree with the broadcasted message, they express their concern by broadcasting their records. A vehicle getting different information for the same vehicle can verify the disputed record through this method. Since we assumed that traffic moves in both directions, a node can send the verification request to the cell router in the opposite direction. We show an example in Figure 6. Traffic is moving in two directions: the bottom road is eastbound whereas the top road is westbound. Suppose car A transmits a message which is propagated backwards and eventually received by car I. If car I would like to verify the position received from car A, it can send the verification request using cars moving in the opposite direction. Cell routers forward the message to neighboring cells until it reaches the destination cell. The destination can be more than one cell depending on the position of the node when the record was sent, its speed and the cell where it was present. Once the car with the same ID is identified, its position can be verified and a response can be generated. If there is no such node existing in the potential destination cells, or the position information was modified, then the record is

considered to be spurious and is dropped. The sender of that record comes under the scrutiny by other honest nodes. In Figure 6, the rightmost cell is the destination cell. This request is propagated from car I to car e through cars b and d. Since car A falls within the transmission range of car e, its position can be verified using the radar of car e. This verified information is then sent back to car I. This verification method works because of the low processing and propagation time. If the record that is being disputed is from a vehicle very far away, the vehicles drop the information instead of challenging it. Unless many records are being changed, or some other vehicle tells that vehicle in dispute is indeed close by, no request is sent. Global security is based on the

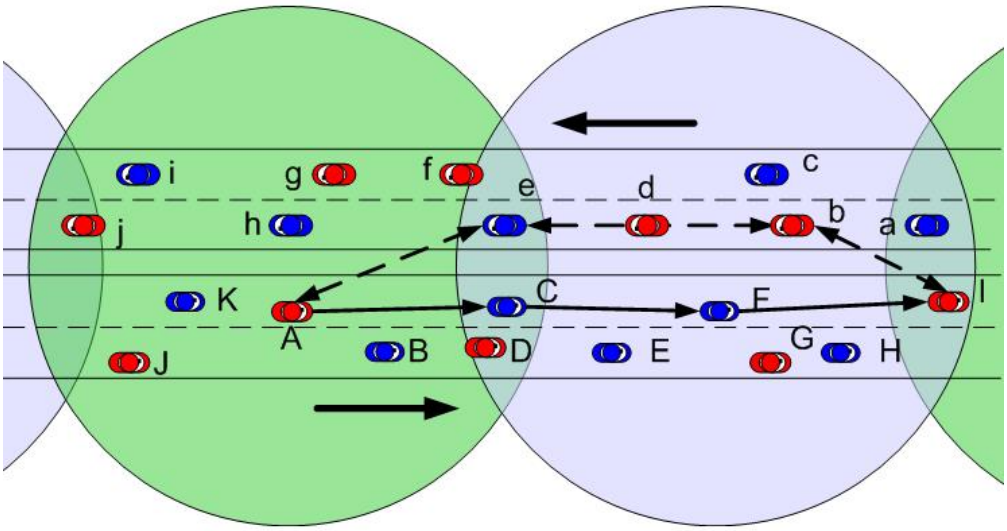

Fig. 6. Message routing among the cells.

fact that the vehicles in the same cell would see and hear almost the same traffic and road situation, so any modification done by malicious nodes can be detected by other honest vehicles. These honest vehicles then broadcast the correct record and isolate the malicious vehicle.

6 Preventing Sybil Attacks

6.1 Threat Model

The threat model assumed in this work includes three types of attack: the position attack, the Sybil attack, and the combination of the position attack and the Sybil attack. When radar's line of sight is blocked, position attacks can happen. An attacker can launch a position attack by revising position packets, replaying bogus position packets and dropping urgent position packets. For example, an attacker may send a fake message created by changing a correct

position into false position or inserting a bogus position when the neighbors' radar does not work under certain scenarios. Since the local neighbors and even the opposite vehicles can not "see" the positions which are pointed by the fake packets, these illusions of vehicles' position can give the attackers an alternate route to their advantage when the traffic is jammed. Even worse, the attacker can drop all the deceleration alerts from reaching other vehicles in a traffic accident by creating illusions of router vehicles.

The Sybil attack [7] is a well-known harmful attack in VANETs whereby a vehicle claims to be several vehicles either at the same time or in succession. The Sybil attack is harmful to network topologies, connections, network bandwidth consumption, and there are some threats even related to human life. For example, a Sybil attacker may give the illusion of 100 vehicles on a one kilometer highway. These illusions change the topology of the network dramatically. The real vehicles will try to communicate with these illusion vehicles but will never receive any acknowledgements from them. The real vehicles retransmit the packets, which consumes network bandwidth. In some urgent situations, for example rescuing people at an accident site on the highway, the illusions will slow down the traffic, hampering the rescue vehicles from reaching the accident site.

The combination of the position attack and Sybil attack is a mixture of these two attacks. The harm of this attack is worse than a single position attack or Sybil attack. Moreover, it is harder to prevent.

What makes an attack possible is the following: the local vehicle has no direct physical knowledge of remote vehicle, therefore the local vehicles perceive the remote vehicles only through abstract information. However, this abstract information can be forged or modified by attackers. The local vehicles perceive the illusion of a vehicle if they receive forged abstract vehicle entities.

6.2 Overview

We propose a solution to prevent some forms of Sybil attacks. Our idea is that if radar works such that it can detect the physical existence of a vehicle, we can use this physical information to remedy the purely abstract information about a vehicle. We compute similarity among three kinds of data: radar detections, oncoming traffic reports and neighbors reports. To average these similarities, each similarity has a weight. When radar works, radar detections are more trustworthy, therefore radar detections have a larger weight; when radar does not work, neighbors' reports have a larger weight. The average position and velocity will be computed if the similarity is close. A history of the road map is maintained by storing these average positions and velocities over a period of

time. When a query according to position needs to be made, vehicles rebuild the target vehicle's map history virtually and make their decision based on this map.

6.3 Sources of Data

We use three types of data resources: radar detection data, oncoming traffic radar detection data, and neighbors' reports.

1) Radar Detection Data: Using radar, we can get the relative velocity and distance from the observer vehicle. We can get the radar detection data by the following formula:

$$\overrightarrow{\nu_r} = \overrightarrow{\nu_l} + \overrightarrow{\nu_r e} \tag{9}$$

where $\overrightarrow{\nu_r}$ is the detected velocity by radar, $\overrightarrow{\nu_l}$ is the local velocity, $\overrightarrow{\nu_r e}$ is the relative velocity computed from radar. Since the velocity is a vector, we define approaching the local vehicle as positive and moving farther from the local vehicle as negative. Similarly, we have the following position formula:

$$\begin{cases} x_r = x_l + x_{re} \\ y_r = y_l + y_{re} \end{cases} \tag{10}$$

where x_r and y_r are the x and y coordinates collected by radar, x_l and y_l are the local xy axis values, x_{re} and y_{re} are the relative xy axis values obtained by the Doppler Effect.

2) Oncoming Traffic Data: We have reactively collected data from oncoming traffic, as described in Sections 3 and 5. We mark the velocity from oncoming traffic as $\overrightarrow{\nu_t}$, and the position as (x_t, y_t).

3) Neighbors Data: We collect data proactively and reactively from neighbors in a cell, as described in Section 3. We mark the velocity from oncoming traffic as $\overrightarrow{\nu_n}$, and the position as (x_n, y_n).

6.4 Pattern Recognition Model

6.4.1 Similarity Computation

Based on classic pattern recognition, we use cosine similarity [13] which computes the similarity of two vectors. The physical meaning of the similarity is

the cosine value of the angle produced by the two vectors. In Figure 7, we can see that if the angle is small, the cosine value is close to one. For example, if the angle is 5 degrees, then cosine(5) = 0.996, so we can declare that the two vectors are almost the same.

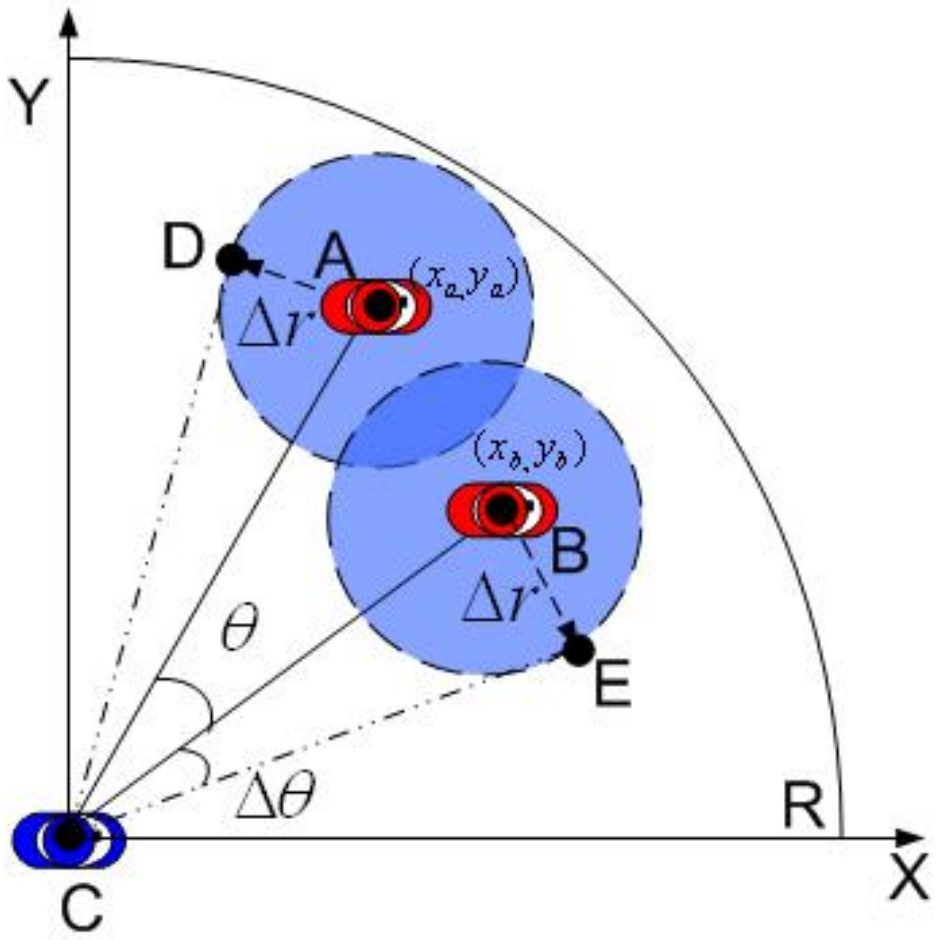

Fig. 7. GPS coordinates positions. Vehicle C receives two GPS position copies A and B of a single vehicle. The GPS tolerance results in a set of real GPS positions, shown as dark shadow. The similarity of the two copies is the cosine value of angle (DCE).

The formula to compute the cosine similarity is the following:

$$Sim(\overrightarrow{A}, \overrightarrow{B}) = \cos\theta = \frac{\overrightarrow{A} \bullet \overrightarrow{B}}{|\overrightarrow{A}||\overrightarrow{B}|} \tag{11}$$

When we apply the xy-coordinates to (11), we get:

$$\frac{\overrightarrow{A} \bullet \overrightarrow{B}}{|\overrightarrow{A}||\overrightarrow{B}|} = \frac{x_1 * x_2 + y_1 * y_2}{\sqrt{x_1^2 + y_1^2} * \sqrt{x_2^2 + y_2^2}} \tag{12}$$

Another solution is using a three dimension vectors $(\overrightarrow{\nu}, x, y)$, if we include velocity in a vector as well. The formula is:

$$\frac{\overrightarrow{A} \bullet \overrightarrow{B}}{|\overrightarrow{A}||\overrightarrow{B}|} = \frac{x_1 * x_2 + y_1 * y_2 + \nu_1 * \nu_2}{\sqrt{x_1^2 + y_1^2 + \nu_1^2} * \sqrt{x_2^2 + y_2^2 + \nu_2^2}} \tag{13}$$

In (13) we see that the velocity affects the similarity. But we need further study to find which formula is more appropriate.

We can get three types of data as described in Section 6.3. Since they are from different sources of data, the three types of data are not exactly same. They will have some precision tolerance, as shown in Figure 8 where C_d, C_e, C_f are close positions to the vehicle's real position C, but C_b is not a close position. We can not directly use any one of these data, however we can give them weights w_i depending on how much they will contribute to the confidence of the similarity. When radar works, we weight the radar's detections more than neighbors' reports because radar is more reliable than neighbors' reports. When radar does not work in some situation, for example the line of sight is blocked by a large truck, neighbors' report will be given more weight. Since the majority of vehicles behave properly, in most cases, neighbors' reports are trustworthy. If, at the same time, neighbors are allied to lie and radar does not work, using the map history, to be discussed in Section 6.5, may help in some cases. We will study the cases which map history does not help in future work.

From (11), we can get

$$Sim_r = Sim(\overrightarrow{A}, \overrightarrow{B})_r * w_r \tag{14}$$

$$Sim_t = Sim(\overrightarrow{A}, \overrightarrow{B})_t * w_t \tag{15}$$

$$Sim_n = Sim(\overrightarrow{A}, \overrightarrow{B})_n * w_n \tag{16}$$

$$Sim(\overrightarrow{A}, \overrightarrow{B}) = Sim_r + Sim_t + Sim_n \tag{17}$$

We mark $Sim(\overrightarrow{A}, \overrightarrow{B})_r$ as the similarity computed from radar detected data, w_r as the weight value for the similarity from radar detected data, $Sim(\overrightarrow{A}, \overrightarrow{B})_t$

as the similarity from oncoming traffic data, w_t as the weight value for the similarity from oncoming traffic data, $Sim(\overrightarrow{A}, \overrightarrow{B})_n$ as the similarity from neighbors' data, and w_n as the weight value from the similarity from neighbors' data. We trust radar detection data more than oncoming traffic data which is trusted more than the neighbors' data, because it is more difficult to attack in the sequence radar, oncoming traffic, and neighbors. Thus we have $1 > w_r > w_t > w_n > 0$.

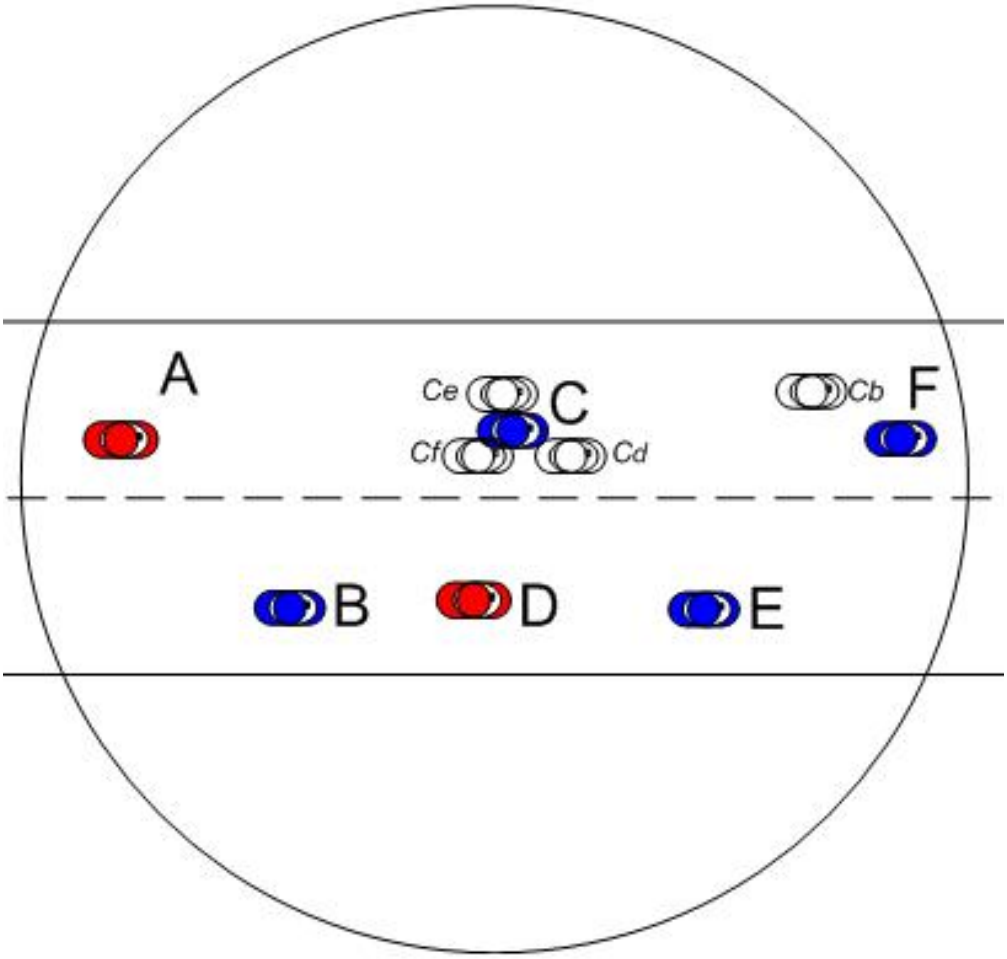

Fig. 8. Similarity computing with precision tolerance. C_e is the position detected by radar; C_f is the position detected by oncoming traffic radar; C_d is the position reported by neighbors; C_b is the false position reported by neighbors.

Once we compute the similarity from all the resources, we can apply a threshold to discriminate the "real" or close to real data from the fake data by this formula:

$$Sim(\overrightarrow{A}, \overrightarrow{B}) \leqslant Sim_{threshold} \tag{18}$$

The determination of the threshold $Sim_{threshold}$ is part of our future work. Once all data from different sources are determined to be highly similar each other, we compute the average of these data as in (19).

$$\begin{cases} \overrightarrow{\nu_{avg}} = \frac{1}{N} \sum_{i=1}^{N} \overrightarrow{\nu_i} \\ x_{avg} = \frac{1}{N} \sum_{i=1}^{N} x_i \\ y_{avg} = \frac{1}{N} \sum_{i=1}^{N} y_i \end{cases} \tag{19}$$

6.5 Map History

The purpose of map history is to classify new data as "real data" or "fake data" based on the history of a vehicle. The basic idea is that any vehicle without historical consistency is highly suspect.

6.5.1 Map History Overview

Map history is built on the local host vehicle about a remote vehicle's history. If a vehicle has just joined the traffic, we do not rely on this vehicle to transmit, retransmit, or perform any other operations. We will give this vehicle more time to build a history on our local host vehicle. If a vehicle has history, but the average position data is inconsistent based on the map history, we reject this data and put this vehicle in the distrust table. We give two examples in Figure 9. In the first example, we compute a position by (18) and (19). If this position is at A, we can reject the vehicle because the average position is outside the road. In another example, the computed average position is at B. We reject the vehicle because it is not in the region $t_0 - t_1$ where it is supposed to be. t_1 is the last received position, and t_0 is the predicted position.

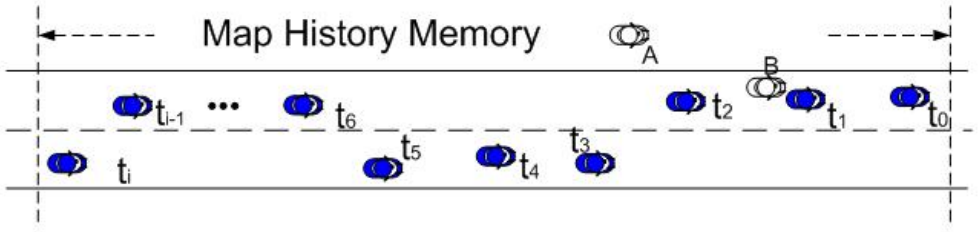

Fig. 9. Map history Example. A is an impossible position because it is outside the road; B is an incorrect position because it is supposed to be between t_0 and t_1, where t_0 is the predicted position and t_1 is the last received position.

Map history is stored in the local host vehicle's memory or storage device. It is composed of N segments. Each segment describes the data at a certain level.

We term each stage as a *Stack* as shown in Figure 10. For example, Stack One includes records which are collected each second. All the records are processed by the method described in Section 6.4.1 before it is recorded into the Stacks. In one second, a vehicle moves at most 33 meters if we assume the maximum speed is 75 miles per hour. Therefore, the interval between t_0 and t_1 is at most 33 meters. Because of transient velocity, some intervals are shorter than 33 meters. In Figure 10, an overview of the Stack is shown. Stack Two includes records which are collected every 10 seconds. Therefore we get more sparsely recorded positions in Stack Two as shown in Figure 10.

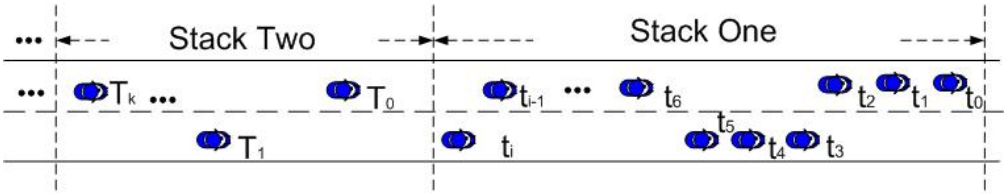

Fig. 10. Map history Overview. Stack One has more frequently collected data: one record per second and a shorter time duration, which is dependent on how detailed the position is recorded in Stack One. Stack Two has a larger interval of time between two records which builds a sparser position history.

6.5.2 Filtering Out Impossible Positions

With the map history Stacks and the processed data from Section 6.4.1 in hand, we can separate the "real data" from the "fake data".

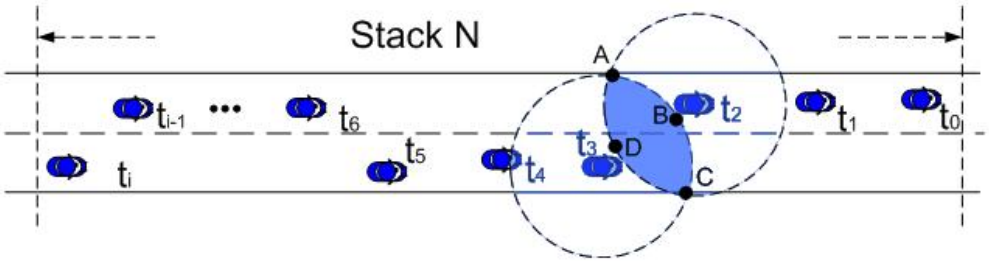

Fig. 11. Screen Vehicles Using Map History. If a vehicle does not fall into the shaded region, ABCD, it is a suspected vehicle. The circles are based on the vehicle velocity and interval of records.

For each record, we have time, velocity, and position attributes. Subsequent

vehicle positions can be described by the following formula:

$$\begin{cases} (x - x_i)^2 + (y - y_i)^2 \leq \gamma^2 \\ \gamma = t * \nu_i \\ 0 \leq \nu \leq v_{max} \\ \{x, y\} \in Road \end{cases} \tag{20}$$

For any Stack N, we have a predicted record t_0 which is given based on the maximum velocity:

$$\begin{cases} (x - x_i)^2 + (y - y_i)^2 \leq (t * v_{max})^2 \\ \{x, y\} \in Road \end{cases} \tag{21}$$

In (20) and (21), x_i, y_i are the coordinates of record i in any Stack N; x, y are the coordinates of new data or record; t is the length of sample interval of Stack N; v_{max} is the maximum velocity on the road by consulting the digital map. We can consult the digital map with the (x, y) coordinates to check if (x, y) is on the road. So far, we can screen all the data by solving (20) and (21). The physical meaning is that if the new data inside the shaded region in Figure 11, we can declare the data acceptable. Otherwise it is not acceptable, and the vehicle will be isolated.

7 Isolating Malicious Vehicles

In this section, we use position security (local security and global security) and map history security to isolate vehicles with abnormal behavior in some position attacks, some Sybil attacks and some combinations of position attacks and Sybil attacks. To isolate compromised vehicles, all vehicles in a cell maintain three tables in memory: trust table, question table and distrust table, shown in Figure 12. Each table consists of records indexed by vehicle ID. An example record is shown in Figure 13. Each record consists of a set of Stacks. Each Stack contains a map history at a certain level as addressed in Section 6.5. All the records which are beyond the timeout limit will be deleted. We are virtually guaranteed that the trust table will be large and as a consequence, the task of determining a cell leader and routers will be always successful.

Initially, a new vehicle, as an observer, enters the road. It first enters a cell and broadcasts a "HELLO" message. The cell leader sends back its ID, the

cell routers' IDs and all the other members' IDs. When the leader sends information to the observer, members in the cell will hear it. If there is bogus information in the message, a member may dispute the leader as we mentioned in Section 5. The observer places records of the leader and routers into its trust table and places other members' records into its question table. From then on, the observer starts to build its own map history. Figure 12 shows the tables and state transitions. If an observed vehicle behaves normally for a certain period of time, the observer will place it into the trust table. If an observed vehicle behaves abnormally, the observer will place it into the distrust table. Otherwise the vehicle will stay in the current table. Here normal behavior means the reported position is detected by radar. If not receiving reports from an observed vehicle or not detecting the observed vehicle for a while by observer's radar or oncoming traffic radar, the observer will move the observed vehicle from the trust table to the question table.

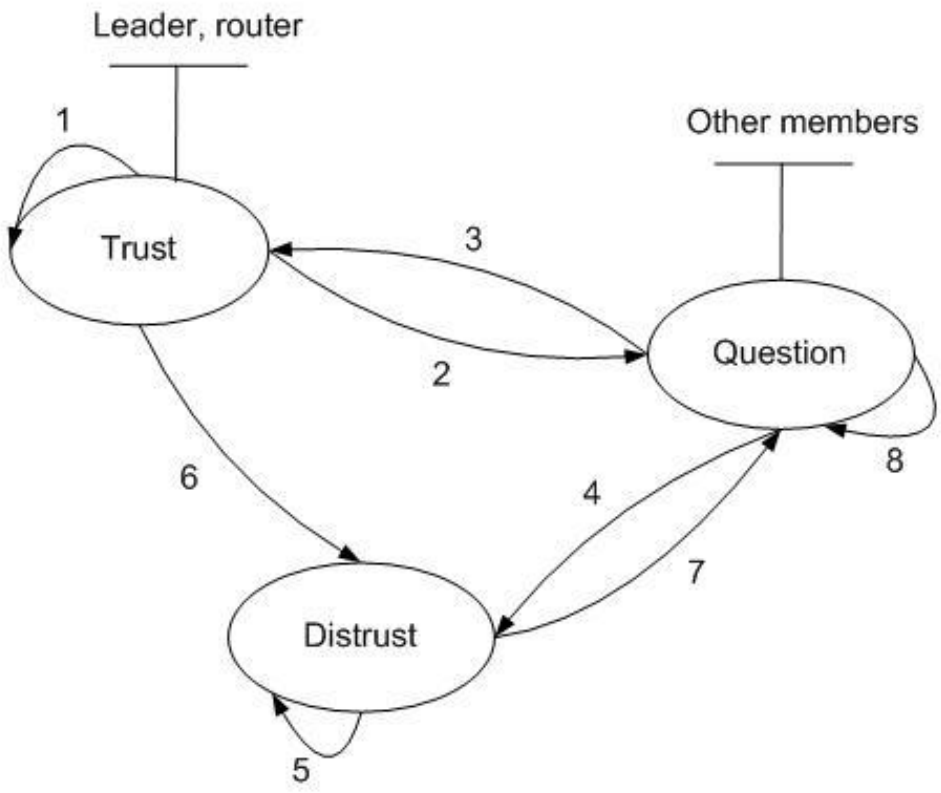

Fig. 12. State transitions. State transitions 1, 3, 5, 7: if confirmed by radar or opposite vehicles' radar, State Transitions 2, 4, 6, 8: if not confirmed by radar or opposite vehicles' radar

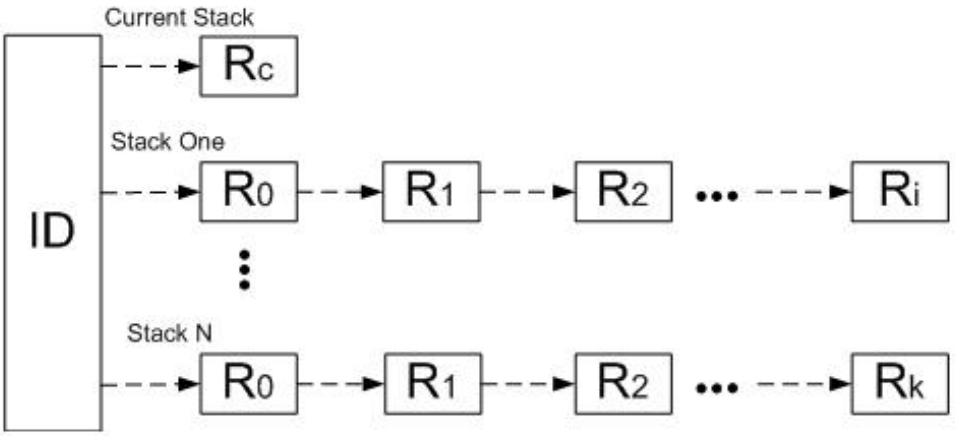

Fig. 13. Map history Structure. The first stack has more detailed records and shorter intervals; the second stack has less detailed records and longer intervals. The higher the stack number is, the less detail it has.

8 Possible Attacks

As an example, we show a planned attack in Figure 14. The attack is composed of one centered vehicle (A) and four large trucks (P, O, X, and Y) which are driving parallel to each other, blocking the lanes. These five vehicles maintain a relative distance apart, which is about the size of a cell. Since Vehicle A is at the center of these large trucks, it will most likely be a cell leader. In this scene, the radar does not work because the line of sight is blocked by the four trucks. Vehicle A, as a cell leader, can receive other cell leaders' position report packets. If vehicle A randomly picks up one packet and revises a random record in the packet, it is very hard to detect the modification. We will study this attack in future work. However, this kind of modification can do little harm to the system. First, the modification can only affect a very few vehicles which can be ignored from the macro level of the network. Second, radar does not work and it cannot harm the system. In the scenario without radar, we can apply the method proposed by Leinmüller [10]. They detect radio signals and use a hard threshold to determine the compromised vehicles. If the compromised vehicles modify more packets to increase the harm to the system, the chance to be detected by our challenge method increases, too.

Another possible attack is shown in Figure 15. Attacker vehicle A with ID ID_a uses the location L_c of vehicle C. A claims to vehicle B that its identity is ID_a, and that its location is L_c. Vehicle B verifies that there is a vehicle at L_c, then concludes that it is ID_a. However this attack can be solved by our map history described in Section 6.5. If B does not have a map history of ID_a, A cannot harm the system because it is in the question table. We do not use this vehicle to do any operations, such as transmit packets, retransmit

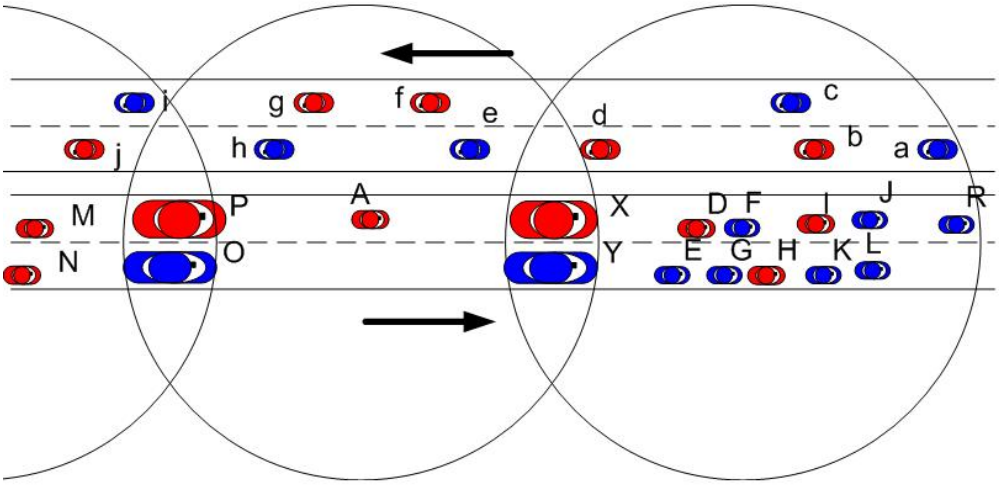

Fig. 14. A planned attack scenario: four big trucks P, O, X, Y are driving parallelly, vehicle A is always driving near the center of a cell. They keep relative distance.

packets, and report data, etc. If B has a map history of ID_a, we can easily to identify the illusion position of A. A segment of the Stack in the map history is shown in Figure 16. Which segment is selected depends on L_c. We assume a segment of Stack N is selected. All the vehicles in Figure 16 represent the previous position of A. We calculate possible next positions of A shown as a shaded region in Figure 16. Since L_c, shown as T_0, does not fall into the shaded region, we conclude that A lies. We put A into distrust list to isolate A. Even if L_c happens to fall into shaded region, it can not always fall into the shaded region because the relative position of vehicles is transient. And A can not compute the same map history with B because the map history depends on three sources of data which are collected at a specified situation and a specified time for a specified vehicle.

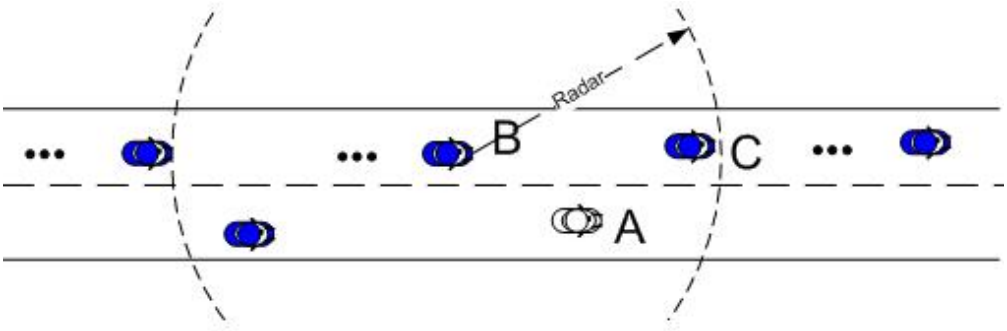

Fig. 15. A possible Sybil Attack. A obtains C's position L_c. A claims to victim B that its position is L_c, and that its ID is ID_a. B detects a vehicle is at L_c then concludes that it is the position of A.

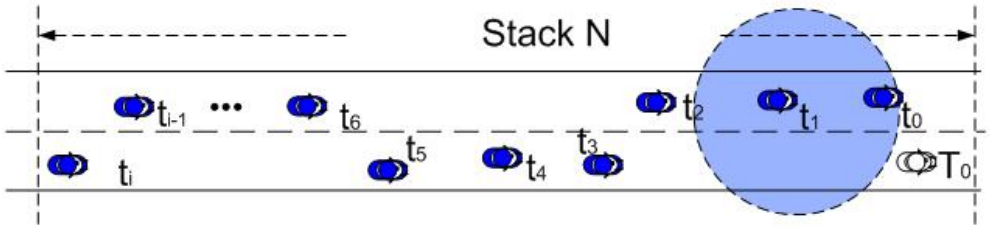

Fig. 16. A segment of Stack in map history.

9 Simulation Results

In our experiments, we simulated a two-direction 3 km highway with two lanes in each direction. The cell radius was 100 meters, traffic arrival rate was 1600 vehicles/hour, mean velocity was 33.3 m/s, and transmission radius was 100 meters.Each simulation has some number of compromised, or malicious, vehicles and a single observer vehicle. When the observer enters the simulated highway, it initiates a request to find all of the compromised vehicles. The simulation terminates when the observer reaches the end of the 3 km highway.

We wanted to investigate the amount of time required to detect a certain number of compromised vehicles. In our first set of experiments, we inserted 16 compromised vehicles and varied the total number of vehicles on the highway. Each experiment was run 10 times, and we measured the average amount of time needed to detect the 16 compromised vehicles. As a comparison, we show in Figure 17, the average amount of time using our cell-based routing as opposed to message flooding, where each node re-broadcasts the received message. As expected, cell-based routing is more efficient than flooding. The time needed to detect the compromised vehicles increases when the number of vehicles decreases. The reason is that in low density traffic vehicles must physically drive the packet to the next cell if there are no middle nodes available as routing nodes.

To determine how transmission range affects the time needed to detect malicious vehicles, we ran a set of experiments with a 100 m transmission range and a set with a 500 m transmission range. The vehicle density was about 30 vehicles per kilometer per lane on the highway. The compromised vehicles were 5% of the total vehicles and were randomly deployed along the highway. In each set of simulations, we varied the length of the highway to investigate the effect of transmission range. Since the time depends on the number of intermediate hops, the increased range of transmission would certainly decrease the time. However, the increased transmission range would increase the probability of packet collisions. Figure 18 shows the time taken to detect the malicious vehicles with respect to their distance from the vehicle generating the verification request. As expected, with a 500 m range, the time is much

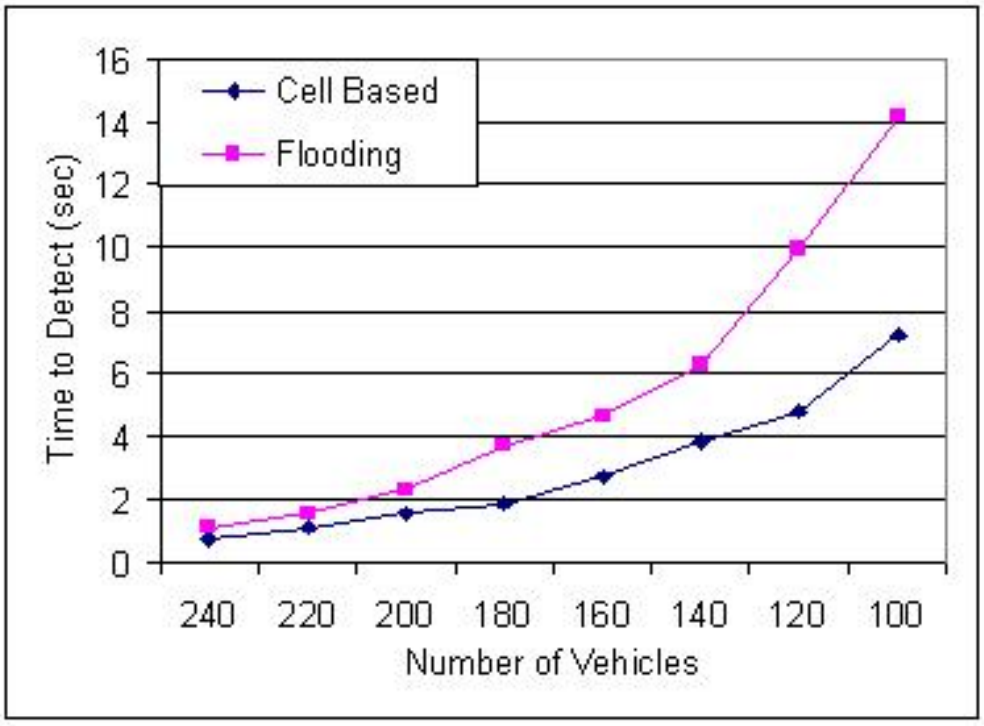

Fig. 17. Time needed to detect the 16 malicious vehicles as the total number of vehicles varies.

less, but the re-broadcasting of packets would need to be handled carefully to avoid an increase in collisions.

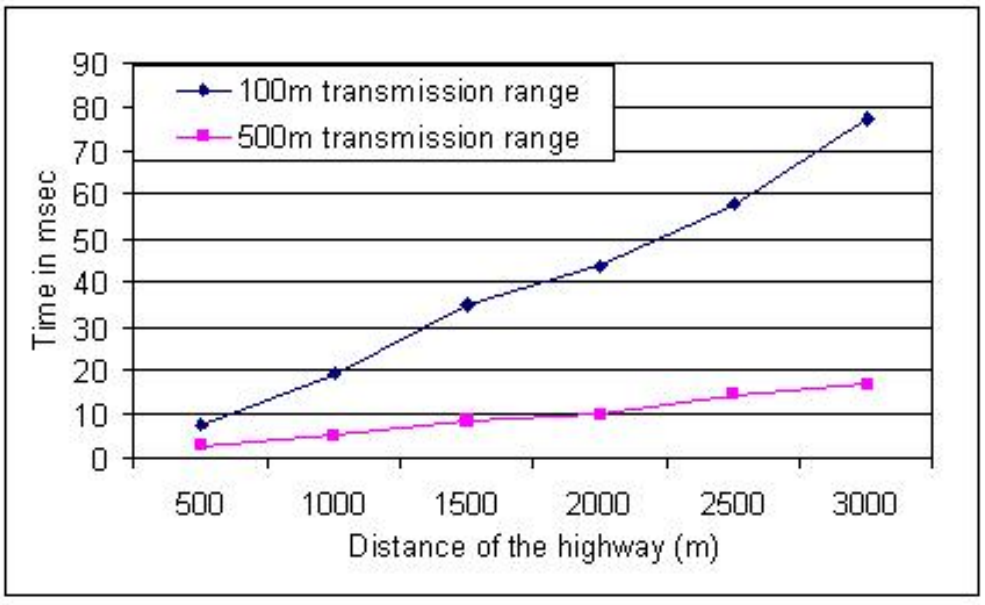

Fig. 18. Average time to detect malicious vehicles.

Finally, we wanted to investigate how many compromised vehicles could remain undetected in our system. We randomly distributed compromised vehi-

cles along the highway. The vehicle density was about 30 vehicles per kilometer on the highway. The transmission radius was 100 meter. We varied the percentage of all vehicles that were compromised from 0% to 30% because our assumption is that the majority of vehicles are honest. Compared with previous compromised vehicle rate (5%), we enlarged this number to 30 % to investigate a larger scope. We measured the number of compromised vehicles detected in 30 seconds and show the results in Figure 19. As expected, the number of undetected compromised vehicles decreases when the rate of compromised vehicles decreases.

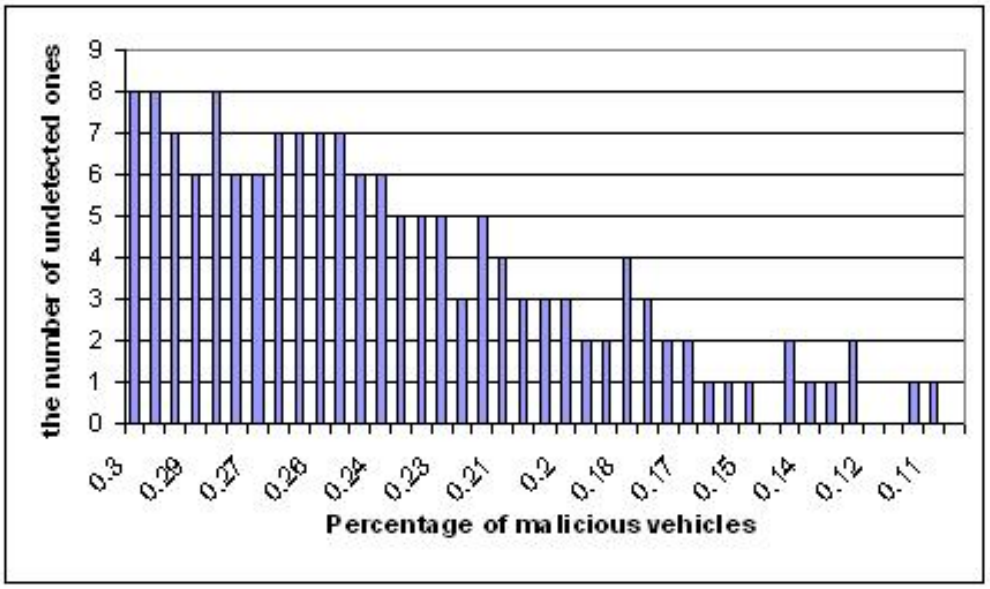

Fig. 19. The number of undetected compromised vehicles as the percentage of compromised vehicles increases.

10 Concluding Remarks and Future Work

Detecting false position information and reducing the chances of attack are the keys to success in securing VANETs. Our paper focuses on this prime area. The radar acts as the "eye" of the system and verifies the information received from the vehicles within its transmission range. The capability to verify records is also used for achieving global security. Our approach is efficient in determining compromised vehicles and reduces the burden on channel availability.

We are working on increasing the precision of our system to detect all the compromised vehicles. There is some inherent imprecision due to the technology and hardware used. GPS has a 2-5 meter tolerance whereas radar has 5 meter of imprecision. We are working on simulating the Sybil attack and some combination of Sybil attacks and position attacks.

There are some special cases that will be further studied: 1) When the line of sight of radar is blocked by obstacles, attackers may launch some Sybil attacks. 2) When a honest vehicle with a long and valid map history is leaving, attackers may launch some Sybil attacks. 3) Cell routers and leaders may be a target of selective attack. 4) The size of map history needs to be studied.

Acknowledgements

We thank Gyanesh Choudhary for helpful suggestions in this work.

References

[1] Car 2 Car Communication Consortium, http://www.car-to-car.org/.

[2] US Department of Transportation, National Highway Traffic Safety Administration, Vehicle safety communications consortium, http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/CAMP3/pages/VSCC.htm.

[3] A. Takahashi, N. Asanuma, Introduction of Honda ASV-2 (Advanced Safety Vehicle-Phase 2), in: Proceedings of the IEEE Intelligent Vehicles Symposium, Detroit, USA, 2000, pp. 694–701.

[4] J.-S. Park, U. Lee, S. Y. Oh, M. Gerla, D. S. Lun, Emergency related video streaming in VANET using network coding, in: Proceedings of the ACM Workshop on Vehicular Ad Hoc Networks (VANET), Los Angeles, CA, 2006, pp. 102–103.

[5] B. Parno, A. Perrig, Challenges in securing vehicular networks, in: Proceedings of ACM HotNets, 2005.

[6] R. Möbus, M. Baotic, M. Morari, Multi-Object Adaptive Cruise Control, Hybrid Systems: Computation and Control 2623 (2003) 359–374.

[7] J. Douceur, The sybil attack, Lecture Notes in Computer Science: Revised Papers from the First International Workshop on Peer-to-Peer Systems 2429 (2002) 251–260.

[8] P. Golle, D. Greene, J. Staddon, Detecting and correcting malicious data in VANETs, in: Proceedings of the ACM Workshop on Vehicular Ad Hoc Networks (VANET), Philadelphia, PA, 2004, pp. 29–37.

[9] T. Leinmüller, E. Schoch, Greedy routing in highway scenarios: The impact of position faking nodes, in: Proceedings of the Workshop on Intelligent Transportation (WIT), 2006.

[10] T. Leinmüller, E. Schoch, F. Kargl, C. Maihöfer, Improved security in geographic ad hoc routing through autonomous position verification, in: Proceedings of the ACM Workshop on Vehicular Ad Hoc Networks (VANET), Los Angeles, CA, 2006, pp. 57–66.

[11] M. Raya, J.-P. Hubaux, The security of vehicular ad hoc networks, in: Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks, Alexandria, VA, 2005, pp. 11–21.

[12] T. Leinmüller and E. Schoch and F. Kargl and C. Maihöfer, Influence of falsified position data on geographic ad-hoc routing, in: Proceedings of the European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS), 2005.

[13] S.-T. Yuan, J. Sun, Ontology-based structured cosine similarity in speech document summarization, in: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence, Washington, DC, USA, 2004, pp. 508–513.

[14] J.-P. Hubaux, S. Capkun, J. Luo, The security and privacy of smart vehicles, IEEE Security and Privacy Magazine 2 (3) (2004) 49–55.

[15] M. Raya, A. Aziz, J.-P. Hubaux, Efficient Secure Aggregation in VANETs, in: Proceedings of the ACM Workshop on Vehicular Ad Hoc Networks (VANET), Los Angeles, CA, 2006, pp. 67–75.

[16] M. Raya, P. Papadimitratos, J.-P. Hubaux, Securing vehicular communications, IEEE Wireless Communications Magazine (2006) 8–15.

[17] S. Capkun, J.-P. Hubaux, Secure positioning of wireless devices with application to sensor networks, in: Proceedings of IEEE INFOCOM, Vol. 3, 2005, pp. 1917–1928.

[18] M. G. Kuhn, An asymmetric security mechanism for navigation signals, in: Proceedings of the Workshop on Information Hiding, Toronto, Canada, 2004, pp. 239–252.

[19] F. Armknecht, A. Festag, D. Westhoff, K. Zeng, Cross-layer privacy enhancement and non-repudiation in vehicular communication, in: Proceedings of the Workshop on Mobile Ad-Hoc Networks (WMAN), Bern, Switzerland, 2007.

[20] J. Y. Choi, P. Golle, M. Jakobsson, Tamper-evident digital signatures: Protecting certification authorities against malware, in: Proceedings of the IEEE International Symposium on Dependable, Autonomic and Secure Computing (DASC), 2006, pp. 37–44.

[21] K. Plößl, T. Nowey, C. Mletzko, Towards a security architecture for vehicular ad hoc networks, in: Proceedings of the International Conference on Availability, Reliability and Security (ARES), Washington, DC, USA, 2006, pp. 374–381.

[22] F. Dai, J. Wu, Proactive route maintenance in wireless ad hoc networks, in: Proceedings of the IEEE International Conference on Communications (ICC), Vol. 2, 2005, pp. 1236–1240.

[23] N. Sastry, U. Shankar, D. Wagner, Secure verification of location claims, in: Proceedings of the ACM Workshop on Wireless Security (WiSe), San Diego, CA, 2003, pp. 1–10.

[24] T. Suen, A. Yasinsac, Ad hoc network security: Peer identification and authentication using signal properties, in: Systems, Man and Cybernetics (SMC) Information Assurance Workshop, 2005, pp. 432–433.

[25] B. Xiao, B. Yu, C. Gao, Detection and localization of Sybil nodes in VANETs, in: Proceedings of the Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks, Los Angeles, CA, USA, 2006, pp. 1–8.

[26] J. Newsome, E. Shi, D. Song, A. Perrig, The Sybil attack in sensor networks: analysis & defenses, in: Proceedings of International Symposium on Information Processing in Sensor Networks (IPSN), Berkeley, CA, 2004, pp. 259–268.

[27] C. Piro, C. Shields, B. N. Levine, Detecting the Sybil attack in mobile ad hoc network, in: Proceedings of the International Conference on Security and Privacy in Communication Networks, 2006, pp. 1–11.

[28] Sensor Technologies and Systems, Forward looking vehicle radar system (FLVRS), http://www.sensor-tech.com/sub%20pages/products/AUTOMOTIVE/flvrs.html (2006).

[29] Toyota, Pre-crash safety, http://www.toyota.co.jp/en/about_toyota/in_the_world/pdf2007/safety.pdf (2007).

[30] US Department of Transportation, National Highway Traffic Safety Administration, Event data recorders, Federal Motor Vehicle Safety Standards 215 Part 49, Code of Federal Regulations Part 563 (2006).

[31] US Department of Transportation, National Highway Traffic Safety Administration, Event Data Recorders Q&As, www.nhtsa.dot.gov/staticfiles/DOT/NHTSA/Rulemaking/Rules/Associated%20Files/EDR_QAs_11Aug2006.pdf (Aug. 2006).

[32] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, J.-P. Hubaux, Eviction of misbehaving and faulty nodes in vehicular networks, IEEE Journal on Selected Areas in Communications 25 (8) (2007) 1557–1568.

[33] Z. J. Haas, A new routing protocol for the reconfigurable wireless networks, in: Proceedings of the IEEE Conference on Universal Personal Communications, Vol. 2, 1997, pp. 562–566.

[34] M. Käsemann, H. Füler, H. Hartenstein, , M. Mauve, A reactive location service for mobile ad hoc networks, Tech. Rep. TR-02-014, Department of Computer Science, University of Mannheim (2002).

[35] F. Picconi, N. Ravi, M. Gruteser, L. Iftode, Probabilistic validation of aggregated data in vehicular ad-hoc networks, in: Proceedings of the ACM Workshop on Vehicular Ad Hoc Networks (VANET), Los Angeles, CA, 2006, pp. 76–85.

[36] L. Tian, Y. Zhou, L. Tang, Improving GPS positioning precision by using optical encoders, in: Proceedings of Intelligent Transportation Systems, Dearborn, MI, 2000, pp. 293–298.