A General Testing Framework Based on Veins for Securing VANET Applications

Liang Ming, Gang Zhao, Minhuan Huang†, Xiaohui Kuang†, Jingzhe Zhang, Huayang Cao†, Fei Xu
National Key Laboratory of Science and Technology on Information System Security
Beijing 100101, China
mingliang79@aliyun.com, zg@public.bise.ac.cn, hminwell@gmail.com, xiaohui_kuang@163.com,
zhangjingzhe21@163.com, caohuayangwork@163.com, xufei1023@126.com

***Abstract*—Vehicular Ad Hoc Networks (VANETs) can make vehicles communicate with each other as well as roadside infrastructure units (RSUs) in order to increase transportation efficiency and road safety in smart city. VANET applications in Intelligent Transportation Systems (ITS) include Traffic information systems, Road Transportation Emergency Services, On-The-Road Services, and so on. But more and more cyber threats can compromise these applications and limit our benefits, such as man-in-middle, spoofing and denial of service (DoS). Testing and securing VANET applications themselves is an important concern in ITS design and implementation. In this paper, we analyze security requirement and threat source of VANET applications, and then focus on providing a general testing framework based on Veins simulation platform to secure VANET applications. By use of this testing framework, a proof-of-concept example with malicious messages attack in an expressway tolling application over VANET is also given to demonstrate the testing framework effective and practicable. Experimental results of the example shows the testing framework is suitable for VANET security test, and the attacks will bring about 60.60% more total toll fees, 8.39% longer travel distances, and 43.63% more travel times. Consequently, we also give some countermeasures to improve the security of expressway tolling application. Finally challenges from combining the packet simulators into one framework, and limitations are reported and discussed.**

Keywords-vehicle Ad Hoc networks, VANET Applications, Veins, tolling application, testing framework

I. INTRODUCTION

Vehicular Ad Hoc Networks (VANETs) enable communication among vehicles and between vehicles and roadside infrastructure units (RSUs), and form the backbone of future ITS [1]. Connected Vehicles have some great applications already. Such as Google's public commitment to Self-Driving Car Project, Apple's mysterious strategy on iCar, and Baidu's CarLife. Emergency market for driverless cars, including Tesla and Baidu on environmental-fridendly electric motors. In addition, suppliers, like Delphi Automotive and Mobileye, develop tunkey systems for automakers to build into their vehicles [2]. Based on VANETs, various VANET applications have rapid developments on wireless networking, traffic information systems, road transportation emergency services, automatic incident detection, and so on, which improve the driving experience of users greatly. All these VANET applications make ITS play an important role in smart city.

Securing VANET applications is an emerging area in ITS research with the deployment of smart city. Testing VANET applications is effective way to secure VANET applications and make traffic smooth, since a lot of new threats will be imported into the transportation [3]. In fact, even authored node can have chance to make problem on the system, such as malicious vehicles and RSUs. As a kind of information networking system, VANET applications will encounter much more cyber threats in the future. With the development of information technology, new vulnerability will also bring unintended security problem on VANET applications.

Various VANET applications deployed by connected vehicles expands security vulnerability inherited from wireless communications, particularly in message spoofing and denial-of-service (DoS) attacks. This paper will research the concerns about security and threat of VANET applications and focus on security requirement, threat sources, and specially the general testing framework of VANET application for its higher security by simulation. The rest of the paper is organized as follows: Section II summarizes the ITS reference framework and other related research work about testing and evaluation of VANET applications. Section III analyzes security requirement and the threat source of VANET applications in detail. In Section IV a general testing framework of VANET applications in ITS is presented by a selected proof-of-concept example, and the suggestion about improving security of VANET applications is also given. Section V discusses challenges and limitations of this proposed testing framework. The last Section VI concludes the whole paper.

II. RELATED WORK

In the last twenty years, plenty of studies were conducted by the United States Departments of Transportation (DOT), research institutes, and laboratory in the university to find effective frameworks for designing, testing, evaluating variable VANET applications in ITS. The Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT) is the latest reference architecture model of VANET applications [4]. ARC-IT provides a common framework for planning, defining, and integrating intelligent transportation systems, including all of the Connected Vehicle applications. It covers all of the scope and content from both National ITS Architecture Version 7.1 and the

†Corresponding author

Connected Vehicle Reference ITS Architecture (CVRIA) Version 2.2. The enterprise view describes the relationships between organizations and the roles those organizations play within the cooperative ITS environment. The functional view addresses the analysis of abstract functional elements and their logical interactions. The physical view describes the transportation systems and the information exchanges that support ITS. The communications view describes the protocols necessary to provide interoperability between physical objects in the physical view.

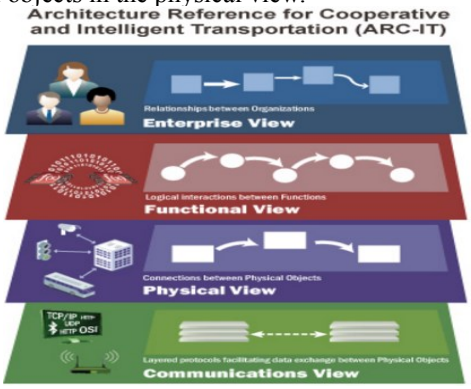


Figure 1. A reference framework of ITS

In order to securing VANET applications, many testing work and research was performed, such as conformance testing [5], reliable v2x communications test [6], performance test [7][8], and so on. In [5], it presents a testing platform for deploying the ITS, but the aim of this platform is to emphasize all properties to check and then to test them on components in order to comply with the standard specification, such as communication protocol conformance testing, interoperability testing, and so on. In [6], it focuses on channel security and researches an testing method on evaluating reliability of bidirectional communication channels with redundant equipment, availability of V2X bidirectional communication channel with operation sessions in the random moments of time. In [7], it presents a method to assess the performance of consensus ensemble prediction of ITS, but it is not a common testing method for other VANET application. In [8], it provides a Application Performance Simulation System to test quantitative evidence of the minimum performance for the detection system of VANET application, but this method is only created for acceptable performance of the detection system, and does not focus on the security testing framework. In a word, all these works have found some ways to test ITS in its performance, but did not give a general testing framework and method for securing VANET applications.

III. Security Requirement and Threat Sources in VANET Applications

Securing VANET applications is important, since VANET applications must be secure before they can reliably be used to improve the efficacy of the surface transportation system. In the view of communication, the security requirement of VANET applications includes physical layer security, data link layer security, network layer security, transport layer security, and application layer security. And as information systems, VANET applications must apply confidentiality, integrity, availability, non-repudiation. What's more, information security, personnel security, operational security, and security management of VANET applications is also important.

Based on the model and security requirement of ITS, according to the multiple-layer stacks of communication protocols, VANET applications may encounter some threats and attacks, which focus on malicious messages and packets:

- Application Layer: The following information concerned threats should be considered: message replay attack, message modification attack, malicious message attack, and other attack ways of man-in-middle, spoofing, sniffing and denial of service. Message replay attack means the adversary resends old messages initially sent by legitimate users in order to increase the network traffic and cause congestion. Message modification attack means the adversary, as a man-in-middle, modify the messages and then sent it out to make problem or mislead users. Malicious message attack includes malicious message attack in Vehicle to Vehicle (V2V) communication and malicious message attack in Road-Side Units (RSU) to Vehicles communication, and the adversary sends bogus information to other nodes in the network; thus misleading users and possibly creating havoc.
- Transport Layer and Network Layer threat: the following communication concerned threats should be considered: forging RSU attack, denial of service (DoS), foundational keys leak, and other communication disturbing attack. Forging RSU attack is to get the information of vehicle node and then disturb the vehicle networking communication. Denial of service in transport and network layer is to make network resource unavailable to its intended users by temporarily or indefinitely disrupting services in nodes connected to the vehicle networking. Foundational keys leak is a kind of threat because of master keys used in V2V communication can be sniffed and abused by attacker.
- Data Link Layer and Physical Layer: the following signal and physical foundation concerned threats should be considered: physical damage, building obstacle, and energy disturbing. Physical damage is to make the RSUs or sensors in car broken or disordered physically. Building obstacle is to interrupt wireless signal communication of vehicle networking by setting up obstacle, such as building, wall, and so on. Energy disturbing is to disturb wireless communication of vehicle networking by energy and power control.

In these threats above, man-in-middle attacks may compromise confidentiality of VANET applications. And sniffing may compromise message integrity of VANET applications. And sniffing and spoofing may compromise message non-repudiation of VANET applications. And DoS,

physical attack, signal interruption and energy disturbing may compromise availability of VANET applications. Besides these technical threats, VANET also confronts many risks on security managements, such as fault of operation, personal security setting fault, keys leak because of loss of vehicle, misusing in rental car, and so on.

IV. A General Testing Framework Based on Veins

A. Testing Framework

In this section we will introduce a general testing framework based on Veins simulation platform for securing VANET applications, as shown in Fig. 2. Veins is an open source platform for running vehicular network simulations [9]. It is based on two well-established simulators: OMNeT++, an event-based network simulator, and SUMO, a road traffic simulator, and offer a comprehensive suite of models for Inter-Vehicle Communication (IVC) simulation. In Fig. 2, the testing module includes message construction, analysis and evaluation, data collection, monitor, and interface components. Message construction component is responsible for customizing testing message, which is sent into the Veins by means of Interface component. Analysis and evaluation component is responsible for making analysis and evaluation based on the response of Veins. Data Collection component is responsible for gathering metric data of testing and evaluation produced by Veins. Monitor component is responsible for outputting the result of analysis, evaluation, and data collection. Channel component is a common communication mechanism used between Veins and testing module in OMNeT++.

By means of analysis and evaluation component, we can do two kinds of testing and evaluation. One is for analyzing the metric value to find out the risk of VANET applications, the other is for decoding response message to check the error result of the testing message, and find out the vulnerability of VANET applications.

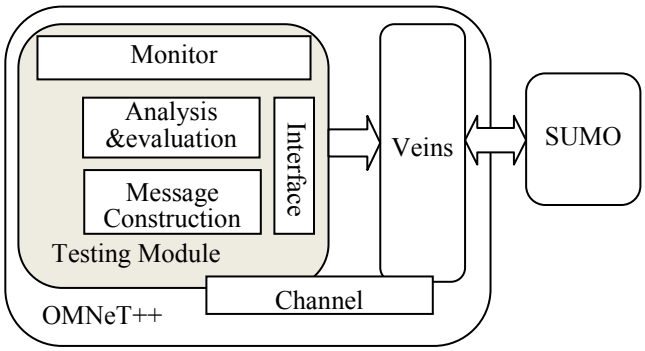


Figure 2. A general testing framework based on Veins

When using this testing framework to make security test, The testing process includes four steps:

- The first step is setting metrics. Based on the testing framework in Section IV A, we should set up some metrics for testing and evaluation according to the key requirement and concerns of the VANET applications. We can also output the value of metrics by editing the source code of Veins.
- The second step is setting up testing environment on Veins. Veins provides a good platform to create VANET applications as needed. We can design various scenarios and build VANET applications by use of existed modules in Veins or writing new function plugins by ourselves. We should also set some probe vectors and scalars in the modules and plugins to output the metric data.
- The third step is customizing attacks. We can simulate attacks by modifying the configure file or definition file of Veins in advance easily. For example, message spoof can be simulated by means of spreading malicious message in source code, and the fault of devices can be simulated by shutting a node down in Veins configure file, and physical disturbance can be simulated by setting up obstacle in Polygon Definition file of SUMO.
- The fourth step is data analysis. We can analyze the values of metrics, and then calculate and evaluate the effort of attacks. This is a kind of application-oriented testing and evaluation, so we can find out to what extent the destroy occurred by the attacks. Based on these results, we can also give some countermeasures for confronting these attacks and reducing its risks.

B. A Testing Case

In the following, we demonstrate the testing framework by a sample of VANET application Expressway Tolling service. According to the threat model, we customize malicious message to attack the Expressway Tolling service by using the testing framework above and then check the metric value; thus find the risk of this VANET application. We research the effects of three different scenarios under malicious message attacks: an expressway tolling scenario with VANET and bad RSU (shortly named VANET with bad RSU scenario), an expressway tolling scenario with VANET and bad vehicles (shortly named VANET with bad vehicles scenario), and an expressway tolling scenario without VANET (shortly named Free-VANET scenario).

In the VANET with bad RSU scenario, malicious RSU will inform all vehicles in its communication range of a pretend incident on the current lane, and the pretend incident warning will be accepted by passing vehicles and make them reroute paths to avoid the incident. In the VANET with bad vehicles scenario, malicious vehicles will inform other vehicles nearby of a pretend incident on the current lane, and the pretend incident warning will be accepted by other vehicles and make them reroute paths to avoid the incident. While in Free-VANET scenario, malicious node can not affect others because of no available V2V communication. To study the effect of malicious message in this VANET application, we create the Expressway Tolling service on Veins and use the testing framework in Section IV A to perform security test. In this example, we employ a realistic map obtained from OpenStreetMap website for simulation [10], and select an about 9 sq.km along Airport Expressway

of Beijing, China, as shown in Fig. 3. In Fig. 3, 100 vehicles travel in certain interval of 1s from the entrance along the Airport Expressway to set up a classic VANET transportation application scenario.

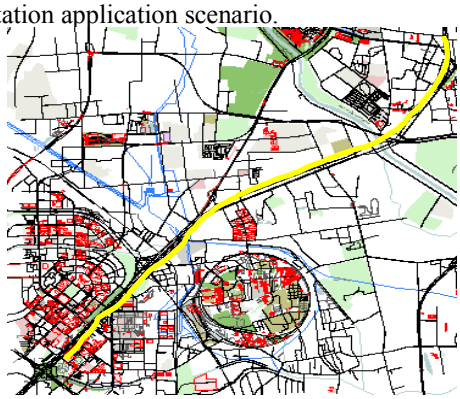


Figure 3. Screenshots of airport expreeway in Beijing, China

TABLE I. SIMULATION PARAMETERS OF ROAD TRAFFIC SETUP

Parameter	Value
acceleration	2.6m/s2
deceleration	-4.5m/s2
driver imperfection	0.4
vehicle's netto-length	2.5m
min gap between two vehicles	4.0m
vehicle's maximum velocity	33.3m/s (120km/h)

TABLE II. INET FRAMEWORK MODULE PARAMETERS

Parameter	Value
mac1609_4.txPower	20mW
mac1609_4.bitrate	18Mbps
phy80211p.sensitivity	-89dBm
phy80211p.thermalNoise	-110dBm
appl.dataInterval	3s
connectionManager.carrierFrequency	5.89GHz
connectionManager.pMax	20mW
connectionManager.sat	-89dBm
connectionManager.alpha	2.0

In our experiments, each vehicle is capable of sending messages and receiving messages. The interval of sending data was defined at 3s. The bit rate was defined at 18 Mbit/s in the MAC layer, and the transmission power at 20 mW. Thus, the communication range is approximately 300m when employing a two-ray ground propagation model [11]. Specially RUS1 is a bad RUS in the VANET with bad RSU scenario, and vehicle 11 and vehicle 91 are two bad vehicles in the VANET with bad vehicles scenario. Each scenario was simulated 2 times to gather enough data for analysis. Yellow trace in Fig. 3 shows the pre-defined route where the vehicles are simulated. TABLE I show the configuration and parameters that were used to execute the simulations.

C. *Metrics and Data*

We will use six metrics to make test and performance evaluation, which includes toll fee, total toll fee, travel times, total travel times, travel distance, and total travel distance.

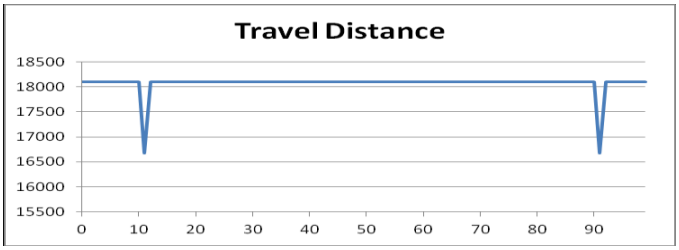
Toll fee of vehicle is the toll price multiply the distance of current vehicle. According to the toll policy of international convention, toll prices are variable based on real-time traffic demand. Average toll prices may range from 35 cents to 65 cents per mile during lighter traffic, and 75 cents to 105 cents during rush hour, aiming to ensure the lanes are moving at 60 km/h or faster.

Total toll fee is the total amount of all tested vehicles. Travel time is the time cost that vehicle travel from the source to the destination in the route. Total travel time is the total amount of travel time of all tested vehicles. Travel distance is the distance that vehicle travel from the source to the destination in the route. Total travel distance is the total amount of travel distance of all tested vehicle. We measure these metrics in all three scenarios respectively in order to analyze the effort of attacks and verify practicability of the testing framework. In Fig. 4, 5, and 6, the horizontal coordinate is index of vehicles

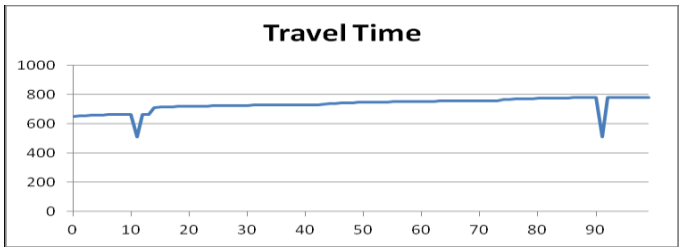
In VANET with bad RSU scenario, the toll fee of each vehicle is shown in Fig. 4(a) and the total toll fee of all vehicles is 669.66 yuans; the travel distance of each vehicle is shown in Fig. 4(b) and the total travel distance of all vehicles is 1807854.748 m; the travel time of each vehicle is shown in Fig. 4(c) and the total travel time of all vehicles is 73218.3s. The total simulation time in this scenario is 988.8s.

In VANET with bad vehicles scenario, the toll fee of each vehicle is shown in Fig. 5(a) and the total toll fee of all vehicles is 454.83 yuans; the travel distance of each vehicle is shown in Fig. 5(b) and the total travel distance of all vehicles is 1710747.60 m; the travel time of each vehicle is shown in Fig. 5(c) and the total travel time of all vehicles is 56140.2s. The total simulation time in this scenario is 919.3s.

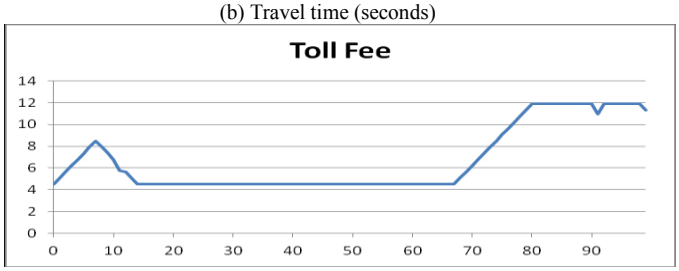
In Free-VANET scenario, the toll fee of each vehicle is shown in Fig. 6(a) and the total toll fee of all vehicles is 416.97 yuans; the travel distance of each vehicle is shown in Fig. 6(b) and the total travel distance of all vehicles is 1667908.88m; the travel time of each vehicle is shown in Fig. 6(c) and the total travel time of all vehicles is 50978.7s. The total simulation time in this scenario is 717.5s.



(a) Travel distance (meters)

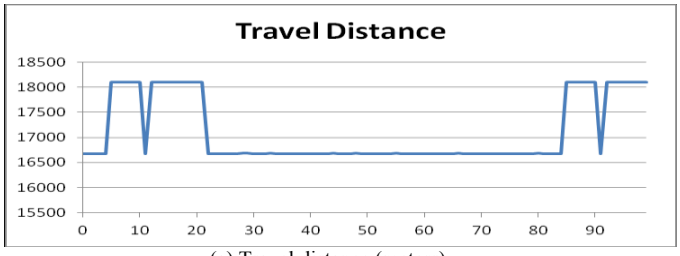


(b) Travel time (seconds)

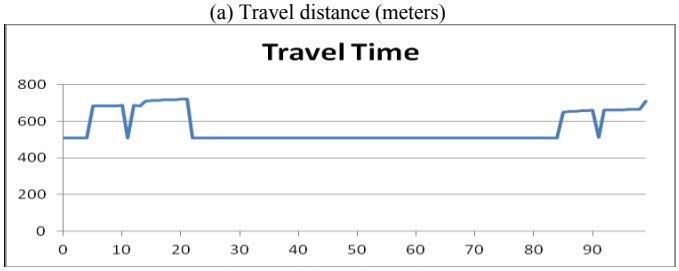


(c) Toll fee (yuans)

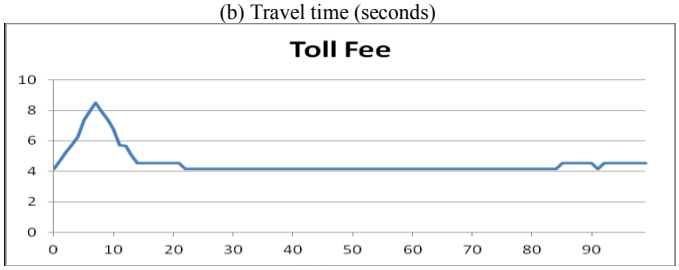
Figure 4. VANET with bad RSU scenario



(a) Travel distance (meters)

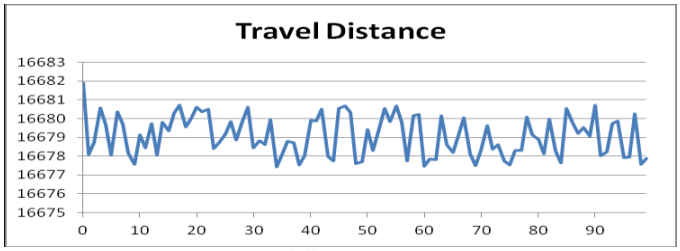


(b) Travel time (seconds)

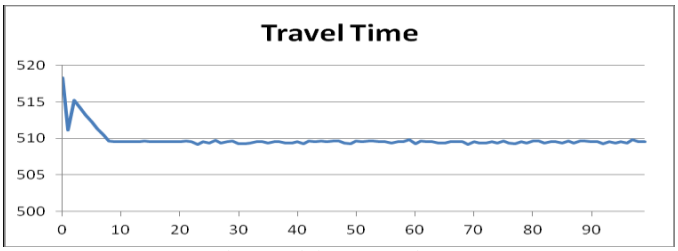


(c) Toll fee (yuans)

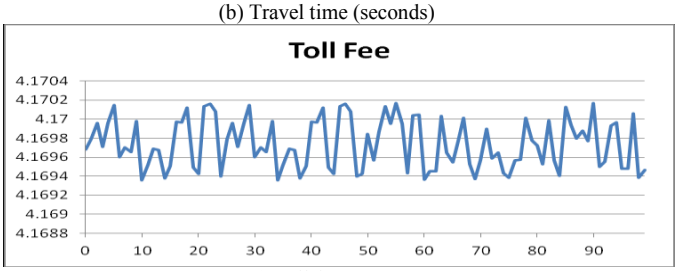
Figure 5. VANET with bad vehicles scenario



(a) Travel distance (meters)



(b) Travel time (seconds)



(c) Toll fee (yuans)

Figure 6. Free-VANET scenario

D. *Analysis and Suggestion*

In Fig. 4 (a) and (b), the curves of travel distance and travel time are almost flat except two sharp bends near vehicle 11 and vehicle 91. This is because only bad vehicle 11 and bad vehicle 91 are not affected by the malicious message. In Fig. 4 (c) the slops are produced by both real-time traffic based toll price algorithm and the malicious messages spread by bad RSU. In Fig. 5 (a) and (b), the two sharp bends near vehicle 11 and vehicle 91 are also created by the malicious message. But the curves in Fig. 5 (a) and (b) are not as flat as those in Fig. 4 (a) and (b), which is because only vehicles close to the bad vehicle 11 and bad vehicle 91 are affected by the malicious message. Fig. 5 (c) shows the traffic is smooth and toll fee keeps on the lowest degree after about vehicle 22. And the slops in Fig. 5 (c) are also produced by real-time traffic based toll price algorithm and the malicious messages spread by bad vehicles. In Fig. 6 (a), (b), and (c), three curves all varies in a very narrow range separately. That means a normal status of all 100 vehicles without effect of malicious message.

From Fig. 4 and Fig. 6, we can find that the toll fee of each vehicle in VANET with bad RSU scenario is much higher than that in Free-VANET scenario, and the total toll fee in VANET with bad RSU scenario is 60.60% more than that in Free-VANET scenario. The similar result can be found in travel distance and travel time in these two simulation scenarios. The total travel distance and travel time is 8.39% and 43.63% more than that in Free-VANET scenario individually. In VANET with bad RSU scenario, every vehicle in the toll expressway will be affected by RSU, so almost the travel distance and travel time of each vehicle in VANET with bad RSU scenario is more than that in Free-VANET scenario, that means bad RSU controlled by attacker can make great effect on ITS.

From Fig. 5 and Fig. 6, we can find that the toll fee of each vehicle in VANET with bad vehicles scenario is much higher than that in Free-VANET scenario, and the total toll fee in VANET with bad vehicles scenario is 9.08% more

than that in Free-VANET scenario. The similar result can be also found in travel distance and travel time in these two scenarios. The total travel distance and travel time is 2.57% and 10.12% more than that in Free-VANET scenario individually. In VANET with bad vehicles scenario, only several vehicles before and after bad vehicles in the toll expressway can be affected by bad vehicles, so the travel distance and travel time of affected vehicles in VANET with bad vehicles scenario is more than that in Free-VANET scenario. And when bad vehicles ran out of the range of concerned area, it's forging message will not make effect on other vehicles. So generally the bad vehicles can only make trouble in a limited area and time. But if the bad vehicles can find a way to spread the message to all the vehicles in the whole network, of course, that's another matter.

The countermeasures must be applied to solve these problem of VANET applications. From the comparisons above, we can find that message attack through hijacked vehicle and RSU can compromise VANET applications, and make great trouble on public transportation. The inherit problem is lack of content check and user authentication. So we may build up a group-judge mechanism to ensure the truth of disseminated message in the VANET. And we also need enhance the authentication of sending message, and monitor all the notifies in the VANET, while improving the ability of road network. In fact there are many attacks and threat listed in Section III, which can compromise the ITS and disturb normal traffic deadly, such as DoS attack, message replay, and so on.

V. Challenges and Limitations

The general testing framework based on Veins allows us securing transportation control systems by penetration testing and message customization which deal well with the threat concerning message. But a lot of security problems may still exist, such as social engineering threat, structure vulnerability, energy disturbing power interruption in physical layer, and so on. In such situation, VANET applications should be not only tested by message testing and cyber attack in communication view but also checked in system level and physical view.

With the development of VANET applications, the application protocols will be more and more in the future, so combining different protocols of message customizations into one framework will be a coming challenge, which requires extending the testing framework described in Section IV with new protocol stack modules.

VI. Conclusion

Securing VANET applications is a very important concern in smart city. Because VANET applications are deployed openly and widely in the public transportation, more and more attacks and threats can greatly affect VANET applications in availability and compromise its integrity and confidentiality, even make road traffic much worse than before. This paper: (1) provides the security requirement of VANET applications and a classification of VANET threats, which make a good guidance of VANET application testing concern; (2) presents a general testing framework based on Veins for securing VANET applications; (3) demonstrates the testing framework by a selected proof-of-concept examples about bad vehicles and RUS in VANET security compromise. (4) analyzes the challenges and limitations of the proposed general testing framework. In the future, we will enrich the modules in the testing framework for supporting various VANET applications in smart city and simulating variable cyber threats from physical attacks to social engineering attacks.

Acknowledgment

The authors would like to acknowledge Dr. Gopal Gupta, the head of Computer Science Department, the University of Texas at Dallas in USA for his great support in their co-research program, acknowledge China Scholarship Council for their financial support and research project. They are also grateful to the editor and the anonymous reviewers for their comments and suggestions on improving the presentation of this work.

References

[1] Rene Oliveira, Carlos Montez, Azzedine Boukerche, and Michelle S. Wangham, "Reliable data dissemination protocol for VANET traffic safety applications", Ad Hoc Networks, Issue 63, 2017, pp. 30-44.

[2] Rodolfo I. Meneguette and Azzedine Boukerche, "SERVitES: An efficient search and allocation resource protocol based on V2V communication for vehicular cloud", Computer Networks, Issue 123, 2017, pp. 104-118.

[3] Prinkle Sharma, Hong Liu, Honggang Wang, and Shelley Zhang, "Securing wireless communications of connected Vehicles with artificial intelligence", 2017 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1-7.

[4] National ITS Architecture V8.1. http://local.iteris.com/arc-it/, 2018-2-19.

[5] Hacène Fouchal, Geoffrey Wilhelm, Emilien Bourdy, Geoffrey Wilhelm, and Marwane Ayaida, "A testing framework for Intelligent Transport Systems", 2016 IEEE Symposium on Computers and Communication (ISCC), pp. 180-184.

[6] Igor Kabashkin, "Reliable v2x communications for safety-critical intelligent transport systems", 2017 Advances in Wireless and Optical Communications (RTUWO), pp. 251-255.

[7] Hongyuan Zhan, Gabriel Gomes, Xiaoye S. Li, Kamesh Madduri, Alex Sim, and Kesheng Wu, "Consensus ensemble system for traffic flow prediction", IEEE Transactions on Intelligent Transportation Systems, vol. PP, Issue 99, 2018, pp. 1-12.

[8] Thiago Mendonca, Aldo Fabregas, and Troy Nguyen, "Application-driven traffic sensor system acceptance tests for intelligent transportation systems", 2017 Annual IEEE International Systems Conference, pp. 1-8.

[9] Veins: The open source vehicular network simulation framework, http://veins.car2x.org/, 2018-03-18.

[10] Openstreetmap,http://www.openstreetmap.org/#map=12/40.0049/116.5311, 2018-03-30.

[11] C. Sommer, S. Joerer, and F. Dressler, "On the applicability of two-raypath loss models for vehicular network simulation", IEEE Vehicular Networking Conference (VNC), 2012, pp. 64-69.