Privacy-Enabled Probabilistic Verification in Broadcast Authentication for Vehicular Networks

by

Kanika Grover

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
December 13, 2014

Keywords: VANET, Broadcast, Authentication, Privacy, ECDSA, Wireless

Approved by

Alvin Lim, Chair, Associate Professor of Computer Science and Software Engineering
Prathima Agrawal, Ginn Distinguished Professor of Electrical and Computer Engineering
David Umphress, Associate Professor of Computer Science and Software Engineering
Xiao Qin, Associate Professor of Computer Science and Software Engineering

Abstract

Vehicular Ad hoc Networks (VANETs) possess an extremely dynamic nature accompanying the high mobility feature. In VANETs, each vehicle sends out safety messages at regular intervals of $100 - 300$ ms. Since the purpose of VANETs is to ensure the safety of human life on the road, therefore, it becomes extremely important to secure these messages. IEEE 1609.2 security standard for VANETs recommend the use of secure Elliptic Curve Digital Signature Algorithm (ECDSA) signatures for authenticating broadcast messages.

ECDSA is a digital signature based on elliptic curve cryptography such that the elliptic curve is defined over a finite field of a prime number. It is considered to be very secure because solving its discrete logarithmic is hard, with primes of sizes 224, 256-bits and greater. Yet, ECDSA is computationally expensive, an ECDSA signature generation takes 4 milliseconds while the verification takes 22 milliseconds on a 400 Mhz processor. Besides, when all vehicles will be broadcasting messages at a frequency of 10 Hz, the verification queue size will increase at a rapid rate. Since the messages are valid only for a certain time period, some of them will time out waiting to be verified. Malicious vehicles can take advantage of this fact by increasing signature verification time through signature flooding of fake messages.

Therefore, smart verification strategy is required. Hence, we design a probabilistic verification method using highly secure ECDSA. At the same time, we make available a privacy controlled mechanism, where Registering Authorities (RA) are the entities responsible for disclosing the original identities of the vehicles communicating with pseudonyms. In accordance, with the early deployment stages of VANETs, our solution does not require a strong backbone of the infrastructure entities RA. Another advantage of our solution is that the vehicles use the information available in the broadcasts to compute the probability. Thus, it does not increase the communication overhead of the broadcasts.

Our V2I communication for privacy controlled mechanism is a lightweight mutual authentication mechanism such that it guarantees the authenticity of both the infrastructure and vehicles. Our V2V communication employs a one-by-one verification approach to support the dynamism in vehicular ad hoc networks. Our method is based on the distance and direction of the communicating vehicles with respect to each other, i.e., whether the vehicles are coming close or moving away. All our simulations are performed on realistic VANET scenarios, generated with the help of urban mobility model SUMO on real city map and Nakagami propagation model. Therefore, this work presents a practical privacy enabled probabilistic signature verification solution for realistic VANET scenario.

This approach integrates security and privacy of vehicles in VANET, such that the vehicles cannot be compromised by an outsider and messages cannot be forged by an attacker. We enlist the harmful attacks for VANETs and analyze our scheme to make sure that all the discussed attacks are prevented. We also develop a reactive and adaptive channel hopping countermeasure for DoS jamming attack, which uses weight based channel selection, such that the weight of detected jammed channel(s) is reduced while increasing the weight and hence probability of selection of un-jammed channels.

We implemented the proposed scheme in an event based network simulator, ns2. We compare its performance with IEEE Std. 1609.2 and two most widely used broadcast authentication algorithms, TESLA and Signature Amortization. Our results show that our scheme has an average of 68% reduction in message loss caused by delay in the verification queue. We also obtain a high packet processing ratio as the number of in-range broadcasting vehicles is varied. When packet error rate is introduced in the network, an average of 80% improvement such that the minimum average improvement is 46% is obtained by our scheme.

Acknowledgments

There are many people in Auburn to whom I owe much gratitude for helping me pursue my doctoral dreams. Foremost among them is Dr. Alvin Lim, who has truly been an outstanding advisor. Without his continuous support and guidance, this dissertation would never have been possible. I would also like to thank Dr. Prathima Agrawal, Dr. David Umphress and Dr. Xiao Qin for serving as members of my advisory committee. Thanks are also in order for Dr. Shiwen Mao, the external reader for my dissertation, for reviewing this document.

I would also express my gratitude to Dr. Kai Chang, Dr. David Umphress, Dr. Wei-Shinn Ku and Dr. Daniela Marghitu for shaping my graduate student career at Auburn. I would also like to acknowledge the efforts of Ms. Michele Wheeles, Ms. Jo Lauraitis, Ms. Barbara Mccormack, Ms. Penny Christopher and Ms. Jennifer Jackson in helping me keep my school and immigration paper work in order.

My thanks also go out to my lab-mates for creating a cheerful lab ambience, in particular, Seungbae Lee, Dongjin Kim, Song Gao and Jian Fang. I equally express my acknowledgment to all my colleagues, especially, Yasmeen Rawajfih and Gautam Dudeja. I would also like to thank all my friends in Auburn for the wonderful time I spent here that has left an inerasable imprint on my heart, especially, my Kannada friends. I am also grateful to the valuable advices of Dr. Qing Yang and Dr. Santosh Kulkarni.

Above all, I would like to express my deepest gratitude to my family for their love, compassion and support in my endeavor. Together they define my existence and it is to them that I lovingly dedicate this work.

Table of Contents

Abstract ii

Acknowledgments iv

List of Figures viii

List of Tables xi

1 Introduction and Motivations 1

1.1 Wireless Vehicular Ad Hoc Networks 1

1.1.1 Communication in VANETs 2

1.1.2 Need for authentication 2

1.2 Authentication in Wireless Ad Hoc Networks 3

1.2.1 Broadcast Authentication 4

1.3 Challenges of Broadcast Authentication in VANETs 5

1.4 Contributions of this work 6

1.5 Structure of Dissertation 7

2 Background 8

2.1 Broadcast Communication for VANETs 8

2.2 Attacks on Vehicular Communication 9

2.2.1 Threat Model 9

2.2.2 Attacks 11

2.3 Broadcast Authentication Properties 12

2.4 Realistic VANET Scenario 14

2.4.1 Network Simulator for Wireless Access in Vehicular Environments . . 14

2.4.2 Mobility Model for Urban Environment: SUMO 14

2.4.3 Radio Propagation Model: Nakagami 17

3 Jamming and Anti-jamming in Wireless Ad Hoc Networks 19
3.1 Jamming and Jammers 19
3.2 Anti-Jamming: Jamming Detection and Countermeasure 20
3.3 Most Popular Countermeasure to Jamming: Channel Hopping 21
3.4 Roulette Wheel Scheme for Channel Selection 22
3.4.1 Weight-based channel selection 22
3.4.2 Algorithmic Details 25
3.4.3 Evaluation 26
3.4.4 Comparison with related work 27
4 IEEE Std. 1609.2 and Statement of Problem 28
4.1 IEEE Std. 1609.2 VANET Broadcast Authentication: Theory 28
4.2 IEEE Std. 1609.2 VANET Broadcast Authentication: Simulations 30
4.3 Problem Statement 34
5 Related Work 36
5.1 Authentication in VANET 36
5.2 Privacy in VANET 37
5.3 Shortcomings of Existing methods 40
6 Proposed Solution 42
6.1 Conceptual Overview 42
6.2 Detailed Solution 44
6.2.1 System Initialization 44
6.2.2 Vehicle Registration and Mutual Authentication 45
6.2.3 Pseudonym Generation 48
6.2.4 Message Authentication 49
7 Results and Discussion 54
7.1 Simulations Settings 54
7.2 Results 54

7.3 Discussion 63
7.3.1 Feature comparison with Existing Schemes 65
7.3.2 Vehicle Registration Overhead Comparison 65
7.3.3 Mobility Prediction Model Validation 67
7.3.4 Probability Estimation Model Validation 78
7.4 Security Analysis 81
7.4.1 Identity Attack 81
7.4.2 Bogus Information Attack 81
7.4.3 Message Forgery Attack 81
7.4.4 Replay Attack 82
7.4.5 Location Traceability 82
7.4.6 DoS Jamming Attack 82
8 Conclusions and Future Work 83
8.1 Conclusion 83
8.2 Future Work 84
Bibliography 86

List of Figures

1.1 Need for authentication in VANETs 3

2.1 Comparing TwoRayGround and Nakagami RF propagation 18

3.1 Types of jammers in wireless networks 20

3.2 Performance Evaluation 27

4.1 Network Layout Representation 30

4.2 Message Loss caused by verification queue delay in IEEE 1609.2 33

4.3 Packet Processed Ratio vs. OBUs in range (IEEE 1609.2) 34

6.1 Vehicle Registration and Mutual Authentication between RA and Vehicle's OBU 47

7.1 Message Loss caused by verification queue delay for 100ms broadcast interval. . 55

7.2 Message Loss caused by verification queue delay for 200ms broadcast interval. . 56

7.3 Message Loss caused by verification queue delay for 300ms broadcast interval. . 57

7.4 Effect of Packet Error Rate on Message Loss for 100ms broadcast interval. . . . 58

7.5 Effect of Packet Error Rate on Message Loss for 200ms broadcast interval. . . . 59

7.6 Effect of Packet Error Rate on Message Loss for 300ms broadcast interval. . . . 60

7.7 Packet Processed Rate vs Number of In-range Vehicles for 100ms broadcast interval. 61

7.8 Packet Processed Rate vs Number of In-range Vehicles for 200ms broadcast interval. 62

7.9 Packet Processed Rate vs Number of In-range Vehicles for 300ms broadcast interval. 63

7.10 Error in Location Estimation using our Mobility Prediction Model 68

7.11 Percentage Change of Probability Values vs. Number of In-range Vehicles – 100ms broadcast interval 69

7.12 Percentage Change of Probability Values for error in mobility prediction – 100ms broadcast interval 70

7.13 Message Loss Ratio for error in mobility prediction – 100ms broadcast interval . 71

7.14 Percentage Change of Probability Values vs. Number of In-range Vehicles for 200ms broadcast interval 72

7.15 Percentage Change of Probability Values for error in mobility prediction – 200ms broadcast interval 72

7.16 Message Loss Ratio for error in mobility prediction – 200ms broadcast interval . 73

7.17 Percentage Change of Probability Values vs. Number of In-range Vehicles for 300ms broadcast interval 74

7.18 Percentage Change of Probability Values for error in mobility prediction – 300ms broadcast interval 74

7.19 Message Loss Ratio for error in mobility prediction – 300ms broadcast interval . 75

7.20 Message Loss Ratio for Past Track History – 100ms broadcast interval 76

7.21 Message Loss Ratio for Past Track History – 200ms broadcast interval 77

7.22 Message Loss Ratio for Past Track History – 300ms broadcast interval 77

7.23 SUMO scenario for probability estimation validation 78

7.24 Probabilitiy estimation when vehicles are coming close 79

7.25 Probabilitiy estimation when vehicles are moving away 80

List of Tables

2.1 Attacks and Threat Categories 10
2.2 IEEE 802.11p PHY and MAC Parameters 15
2.3 SUMO Features 16
3.1 Weight-based Channel Selection 24
3.2 Range of Channel Selection 25
4.1 SUMO Vehicle Attributes 31
4.2 Nakagami Parameters for Urban Model 31
5.1 Related Work 38
5.2 Shortcomings of Existing VANET Authentication and Privacy Schemes 41
6.1 Table of Notations 44
7.1 Comparison of Delay in Authentication 64
7.2 Feature Comparison with other schemes 66
7.3 Vehicle Registration Overhead 67

Chapter 1

Introduction and Motivations

1.1 Wireless Vehicular Ad Hoc Networks

Wireless ad hoc network formed by vehicles on the road are called Vehicular Ad hoc Networks (VANETs). In this network moving vehicles can communicate wirelessly with nearby vehicles and the road side infrastructure units (RSUs). Vehicles possessing the capabilities to form and communicate in a VANET are equipped with wireless On-Board Units (OBUs). Federal Communications Commission has provided seventy-five megahertz of spectrum in 5.9 GHz band (precisely, $5.850 - 5.925$ GHz) for intelligent transportation system (ITS) [1].

VANETs are special characteristics ad hoc networks [2] because they have dynamic and rapidly changing topologies due to their high mobility factor. Moreover, VANETs also experience frequently changing network densities. For instance, during the morning and evening hours, heavy traffic is seen as compared to the afternoon and even less at nights. On the other hand, weekend nights might have more traffic as compared to weekdays. Since vehicular networking integrates the wireless as well as mobility aspects of computing, it becomes non-trivial to handle the integrity of vehicular communications.

Application area of VANETs is very wide encompassing safety and non-safety applications. Safety applications include real time operations for benefiting drivers and passengers, such that accidents can be minimized and traffic conditions can be improved. For example, inter-section collision warning (ICW), electronic emergency brake light (EEBL), road congestion notification (RCN). While the non-safety applications include services such as road status, parking availability, toll collection services (TCS).

Safety applications are a major reason for the development of VANETs, so as to ensure real time safety of human life. According to the latest report by National Highway Traffic

Safety Administration, the year 2012 saw $33,561$ human deaths per 100 million vehicle miles travelled [3], a 3.3-percent increase from year 2011. Considering that VANETs are approaching their deployment in the year $2014 - 2015$ [4,5], it is essential to assure VANET security and privacy alternatives, prior to it's actual implementation.

1.1.1 Communication in VANETs

There are two types of communications in VANET, Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I). The goal of vehicular networking and communication can be achieved by a correct balance of these two communication techniques.

- V2V communication: the means of communication between any two vehicles is primarily broadcast of beacons periodically and at regular intervals by vehicles. These beacons are called Basic Safety Messages (BSM).
- V2I communication: the means of communication can be split into i) unicast messages between vehicles and RSUs, and ii) broadcast of geographical/service information by RSU to all vehicles in vicinity in the form of WAVE Service Advertisement (WSA), where WAVE is Wireless Access for Vehicular Environments.

The above description indicates that safety and warning messages are primarily broadcasted in vehicular networks [6].

1.1.2 Need for authentication

Wireless communication is not reliable. For instance, let us consider a VANET scenario shown in Figure 1.1, where an accident has taken place at the intersection. The vehicles involved in the accident immediately broadcast messages informing the neighboring vehicles about the accident. Nonetheless, there may be a malicious vehicle which can modify the original content of the message and re-broadcast it further. The vehicles receiving erroneous information are not aware of the accident. Such circumstances contradict the purpose of VANETs.

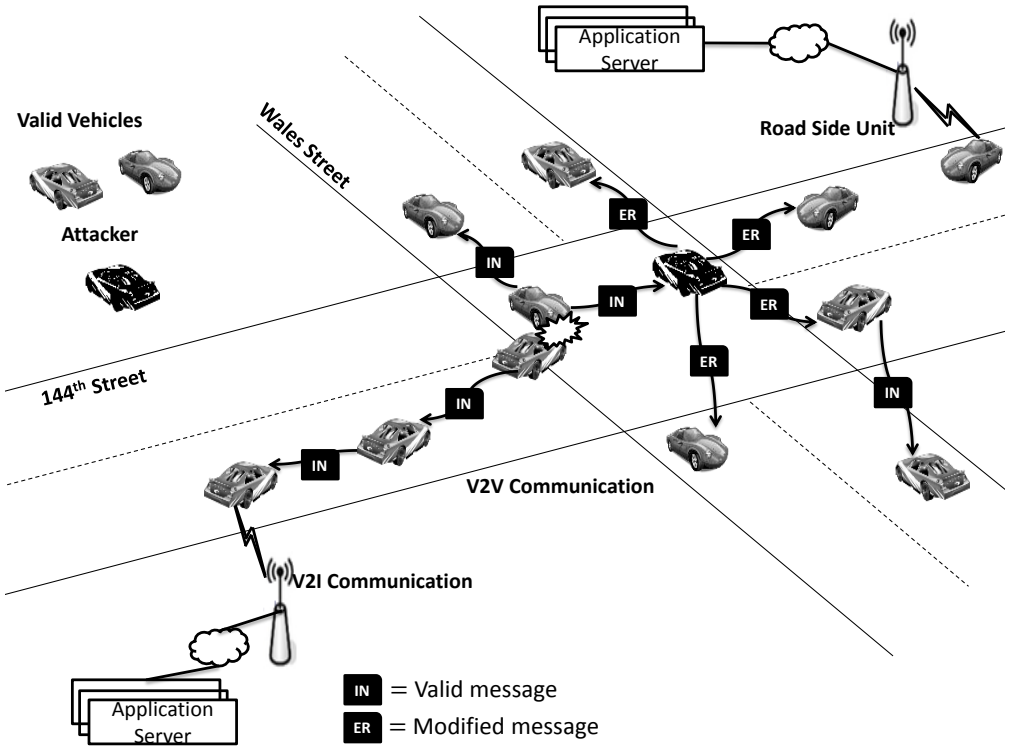

Figure 1.1: Need for authentication in VANETs

Therefore, the authentication of these broadcast messages is important to assure the identity of the sources from which the message originate and the integrity of the message to confirm that it has not been modified in transit.

1.2 Authentication in Wireless Ad Hoc Networks

To best understand wireless ad hoc authentication, we start with the basics of authentication in wireless networks. Authentication in wireless networks can be defined separately for both infrastructure and ad hoc networks. While examining the authentication schemes, we observe that all infrastructure authentication schemes concentrate only on authenticating the wireless devices to be used in the communication. Both the sender and the receiver devices have to be authenticated in a current session before they can communicate with each other. On the other hand, authentication schemes for ad hoc networks can use either

device authentication by establishing a secure communication link or can perform message authentication based on pre-shared information.

Infrastructure networks have all the wireless devices connected to access points such that all the communicating devices are required to complete the authentication process with the access point (AP) before starting to communicate with any other network device. For authentication, wireless devices send an authentication request to the AP. AP asks the wireless device to input a pre-shared-key/MAC address/encrypt a challenge. If the desired response is received, the wireless device is authenticated for communication in the network [7]. Once the devices are authenticated, communication between authenticated devices is assumed to be secure and separate messages are not required to be authenticated individually.

On the other hand, in ad hoc networks, authentication of two wireless devices follows a mutual authentication process as both the clients need to trust each other for a secure communication. They can use a standard SSL/TLS key exchange and establish an authentication and a secure communication wireless link. In other cases there can be a central authority to authenticate all devices connecting to the network. Unicast communication is simple, device A speaks to device B and device B speaks to device A. They authenticate by using a pre-shared key or message authentication codes (MAC) such that the authentication information is already exchanged between communicating partners. Each of these requires a key management protocol.

1.2.1 Broadcast Authentication

In broadcast transmission, one device is transmitting information; while on the receiving end are all those devices in the transmitting range of the sender. It is not feasible to authenticate each of the receivers with the sender as the receivers can be infinite. Besides, communicating the same information to each device individually will consume a lot more time, power and energy. Hence, infrastructure authentication schemes are not suitable for

this scenario. Due to the above stated reasons, broadcast authentication seems to prefer message authentication to confirm the originality of the source of message.

Authentication schemes for ad hoc networks, discussed above, hold for unicast authentication. The drawback can be observed from the following example, one device is broadcasting and multiple devices are receiving the broadcast message. Because the authentication information is pre-shared, all the devices are aware of the information. Hence any one of them can change the message, leading to message forgery attack; or can imitate to be someone else and can send a message on someone else's behalf, leading to identity attack and bogus information attack. Moreover, since the information is false, it can lead to severe consequences, in real life situations, if the receivers take action regarding the information as authentic.

Therefore, broadcast authentication aims at achieving message authentication to confirm that the message has not been modified while in transit from the sender to the receiver. Besides, because no prior authentication information can be shared as a prevention to message forgery and identity attacks, hence, it becomes important to confirm that the message is sent by a real wireless device [8].

1.3 Challenges of Broadcast Authentication in VANETs

In accordance with the DSRC standard [9], each vehicle is required to send broadcasts at an interval of 100-300 milliseconds. VANET security standard [10] recommends the use of Elliptic Curve Digital Signature Algorithm (ECDSA) [11] for authentication services. Each ECDSA signature verification requires 22 milliseconds of computational time on a vehicle's 400 Mhz On-board Unit (OBU) [12]. As the network density increases, the number of messages in the verification queue increases, increasing the time of verification.

Most of the broadcasted messages are time critical, this inflates the verification delay issue because messages waiting for long time in verification queue will time-out and expire. Therefore, it is required to create a smart verification strategy. Furthermore, each vehicle carries along it's personal data and location, the disclosure of which can lead to identity thefts

and location traceability attacks. The confidentiality and privacy of personal information should be maintained unless overridden by the application or approved by user.

Moreover, in VANETs the vehicles are constantly moving at high velocities and network topologies are changing constantly. Therefore, if the authentication information is sent either prior to message or after the broadcasted message (for example, a shared key), the receiver could have travelled outside the range of the sender and will not be able to receive the authentication information. Hence, it is only appropriate to sent the authentication information along with the broadcasted message.

The above mentioned challenges are accompanied by the requirements of broadcast authentication in a wireless network; primarily, to authenticate the origin of a broadcasted message, to verify that the message has not been tampered with while it was in transit, authenticate the message immediately upon arrival and to make sure that non-repudiation is maintained. Non-repudiation specifies that the originator of the message cannot deny the message transmission.

1.4 Contributions of this work

- We simulate broadcast transmission in VANETs by creating a broadcast scenario, where vehicles exhibit dynamically changing mobility pattern based on urban mobility traces generated from SUMO mobility model, and realistic radio propagation using Nakagami propagation model.
- We implement broadcast authentication for inter-vehicular communication in our test scenario as recommended in IEEE 1609.2 security standard for Vehicular Ad hoc Networks that suggests the use of ECDSA. We study the existing verification delay problem.
- We adopt symmetric key cryptography for V2I communication. This makes the process less computationally expensive. It also accommodates a mutual authentication between the Registering Authority and each vehicle.

- We propose a *mobility-based* solution that supports the dynamic nature of VANETs for V2V broadcast authentication. In particular, we address the verification delay problem using probabilistic verification with ECDSA to reduce the message loss ratio.
- We compare our proposed solution for message verification with IEEE 1609.2 and two widely used solutions, Signature Amortization and TESLA, through ns2 simulations. The results show that our scheme yields the lowest packet loss ratio, in addition to ensuring conditional privacy of the OBUs.
- We study the Denial-of-Service jamming attack, classify various kinds of jammers, and analyze the existing jaimmg detection and countermeasures. Eventually, we develop a reactive and adaptive channel hopping method, countermeasure to jamming, using roulette wheel channel selection.

1.5 Structure of Dissertation

The rest of the paper is organized as follows. Chapter 2 presents the background on VANETs, attacks on VANETs, VANET authentication requirements and realistic simulation scenario. Chapter 3 presents a discussion on jamming, jammers, anti-jamming and finally presents an adaptive channel hopping method. Chapter 4 describes the suggested security standard for VANET, highlighting the verification delay problem. Chapter 5 lists the related work on authentication and privacy in VANETs along with their shortcomings. Chapter 6 proposes a privacy-enabled probabilistic verification algorithm. Chapter 7 presents the results, discussions and security analysis. After summarizing our findings, Chapter 8 presents the conclusions and future work.

Chapter 2

Background

In this chapter, we first explore the broadcast communication in VANETs. Further, we study the threat models and enlist some of the attacks on inter-vehicular communication that require a definite resolution. Then we study the properties essential for broadcast authentication in VANETs. We also design a realistic VANET scenario for simulations, such that we use an urban mobility model and radio propagation model.

2.1 Broadcast Communication for VANETs

In VANETS, information is typically broadcasted, either in the form of WAVE Service advertisement (WSA) or Basic Safety Message (BSM). Through WSA, a WAVE device advertises the service that it offers to all other WAVE devices in vicinity, detailed in IEEE 1609.3 [13]. Usually RSUs broadcast WSA, but service offering OBUs can also do so [14]. Besides, BSM announces the status of a vehicle. All DSRC equipped OBUs transmit BSM periodically and at regular intervals. Security of WSA and BSM communication is maintained by IEEE 1609.2. Since we focus on inter-vehicular safety communications, we will concentrate on BSM.

BSM are used for transmitting warnings or traffic related information, in addition to vehicles' status, including vehicle's location and velocity. The advantages of using BSM are: i) interoperability of vehicle safety applications, without standardization of the applications, ii) various vehicle safety applications that can be performed with the same message, and iii) backward-compatible with future developments due to flexible expansion of messages [10]. These safety messages are broadcasted by vehicles every $100 - 300$ milliseconds [9]. BSM will generally be encapsulated in a WAVE short message (WSM) data [14]. WSM follow

WAVE Short Message Protocol (WSMP), which allows a rapid transmission of messages in an environment where radio frequency varies rapidly. The authentication of these broadcasted messages is critical because it involves real-time safety of drivers and passengers.

2.2 Attacks on Vehicular Communication

VANET operates on wireless channels. Considering the fact that a wireless channel is an extremely unreliable medium of communication and since VANET applications involve human life; it becomes very important to consider various kinds of attacks that are possible in VANETs. Below, we describe the threat model followed by the attacks in VANETs.

2.2.1 Threat Model

Threat model for VANET can be classified according to five categories motivation, membership, mobility, strategy and scope. Some of these are given by [15] and [16]. Below we elaborate on each of the categories of threat model and then consider the combination of these used by the attacks given in the next sub-section.

1. Motivation of Attacker: It can be either *Malicious* or *Rational*. The main aim of malicious attacker is to disrupt the working of network or members of a network. While a rational attacker is more considerate of his own benefits from the attack. For example, a malicious attacker might disable the brakes of a vehicle or damage the software. Whereas a rational attacker may try to obtain information such that it can unlock the vehicle.

2. Membership of Attacker: It can be either *Insider* or *Outsider*. Insider is a member who has or had authorized access to a network. Therefore, it is legally eligible to communicate with other members of the network. Insider threats include misuse of access privileges or disclosure of confidential information. In contrast, outsider is an unauthorized member trying to communicate with a network. Because an outsider does

not have licensed access, it can only launch limited types of attack. Outsider threats mostly include Denial-of-Service (DoS) or Distributed Denial-of-Service (DDoS).

3. Mobility of Attacker: An attacker's mobility state can be classified as *static* (zero velocity) or *mobile* (non-zero velocity). Attacker's with non-zero velocity can either have constant or adaptive mobility. Adaptive theme requires an attacker to have smart capabilities such that it can monitor and change its velocity in response to its environment to make an efficient attack.

4. Strategy of Attacker: It can be either *Active* or *Passive*. An active attacker can interfere in the network operation by transmitting signals or packets on the wireless channel. Passive attacker, on the other hand, eavesdrop on the wireless traffic to collect information about network or valid vehicles.

5. Scope of Attacker: It can be either *Local* or *Extended*. The scope of an attacker can be limited to a part of VANET, such a scope is called local. However, if the attacker can attack entities in various parts of the VANET, such a scope is called extended.

Table 2.1: Attacks and Threat Categories

Attack	Motivation	Membership	Mobility	Strategy	Scope
Identity Attack	R	I	*	A	*
Bogus Information Attack	R	I	*	A	*
Message Forging Attack	R	I	*	A	*
Replay Attack	*	I	N	A	E
Location Traceability Attack	R	*	N	P	E
DoS Attack	M	*	N	A	*

R - Rational, M - Malicious | I - Insider, N - Mobile | A - Active, P - Passive | E - Extended, * - *All Types*

2.2.2 Attacks

In designing VANET communication, security and privacy are very important aspects. Therefore, we consider attacks harming the security of vehicular communication as well attacks threatening the privacy of vehicles involved in vehicular communication. Understanding specific attacks can be helpful in realizing a solution for those attacks. Table 2.1 gives categories of threat model that each attack satisfies. * in the table means that the attack satisfies all types of the category being addressed.

1. Identity Attack: All vehicular on-board units have a unique identification number. In addition, each broadcast message requires a sender address to be added to the message. If the valid vehicles use their original identification numbers, attacker can easily spoof the identity of a valid vehicle from broadcasted messages. The attacker can use the stolen identity to create fake messages.

2. Bogus Information Attack: Attacker may create false or bogus information to disrupt the activities of legitimate vehicle drivers. For example, an attacker may incorrectly broadcast a message with the information about an accident on the highway. This message will cause the legitimate vehicle drivers to change their routes and can cause congestion on an alternative route.

3. Message Forging Attack: Attacker may try to modify the information in a legitimate message broadcasted by a valid vehicle. The attacker may also try to forge the authentication information in the original message by either spoofing or guessing.

4. Replay Attack: Attacker can collect messages that have been broadcasted by valid vehicles over a period of time. It can then replay these messages at a later time when the messages are no more valid. Moreover, since the attacker is mobile, it can collect messages from another geographical area and then replay it when it reaches an area in which the messages are not applicable.

5. Location Traceability Attack: Attacker may get hold of a number of messages sent by a valid vehicle and can trace its movement pattern by obtaining its physical positions.

6. Denial of Service Attack: The most popular Denial-of-Service (DoS) attack is jamming. VANET being a wireless networks is susceptible to interference and intentional attacks from external sources. The act of jamming blocks the communication channel.

2.3 Broadcast Authentication Properties

In designing VANET communication, security and privacy are very important aspects. Security in broadcast communication signifies authentication. Since authentication of broadcast messages in inter vehicular communication (IVC) is crucial. We enlist the necessary properties for broadcast authentication in VANETs as follows.

1. Source Authentication: This is the main goal of authentication techniques for the broadcast transmission. Validating the identity of the source from which the message originates is the most important property of any broadcast authentication protocol. Hence, it is required that each of the receivers receiving a broadcasted message perform source authentication.

2. Data Integrity: It is essential to verify the integrity of a broadcatsed message, to ensure that intermediate vehicles have not altered the message. Data integrity can be achieved by making sure that the message content has not been modified after being sent by the sender and before being received by the receivers.

3. Non-Repudiation: To confirm that the sender can never deny the sending of a message that it has originated, non-repudiation is required. Digital signatures are the authentication certificates for the digital world similar to an individual's signatures in non-digital world. Therefore, digital signatures prove that the sender is the original author of the message; hence, ensuring non-repudiation.

4. Immediate Authentication: If a message is authenticated upon arrival, without any delay in the authentication mechanism, immediate authentication is achieved. For instance, TESLA [17] does not support this property, because its design includes delayed key disclosure. On the other hand, digital signatures support immediate authentication property if they are not amortized.

5. Computation Overhead: Use of authentication mechanisms increases computation overhead of the system. Some operations such as hashing and XORing, used in MAC based protocols such as TESLA, have less overhead. Whereas modular arithmetic based digital signature algorithm (DSA) are more complex and costly.

6. Robustness to Packet Loss: It ensures that the loss of any packet will suspend/terminate the authentication process. For example, in MAC based protocols with delayed key disclosure, if the packet containing key is lost then the corresponding message cannot be authenticated. However, digital signature schemes are robust to these types of losses as they do not require separate authentication packets.

7. No Time Synchronization: The sender and receivers should not be required to time synchronize with each other, which utilizes extra time and energy to make effective time synchronization processes. In addition in vehicular ad hoc networks, the vehicles are travelling with high velocities. Yet there are some protocols, such as MAC-based protocols (such as TESLA) and some of the digital signatures protocols.

8. No Buffering Overhead: Buffering overhead influences the resources of the wireless communicating vehicles in two ways: either the buffering period is very long or the storage utilization is high. TESLA-based protocols might have to buffer for longer periods if keys are lost or if system is designed for a longer key disclosure. Basically digital signatures support immediate authentication with no need for buffering, while signature amortization or similar schemes require buffering.

9. Scalability: It should be easy to add and remove vehicles in VANET without adversely affecting the authentication process. MAC-based schemes account for the number of senders and receivers that can be included in the authentication protocol. For instance, μTESLA schemes require only the base station to be the sole broadcast sender. On the other hand, digital signature schemes have no issues with the number of senders and receivers as long as each one constructs and verifies its signature independently.

2.4 Realistic VANET Scenario

Because it is not advisable to perform initial tests on a real implementation, where human life is involved, we chose to perform simulations. Our simulations setup imitates actual VANET scenarios due to the following reasons. First, we consider a simulator that can model wireless network applications and protocols. Second, our requirement is to obtain realistic mobility traces from a vehicle mobility model which can reproduce an urban environment. Third, we use a radio propagation model that accommodates the fading characteristics of the wireless channel. We further look into each of the requirements and make appropriate selections.

2.4.1 Network Simulator for Wireless Access in Vehicular Environments

To fulfil the first requirement, we selected an event-based network simulator, ns-2 [18] to model network applications and protocols. Ns-2 contains most of the IEEE 802.11 standards, including IEEE 802.11p extension for Intelligent Transportation Systems (ITS) and the standard to add wireless access in vehicular environments (WAVE) [19]. The parameter values used for IEEE 802.11p are given in Table 2.2.

2.4.2 Mobility Model for Urban Environment: SUMO

To obtain realistic mobility traces, firstly, we investigate the requirements for a realistic vehicular mobility model for an urban setting. According to a survey on mobility models

Table 2.2: IEEE 802.11p PHY and MAC Parameters

IEEE 802.11p PHY		**IEEE 802.11p MAC**	
Parameter	Value	Parameter	Value
Frequency	5.9 GHz	CWMin	1.5
CSThresh	-95 dBm	CWMax	1023
NoiseFloor	-99 dBm	SlotTime	13 μs
PowerMonitorThresh	-102 dBm	SIFS	32 μs
HeaderDuration	40 μs	SymbolDuration	8 μs
PreambleCaptureSwitch	1	PLCPDataRate	6 Mbps
DataCaptureSwitch	1	basicRate	6 Mbps
BaiscModulationScheme	BPSK	dataRate	6 Mbps

for VANETs [20] and study on urban mobility models [21], minimum requirements for a realistic vehicular mobility model in an urban setting are car following model, intersection management, traffic lights, stop signs, speed limitations and multi-lanes. After studying various mobility models for VANETs, and matching the requirements for urban environment we chose *Simulation of Urban MObility* (SUMO). SUMO satisfies the minimum requirements for an urban setting along with other additional ones [22].

SUMO is an open source, microscopic and continuous road traffic simulator which can handle large road networks [23]. SUMO provides a visualization tool and can generate traces that are directly used by network simulators. Validation of SUMO mobility model has been shown by [24]. We highlight some features of SUMO, used for our simulations, in Table 2.3.

Table 2.3: SUMO Features

SUMO feature	Description
Car Following Model	Krauss Model [25]
Intersection Management Strategy	Stochastic Turns [1]
Traffic Lights	Yes
Stop Signs	Yes
Speed Limitations	Yes
Velocity	Smooth [2]
Multilane roads	Yes
Lane Changing	Yes
Trip Generation	Random or Activity-based
Path Computation	Dijkstras
Collision free movement	Yes
Real Map support	Yes
Building Platform	C++
Portable	Yes
Ease of Installation	Moderate
Ease of Use	Hard
Visualization Tool	Yes
Generate traces for ns-2	Yes

[1] Vehicle chooses it own speed and direction according to a probability density function when no path is previously defined.

[2] Vehicle does not abruptly brake or accelerate.

2.4.3 Radio Propagation Model: Nakagami

The most commonly used radio propagation model in ns2 simulations is Two Ray Ground model. Two Ray Ground is a deterministic propagation model, with the main drawback of assuming the same received signal strength for the entire transmission range being considered. However, it is not possible for two receivers at different distances to receive same signal strength from a given sender. Therefore, VANETs need a radio propagation model with fading characteristics, such that fading is dependent on the distance between the sender and receiver. Studies conducted by [26] and [27] have shown that Nakagami propagation exhibits a fading channel for a radio signal. Therefore, we carry out an analysis of Nakagami, a probabilistic radio propagation model. Nakagami probability distribution function has been defined by [19] as follows.

$$f(x) = \frac{2m^m x^{2m-1}}{\Gamma(m)\Omega^m} exp[-\frac{mx^2}{\Omega}]$$

$$x > 0, \Omega > 0, m \geq 1/2,$$

m is the Nakagami fading parameter and Ω is the expected value of distribution and can be interpreted as the average received power. Values of m and Ω are functions of distance. Larger values of m gives less severe fading. This observation can be used to select larger values of m for highways as compared to congested urban settings that demand more dramatic fading effects [28].

To compare the deterministic and probabilistic nature of Two Ray Ground and Nakagami propagation model, we simulated both in an environment with transmission range set to 250 m. The results are given in Figure 2.1.

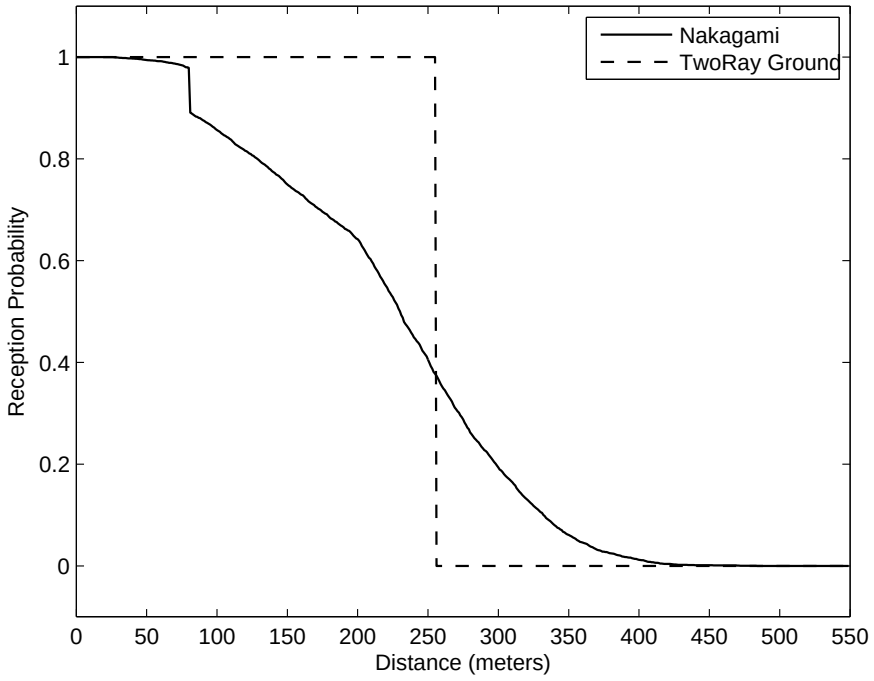

Figure 2.1: Comparing TwoRayGround and Nakagami RF propagation

We observe that Two Ray ground gives a reception probability of 1 for the complete 250 m distance, whereas with Nakagami the reception probability decreases as the distance between the communicating vehicles increases. Hence, we consider Nakagami propagation model for all our further simulations to have a realistic VANET scenario.

Chapter 3

Jamming and Anti-jamming in Wireless Ad Hoc Networks

Wireless vehicular ad hoc networks are prone to DoS attack, "jamming". It can cause performance degradation of a network by interference. Attackers can jam the network using various types of jammers resulting in non-functionality of the network. We study the anti-jamming techniques that can be applied to provide resilience to jamming. Below, we provide a discussion on jamming, jammers and anti-jamming as taken from our paper [29]. Further, we present a genetic weight selection method for anti-jamming channel hopping.

3.1 Jamming and Jammers

Jamming is the activity of blocking the communication channel by achieving one of the following goals: consumption of computational resources; disruption of computational information, state information, or physical network components; or obstructing the communication media between the intended users and the victim [30]. Various types of jammers have been used by several researchers. Jammers can be elementary or advanced; elementary jammers can be classified as proactive or reactive while the advanced ones can be either function-specific or smart-hybrid jammers as shown in Figure 3.1.

Elementary proactive jammers can be constant, deceptive, or random jammers [30]. Their main aim is to keep the channel busy so that the legitimate nodes are not able to access it. On the other hand, reactive jammers, such as RTS/CTS jammer and Data/Ack jammer, disrupt packets sent by one legitimate node to another [31]. Advanced function-specific jammers are the ones having a predefined function like follow-on [32], channel hopping

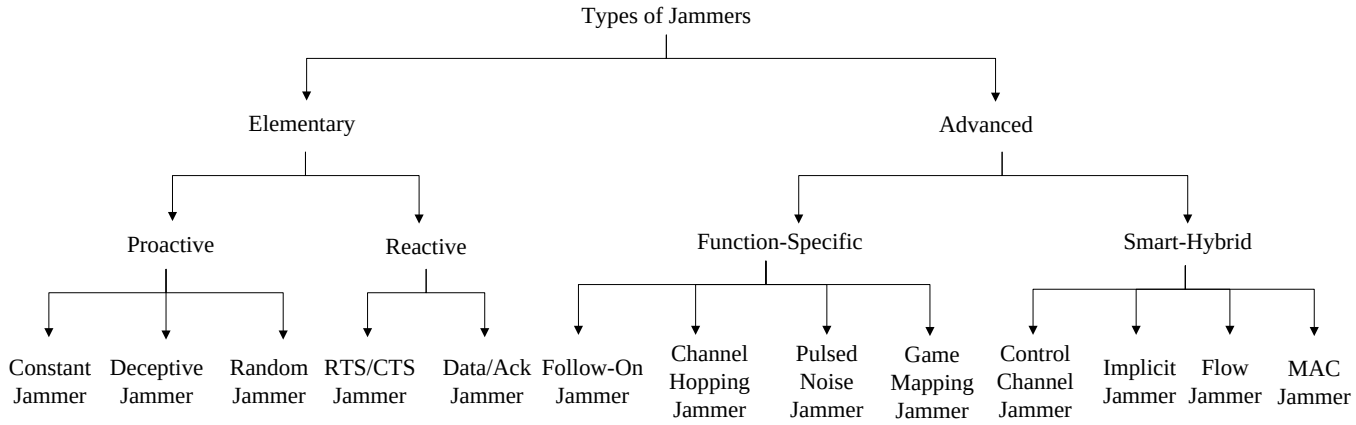

Figure 3.1: Types of jammers in wireless networks

[33], pulsed-noise [34] and game mapping jammers [35]. Smart-hybrid jammers are control-channel jammers [36], implicit jammers [37], flow jammers [38] and MAC jammers [39]. They have power efficiency and effective jamming nature, hence smart.

3.2 Anti-Jamming: Jamming Detection and Countermeasure

Typical jammed network results from two conditions; either the attacker makes the channel unavailable for the sender to access or it corrupts the packets sent by the sender. The former attack can be detected at the sender end using utility threshold and carrier sensing parameters while the latter needs to be detected at the receiver end with the help of packet delivery ratio or signal to noise ratio. In addition, it is required to look into the details of methods which can provide resilience for them. Therefore, we discuss methods providing jamming detection and/or jamming countermeasure.

Authors of [40] have given a detection and mitigation method which maps out the jammed area in wireless sensor networks and routes packets around the affected region. While an evolutionary algorithm to detect jamming at the PHY layer has been proposed in [34], and redirects messages to an appropriate destination node. Authors of [41] propose a hybrid anti-jamming system by combining three defense techniques: base station (BS) replication, base station evasion and multipath routing between base stations. On the other

hand, an effective jamming detection using either location or signal strength consistency check along with packet delivery ratio determination is proposed in [30]. A centralized jamming detection system in [42] computes the jamming index using the signal-to-noise ratio (SNR) and packet dropped per terminal (PDPT) values. Channel surfing (or channel hopping) is another technique proposed in [43].

3.3 Most Popular Countermeasure to Jamming: Channel Hopping

After analyzing all jamming resilience methods, we take into consideration the most popular method, i.e., channel hopping. This scheme seems to be the most appropriate method considering the ease of implementation with no extra hardware requirements. There exist many types of channel hopping solutions to counteract the jamming attacks, such as proactive, reactive, adaptive and code-controlled. Different variations of channel hopping are discussed in [44–48].

Proactive channel hopping is the simplest implementation. Withg proactive scheme, channels are switched after a given time duration has elapsed on the current communicating channel. This takes place irrespective of whether jamming is occurring or not. While the reactive channel hopping hops channels only when jamming is detected in the network. However, due to the impact of energy spill over the adjacent channels, [49] prove that proactive frequency hopping is not very effective.

Reactive schemes usually use channel sensing as a measure of jamming detection and reacts accordingly by hopping channel. Reactive channel selection scheme discussed in [46] is a variation of a pseudo-random number generation. Two other strategies proposed by [44] are straight-forward channel hopping and deceptive channel hopping. In straight-forward scheme, the channel to be hopped onto is selected from the set of unused channels. While in deceptive method, the selection set includes the currently used as well as unused channels.

Adaptive scheme keeps a track of the packet delivery ratio (PDR) of all channels. When the performance (PDR) of the present channel falls below a threshold, communications are

switched to another channel which gives the best PDR value. Each device implementing adaptive channel hopping hops on all channels once every k slots (a slot is defined to be a fixed time interval) to collect the PDR values. Recently, authors of [47] put forward a code-controlled message-driven frequency hopping mechanism. It generates a dynamic hopping pattern each time the channel is changed.

Hence, channel hopping seems to be the most appropriate solution for DoS jamming attacks. Moreover, reactive methods are superior because they use channel hopping only after jamming has been detected. However, existing reactive channel hopping approaches primarily use a random-selection of channels or follow a historic background which makes it easy for jammer to capture the pattern of channels followed by the wireless devices. Though these techniques are far more effective than single-channel solutions due to interference and fading, the threat of jamming attacks determining the hopping pattern may require more adaptive hopping methods.

3.4 Roulette Wheel Scheme for Channel Selection

To overcome the above drawback of ease of pattern assessment by jammers, we suggest a reactive and adaptive channel selection scheme that uses genetic algorithm roulette wheel selection [50] approach for choosing the next communication channel on which the communication is to be hopped onto. We assign weights to the available channels in the spectrum and compute their probability. Upon the detection of a channel being jammed, its weight is reduced which in turn reduces its probability of getting selected while hopping channels. Below we present a weight-based channel selection method following the roulette wheel scheme [51].

3.4.1 Weight-based channel selection

There are various algorithms for selecting elements based on weight. Suppose an element has to be picked randomly from a given set, where each element has a weight; then the

probability of selection of each element with be directly proportional to its weight. For example: $Weight(x) = 1$ and $Weight(y) = 4$. This means that 'y' has four times as much chances to be picked as 'x'. Now, if we have to generate a random number in the interval $[0, 5)$. We will set 'x' to be selected on the generation of '0' while 'y' to be selected on the generation of '1' , '2', '3' or '4'.

In our proposed method, the random number generator will select a number in the range of 0 to 1 and a channel will be selected based on the resulting value. Initially, all N channels will be allotted the same weight, i.e., 1. This makes the probability of selection of each channel as $1/N$. That is, each subset of the possible range of the random number represents a particular channel, where each subset representing a particular channel is initially the same size.

Channel selection is performed based on a roulette-wheel selection technique. If a channel becomes jammed, then the weight of that channel is reduced by a certain amount which is evenly distributed to all other channels in the network. This keeps the total weight of all un-jammed channels constant. This is followed by a random number generation in the range 0 to 1, to determine which channel will be selected next for communication, based on the weight of each channel. This is called the spin of the roulette wheel. Thus, the probability of selecting any one particular channel i is

$$P_i = \frac{W_i}{\sum_{i=1}^{N} W} \tag{3.1}$$

where N is the total number of channels and W_i is the weight allotted to each channel. P_i is directly proportional to W_i, i.e., greater the weight of a channel, higher is its probability of being selected as the next communication channel. This process of channel selection is iterative in nature, such that it will continue until an un-jammed channel is found for successful wireless transmission.

Table 3.1: Weight-based Channel Selection

Channel Number	**Initial Weight**	**Adjusted Weight**	**Initial Probability of selection**	**Adjusted Probability of selection**
1	1	*0.9*	9.0909%	*9.0818%*
2	1	1.01	9.0909%	9.0918%
3	1	1.01	9.0909%	9.0918%
4	1	1.01	9.0909%	9.0918%
5	1	1.01	9.0909%	9.0918%
6	1	1.01	9.0909%	9.0918%
7	1	1.01	9.0909%	9.0918%
8	1	1.01	9.0909%	9.0918%
9	1	1.01	9.0909%	9.0918%
10	1	1.01	9.0909%	9.0918%
11	1	1.01	9.0909%	9.0918%

We assign the initial weight to all channels as 1, hence, the sum of weights $\sum W_i$ of all the channels is equal to N. This value of N is maintained in the system by evenly increasing and decreasing the weights of all the channels when a percentage of weight is deducted from the jammed channel or a percentage of weight is added to the un-jammed channel(s).
For a jammed channel with weight W_i, the new reduced weight will be calculated as:

$$W_i - \frac{W_i}{N-1} \tag{3.2}$$

Then this subtracted weight

$$\frac{W_i}{N-1} \tag{3.3}$$

is evenly distributed among the other $N-1$ channels in the network, except the channel from which it is subtracted.

Taking an example of a system with 11 channels, where jamming is detected on channel 1. We show the computations in Table 3.1 that will take place when this scheme is used to select the next channel for communication. Here we see that 1/10 weight is deducted from

Table 3.2: Range of Channel Selection

Channel Number	**Initial Range**	**Adjusted Range**
1	0 *to* 0.090901	0 *to* 0.0818182
2	0.090901 *to* 0.181818	0.0818182 *to* 0.173636
3	0.181818 *to* 0.272727	0.173636 *to* 0.265455
4	0.272727 *to* 0.363636	0.265455 *to* 0.357273
5	0.363636 *to* 0.454545	0.357273 *to* 0.449091
6	0.454545 *to* 0.545454	0.449091 *to* 0.540909
7	0.545454 *to* 0.636364	0.540909 *to* 0.632727
8	0.636364 *to* 0.727273	0.632727 *to* 0.724545
9	0.727273 *to* 0.818182	0.724545 *to* 0.816364
10	0.818182 *to* 0.909091	0.816364 *to* 0.908182
11	0.909091 *to* 1	0.908182 *to* 1

channel 1 and is equally distributed among the other 10 channels. This causes a change in the probabilities of the channels as well as the probability ranges for each of them.

Table 3.2 shows the range of selection each channel, such that un-jammed channels have a higher chance of being picked as compared to the jammed channel. When we spin the roulette wheel and get a number between 0 and 1, we see that the increase in range corresponding to the unjammed channels increases their probability of selection. As a further consideration, in order to avoid boundary problems, a maximum and minimum weight can be applied to each channel so that the probability of selection remains within certain bounds.

3.4.2 Algorithmic Details

Here we present the algorithm for Roulette Wheel channel selection. It is assumed that detection of jamming has taken place prior to the execution of this algorithm. The detected jammed channel is given as input to this algorithm. Algorithm 1 gives the pseudo code for the channel selector implementation.

Algorithm 1 ChannelWeightAdjustment(jammedChannel)

N = total number of Channels
$delta = weight[jammedChannel]/(N-1)$
$probability[prevChannel] = 0$
for $channel \leftarrow 1$ to N **do**
  if $channel == jammedChannel$ **then**
    $weight[channel] = weight[channel] - delta$
  else
    $weight[channel] = weight[channel] + delta/(N-1)$
  end if
  $probability[channel] = probability[prevChannel] + weight[channel]/N$
  $selectionRange = probability[prevChannel]\ \ to\ \ probability[channel]$
  $probability[prevChannel] = probability[channel]$
end for

3.4.3 Evaluation

Figure 3.2 shows the performance of roulette wheel channel selection algorithm as compared to deceptive channel selection. These experiment runs have been done for the cases when only one channel was left un-jammed, while all the other channels were taken as jammed. For our tests, we have used channel 3 as the un-jammed channel. Our basic purpose is to show that our proposed algorithm converges more quickly than deceptive selection algorithm, taking into consideration the total number of channels in a network.

As seen in Figure 3.2, the deceptive scheme which selects channel based on a pseudo-random number from the complete set of channels, ignores the consideration of a jammed channel. While roulette wheel selection scheme considers the jammed channel and decreases its probability of selection. Though roulette wheel scheme has certain variation due to the value obtained from the spin of the wheel and the slight differences in the probabilities of selection as our range varies from 0 to 1 only. Yet, we can say that its average performance is better than deceptive.

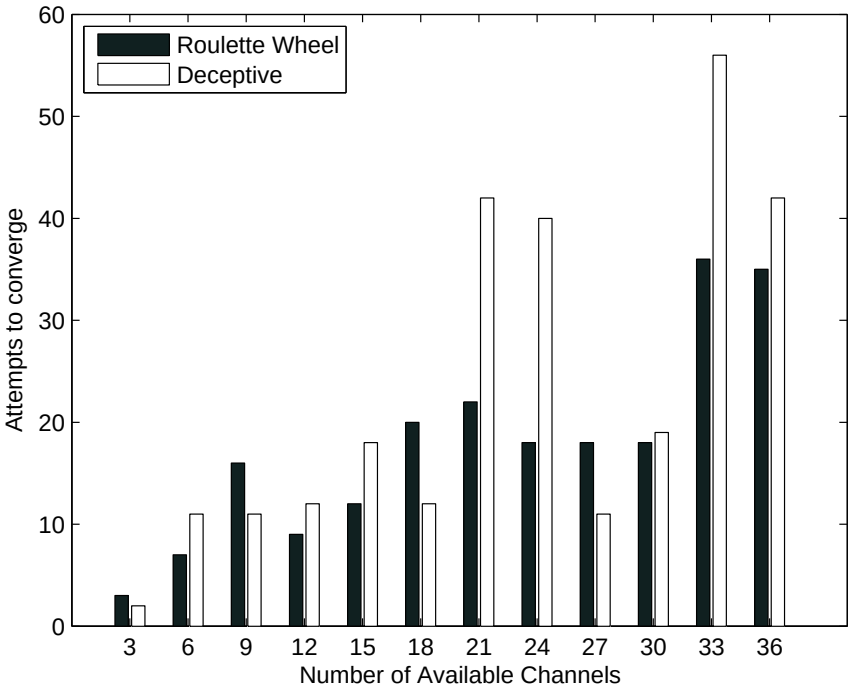

Figure 3.2: Performance Evaluation

3.4.4 Comparison with related work

Compared to proactive, our reactive and intelligent channel selection technique reduces the overhead of switching channels after every predefined interval of time. Moreover, the dynamic nature of this scheme has advantage over the existing schemes such as the heuristic and pseudo-random number generation for the selection of a channel which require the exchange of a seed with all its neighbors. This allows the hop to take place on the same new channel which is selected with the help of the random seed.

The time-based channel selection methods have the weakness of following a predefined pattern of channels, which the jammer can observe and follow. Therefore, the follow-on jammers will more likely render the time-based method ineffective which is difficult to be done in the weight-based method. Our solution also offers benefits by reducing the selection of a poor quality channel as in existing FHSS (Frequency Hopping Spread Spectrum).

Chapter 4

IEEE Std. 1609.2 and Statement of Problem

Using VANET broadcast transmission guidelines, we study the recommended broadcast authentication standard IEEE 1609.2. We study and examine the effects of IEEE 1609.2 on inter-vehicular communication. We will implement IEEE 1609.2 on realistic VANETs and study the problem of long verification delay. We formulate the problem statement after observing the simulation results.

4.1 IEEE Std. 1609.2 VANET Broadcast Authentication: Theory

IEEE 1609.2 security standard for VANETs supports the cryptographic standard Elliptic Curve Digital Signature Algorithm for Inter-Vehicular Communication [10]. ECDSA is a variant of the Digital Signature Algorithm (DSA) which uses elliptic curve cryptography over finite fields. An elliptic curve can be represented by the following cubic equation defined over finite field F_p where p denotes a prime number.

$ECC : y^2 = x^3 + ax + b \ (mod\ p)$ where $a, b \in F_p$ and $p > 3$ such that elliptic discriminant satisfies the following equation $4a^3 + 27b^2 \neq 0 \ (mod\ p)$ [52]

There are two main reasons for the above recommendation. Firstly, ECDSA provides the smallest keys and signatures, as compared to other standardized signatures algorithms [10]. Secondly, ECDSA is highly secure because its security is confirmed by the Elliptic Curve Discrete Logarithm Problem (ECDLP). The ECDLP problem is stated by [11] as:

ECDLP: Given the above equation ECC defined over finite field F_p, given a base point $Q \in ECC(F_p)$ of order n, determine an integer l where $0 \leq l \leq n-1$ such that $P = lQ$

Finding the discrete logarithm of a random elliptic curve, whose order is divisible by a large prime n, with respect to a publicly known base point Q is infeasible.

ECDSA generation and verification algorithms comprise of DSA using elliptic curve domain parameters. ECDSA is a Public Key infrastructure (PKI) scheme. The key pair formed is a private key, d, and a public key, Q. Elliptic curve, E, is defined for a finite arithmetic field F_p, such that p is required to be an odd prime. With that, a base point B is selected on the curve, $B \in E(F_p)$ of order n, where n is a large prime.
ECDSA signature generation process consists of the following stages:

1. Per-message secret number generation, k in the interval $[1, n-1]$
2. Hash of message, $h(m)$, where l are the left most bits of $h(m)$
3. Calculation of the curve point $(x, y) = k * B$
4. Formation of signature (r, s) as: $r = x \ mod \ n$ and $s = k^{-1}(l + r * d)$
5. If $s = 0$, goto Step 1

ECDSA signature verification process consists of the following stages:

1. Verify r and s are integers in the interval $[1, n-1]$
2. Compute $w = s^{-1} mod \ n$
3. Compute $u_1 = l * w \ mod \ n$ and $u_2 = r * w \ mod \ n$
4. Compute $u_1 * B + u_2 * Q = (x, y)$ and $v = x \ mod \ n$
5. Signature is verified successfully if $v = r$

ECDSA has computational complexity based on its embedded functions, scalar multiplication, modular multiplication, modular addition, modular inversion and hash. According to the analysis performed by [53], Montgomery's algorithm for costly operations modular multiplication and modular inversion have a complexity of $O(n)$, where n is the number of bits of the operands. Hash function SHA-256 is used for creating a hash of the message. It's

complexity is $O(M \times n)$ where M is the size of the message to be hashed. Therefore, the total complexity of ECDSA is $O(n) + O(M \times n)$.

4.2 IEEE Std. 1609.2 VANET Broadcast Authentication: Simulations

We simulate a broadcast scenario in ns2, using mobility traces from *SUMO* and *Nakagami* radio propagation models. Each simulation run in ns2 completes in 250 seconds.

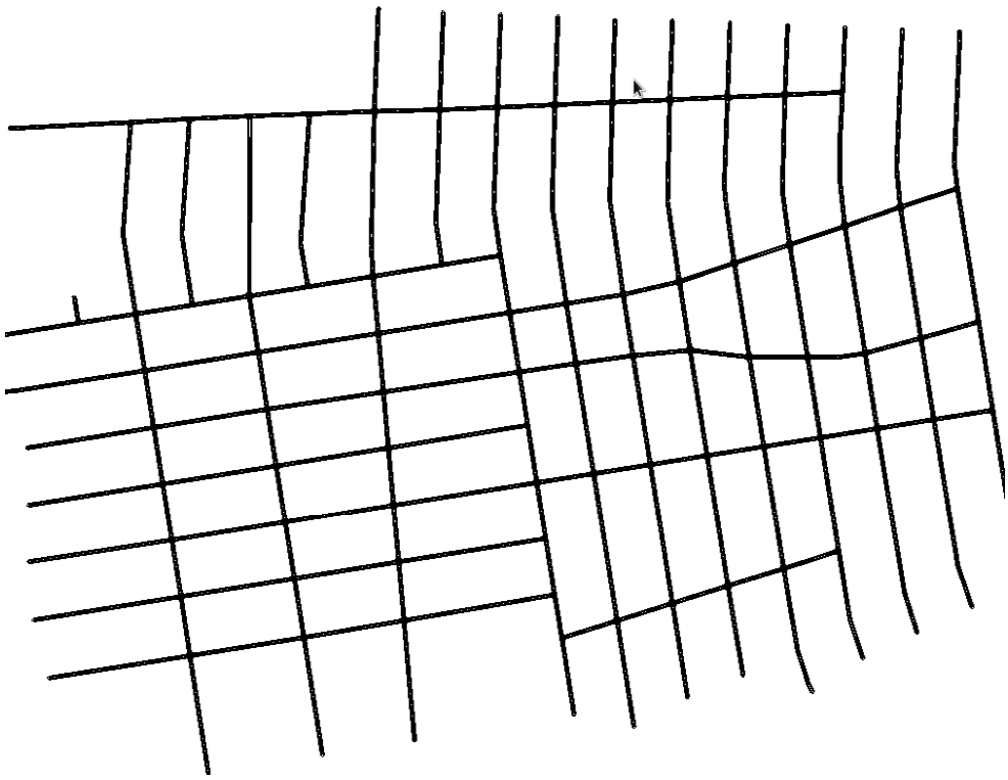

Figure 4.1: Network Layout Representation

For simulating an urban environment, a real city map of New York of area 1378 m X 1019 m is considered. The map has been taken from Open Street Map [54] and processed through Java Open Street Map editor (JOSM) [55] to create a format compatible for SUMO mobility model. The network layout in *SUMO* is shown in Figure 4.1.

In this network, traffic is moving in both directions. Table 4.1 describes the attributes of vehicles that we generated from SUMO for our simulations. In addition, intersection

management with stochastic turns, Krauss car following model and traffic lights have been implemented. The duration of red lights is taken to be 30 seconds on an average.

Table 4.1: SUMO Vehicle Attributes

Vehicle Attribute	Value
Vehicle Length	5 m
Minimum Gap between Vehicles	2.5 m
Max Speed of Vehicles	20 m/s
Acceleration Ability of Vehicles	0.8 m/s^2
Deceleration Ability of Vehicles	4.5 m/s^2
Driver imperfection (0-1)	0.5

The values of Nakagami parameters used for simulating an urban setting are taken from the Vehicle Safety Communications (VSC) project tasks of the [56] as given in Table 4.2. For the Ω function, the gamma values represent the radio signal average attenuation over distance and d0_gamma is the distance at which *gamma* values discontinue. For the function m, m values define the radio signal fading, and d0_m is the distance where gamma value discontinues.

Table 4.2: Nakagami Parameters for Urban Model

Nakagami Parameters	Urban Model Values
gamma0	1.5
gamma1	4
d0_gamma	200
m0	0.75
m1	0.5
d0_m	200

Vehicle on-board units broadcast periodic safety messages every $100 - 300$ milliseconds [9]. These messages contain the vehicle's position, speed and direction of heading. With

IEEE 1609.2, each broadcast is authenticated using ECDSA signatures. On a 400 Mhz processor, ECDSA signature generation takes 4 ms whereas ECDSA signature verification takes 22 ms [12]. In addition, broadcasted messages have an upper bound on their expiry of 1000 ms. Signature verification is on a first come first serve basis while the other messages wait in a queue to be verified before being accepted or acted upon. We implement a circular queue which can hold a maximum of 50 messages at a time.

For analysis, we implement three different scenarios with safety messages broadcasted at a rate of 10 messages per second, 10 messages every 2 seconds and 10 messages every 3 seconds, respectively. 50% vehicles are transmitting safety messages to imitate the initial deployment phases where not all vehicles will have wireless capabilities to generate wireless broadcasts. We call this as *Scenario 1*.

Simulating the above scenario with IEEE Std. 1609.2, we obtain the message loss ratio as shown in Figure 4.2. Here we observe that with the increase in frequency of broadcast and number of vehicles, the message loss ratio due to verification delay increases because the messages time out in the verification queue waiting to be verified. We observe that the message loss ratio for the number of vehicles < 50 with 300 ms broadcast interval is negligible and with 200 ms broadcast interval is $< 5\%$. On the other hand, for high frequency broadcasts with 100 ms broadcast interval, message loss ratio has a steep increase for number of vehicles $25 - 100$; it then takes the shape of a slowly growing curve.

With the broadcast interval set as 300 ms, the maximum packet loss ratio is 0.36 when vehicular density is 225 vehicles in the network area shown in Figure 4.1. While for 200 ms broadcast interval, packet loss ratio goes up to 0.53 and approaches 0.72 as the broadcast interval is changed to 100 ms. Hence, we conclude that due to the delay in verification, many of the messages are timed out and are lost.

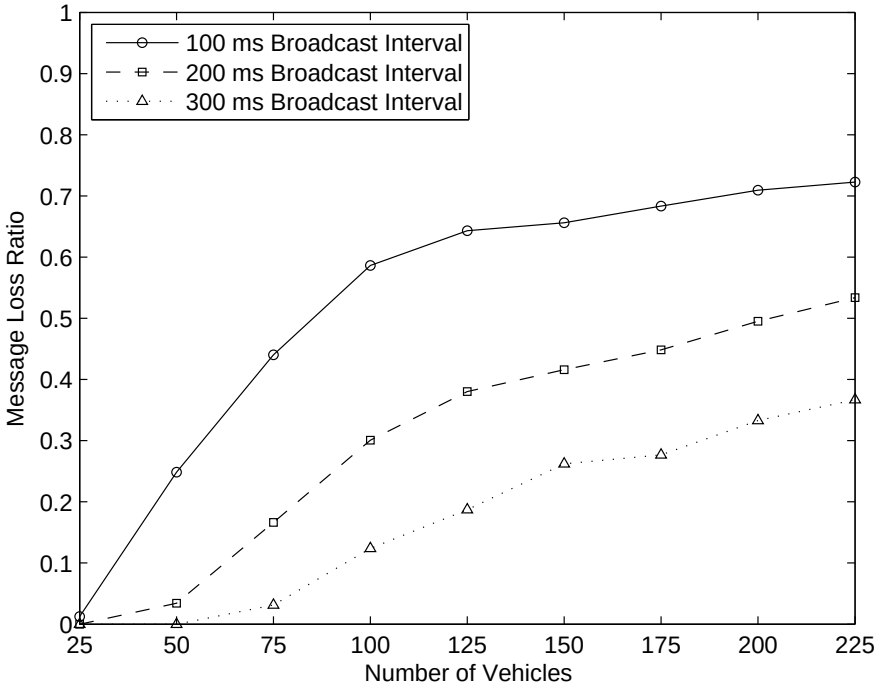

Figure 4.2: Message Loss caused by verification queue delay in IEEE 1609.2

Further, we study the effect of change in number of broadcasting vehicles within the transmission range of a single receiving vehicle. We vary the number of in-range vehicles from 5 to 40. We perform the simulation with broadcast intervals of 100, 200 and 300 ms. We call this *Scenario 2*. For this scenario, we evaluate the percentage of packets processed, i.e., the packets which were taken out from the verification queue before expiration and reached the processing phase or the probability of verification phase. IEEE 1609.2 endorses verification probability to be equal to 1.

Packet processed ratio for *Scenario 2* is given in Figure 4.3. For 100 ms broadcast interval, packet processed ratio decreases when in-range vehicles become > 5 such that it becomes 0.12 for 40 vehicles in-range. While for 200 ms interval, the ratio decreases after in-range vehicles become > 10. For $10 - 40$ in-range vehicles, the packet processed ratio for 200 ms interval is double the value experienced at 100 ms interval. As we increase the broadcast interval further to 300 ms, the ratio rises for all in-range vehicles ending at 0.36

for 40 in-range vehicles. The packets which do time out before reaching the processing stage, are dropped. The reason is again verification delay.

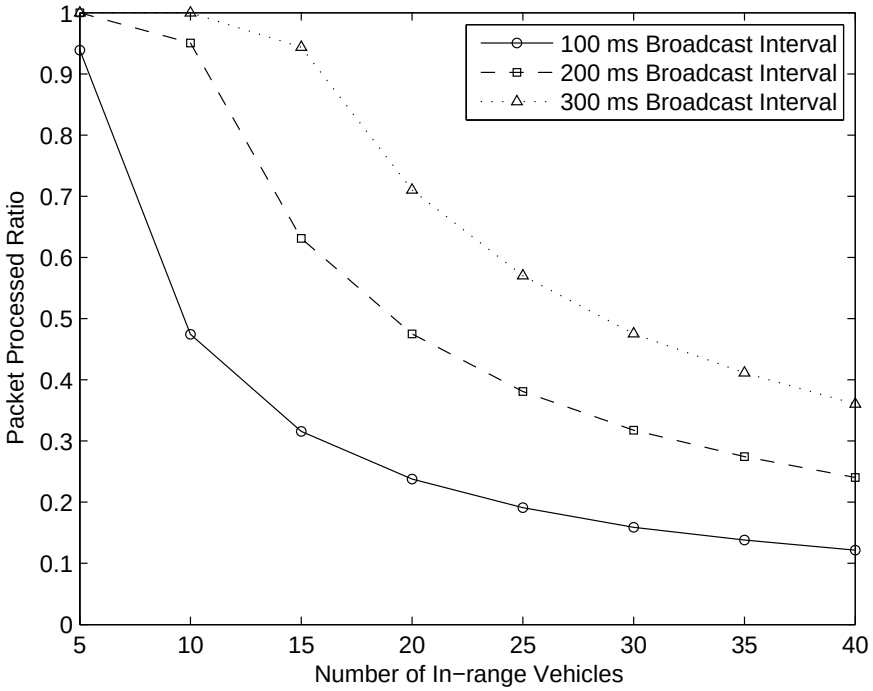

Figure 4.3: Packet Processed Ratio vs. OBUs in range (IEEE 1609.2)

4.3 Problem Statement

As observed in the above simulations, a large number of messages are lost in the queue due to verification delay. Considering the fact that in highly dense environments, the message loss ratio will increase, it is required to design an intelligent authentication strategy. Due to extremely mobile nature of the vehicles, it is essential to transmit the authentication information along with the message. This is because if the authentication information is sent either before or after the message, it might not reach some of the vehicles receiving the message as they travel out of the sender's transmission range.

In addition, privacy of communicating vehicles is essential. In typical digital signature schemes, sender's original identity is appended to it. However, this endangers the privacy

of sender since safety message include vehicle's geographical location. Therefore, we need to establish a mechanism such that the vehicles are not required to append their original identities to the message. At the same time, it is required to have conditional privacy, so that an authority can retrieve the original identity of a vehicle in the cases when malicious activities are detected.

Chapter 5

Related Work

In this section, we summarize recent work on authentication and privacy of vehicular communication, while a condensed model is given in Table 5.1. Security, in vehicular communication, mainly comprises of the authentication because confidentiality is not necessary as all vehicles are required to read the messages. However, action should only be taken on any received message that comes from a legitimate source after it is confirmed by verification.

5.1 Authentication in VANET

A commonly used protocol for broadcast authentication in wireless sensor networks is Timed Efficient Stream Loss-tolerant Authentication (TESLA), generated by [17]. It uses Message Authentication Codes (MAC), along with delayed disclosure of a secret key. TESLA was designed to keep the overhead low for resource-constrained sensor networks. Its main drawbacks are lack of non-repudiation and immediate authentication. Immediate authentication solution [57] can be achieved, but lack of non-repudiation remains unsolved.

On the other hand, vehicular communication requires non-repudiation so that a vehicle cannot deny the transmission of messages in situations of liability such as an accident. Therefore, digital signatures are the preferred choice for broadcast authentication where non-repudiation is a requirement. The most widely used digital signature in broadcast authentication is ECDSA [11]. Nonetheless, computation costs of ECDSA digital signature verification are high.

Signature amortization has been suggested by [58] and [59] to reduce computation and communication overhead by putting a single signature for multiple packets. This creates lack of non-repudiation, compromising the security of messages. Another suggested method

is omitting signatures and certificates with messages sent to a known neighbor, or omitting verification of messages received from known neighbors [60]. Since VANET is a dynamically changing topology, it is not possible to have consistent neighbors for a long period. In addition, this method fails when insider attacks are made. Accelerating signature-based broadcast authentication using inter-nodal cooperation has been implemented by [61]. It requires VANET networks to be implemented over several multi-hops for the scheme to be effective. In a single-hop message broadcast, it acts as all verification scheme.

Another scheme for fast verification of messages is RSU-aided message authentication scheme (RAISE) [62]. Each vehicle needs to associate with an available RSU and obtains a pseudo identity and a shared symmetric secret key. Vehicles make use of symmetric keyed-hash message authentication (HMAC) code for authentication. The messages are buffered on arrival and verified after obtaining the shared symmetric key from the RSU. It requires a very strong RSU backbone and assumes that there is no packet loss between RSU and OBU communication. Further, expensive bilinear operations are used for batch verifications [63] and short signatures [64]. However, the use of bilinear operations is criticized by [65] for the reason that they are suggested in theory and make impractical assumptions. Another technique uses point multiplications for batch verification in emergency communications [66].

5.2 Privacy in VANET

For achieving privacy, it is proposed to preload each vehicle's on-board unit with a set of private and public key pairs along with their public key certificates [15]. Each public key certificate includes a pseudo identity. Each set of key pair and certificate is valid for a particular time period. The number of sets preloaded depends on the storage capacity of OBU. Once all the stored key pairs are exhausted, reloading is required. Location privacy has been used by [67]. They combine ID-based signature [63] and location information of the vehicle to generate pseudo identity. Message verification process has complexity similar to ECDSA verification.

Table 5.1: Related Work

Scheme	Aim	Method
TESLA	Lightweight Message Authentication	MAC with delayed key disclosure
Signature Amortization	Reduce overhead for Message Authentication	One signature for multiple packets
Omitting Signatures to known neighbors	Reduce overhead for Message Authentication	Does not generate/verify signatures and certificates to/from known neighbors
Using inter-nodal cooperation	Reduce computation for Message verification	First tier receivers forward partial results with rebroadcasts
Selective Authentication	Reduce overhead for Message Authentication	Recursive warning messages
Short Signatures using bilinear pairing	Lightweight Message Authentication	Bilinear Pairing
RSU-Aided Authentication	Lightweight Message Authentication, Privacy	keyed-hash message authentication (HMAC)
Preloaded Key Pairs	Privacy	Several Public Private key pairs
ID-based Signatures	Privacy	Original Identity based signature
Group Signatures	Privacy	Group manager generates signatures on behalf of the group members
Batch Verification	Privacy	Multiple signatures are verified in one process
Group Signatures using bilinear pairing	Privacy	Bilinear Pairing
Key Distribution	Privacy, Lightweight Authentication	KJLN key exchange
Group Communication	Privacy, Authentication	Bloom Filter
Lane Departure Warning System	Authentication (Application)	ECDSA

Group signatures are used to solve the privacy problem, such that messages originating from a group of vehicles is signed by the group manager [68]. The identity of the message originating vehicle can only be traced down through the group manager. However, they make use of message authentication codes generated by hashing the components to be signed. In addition, it requires high storage capacity in vehicles to store the neighbors information. Most group signatures [69] and batch verifications [70] are done using bilinear pairing and hence are based on impractical assumptions, according to [65].

Another measure suggested for authentication and privacy is a key distribution scheme designed using Kirchhoff-Law-Johnson-Noise (KLJN), where a roadside key provider is used to distribute keys to the vehicles [71]. Security of this method is determined by information theory, used with KJLN in wireline connection segments only. On the other hand, authors of [72] have opposed the usage of key exchange protocols in VANETs and have proposed a group communication scheme using bloom filters. This scheme will require frequent reformations of the communication groups created by RSUs as vehicles will rapidly travel out of the transmission range of an RSU. There is also a requirement for a strong RSU backbone and only vehicles belonging to a group can send/receive broadcasts from others.

There are many applications that can make use of the V2V security measures, especially authentication. For instance, a Lane Departure Warning System (LDWS) is implemented with Elliptic Curve Integrated Encryption Scheme (ECIES) for the security of V2V communication. They use alarms to indicate the departure of a vehicle from its driving lane, these alarms are authenticated using ECDSA. In [73], vehicle performance monitoring and analysis application has been designed using Rivest Cipher 6 (RC6) cryptographic algorithm combined with compression, such that the data is aggregated to summarize the content and remove duplicates. Though RC6 is a fast and flexible algorithm, however, it requires integer multiplications on rotations and is not universally practical.

The above given literature review addresses many security and privacy issues in VANETs that have been successfully identified. However, before the actual implementation of vehicular communication, we need to find an appealing solution for practical scenarios. Initial deployment phases will not have all vehicles equipped with wireless capabilities, nor will they have a strong RSU backbone fully implemented.

5.3 Shortcomings of Existing methods

As specified in our simulation scenario, the VANET implementation is based on the initial scenarios where i) not all vehicles will have the capabilities of developing a broadcast transmission message, ii) it will not be possible for each vehicle to be connected to an RSU at all times because no strong RSU backbone will be provided throughout, iii) density of vehicles may vary during different hours of the day and hence the connectivity of the vehicles is not guaranteed, iv) all vehicles receiving broadcasts will have to perform independent verification of these messages.

Taking into account the above scenario, we study the existing methods and report their shortcomings. We first discuss the drawbacks of existing VANET authentication schemes followed by the drawbacks of existing VANET privacy schemes in Table 5.2.

TESLA and signature amortization are widely used broadcast authentication techniques, but at the same time lack non-repudiation and immediate authentication. The technique of omitting signatures and certificates for known neighbors requires the maintenance of neighbor list which can have additional overhead and is not suggested in the initial deployment phases of VANET when all vehicles will not have wireless capabilities. On the other hand inter-nodal cooperation and RSU-Aided authentication are not feasible in an environment demanding independent verification. Moreover, inter-nodal cooperation is effective only in a multi-hop environment of broadcast messages; in other cases it acts similar to ECDSA authentication used by the standard IEEE 1609.2. Most of the privacy schemes do not support independent verification. Others, such as ID-based signatures, are prone to message forgery attacks.

Due to these shortcoming, the existing methods are not feasible for implementation in the given scenario. Hence, emerges a requirement for a new technique to be proposed.

Table 5.2: Shortcomings of Existing VANET Authentication and Privacy Schemes

Existing VANET Authentication Scheme	Drawback
TESLA	Non-repudiation, immediate authentication, requires time synchronization
Signature Amortization	Non-repudiation, immediate authentication
Omitting Signatures to known neighbors	Insider attacks, neighbor-list maintenance
Using inter-nodal cooperation	Independent verification, requires strong connectivity between vehicles
Selective Authentication	Not suitable with Privacy alternatives
RSU-Aided Authentication	Requires a strong RSU backbone, increases communication overhead
Short Signatures using bilinear pairing	Theoretical schemes with impractical assumptions [65]
Lane Departure Warning System	Long verification delay
Existing VANET Privacy Scheme	**Drawback**
Preloaded Key Pairs	Storage, reloading
ID-based Signatures	Message forgery attack
Group Signatures	Independent Verification
Batch Verification	Independent Verification
Group Signatures using bilinear pairing	Theoretical schemes with impractical assumptions [65]
Key Distribution	Only used in wireline communication
Group Communication	Frequent group reformations

Chapter 6

Proposed Solution

6.1 Conceptual Overview

As seen in the implementation of IEEE 1609.2, a large number of packets are dropped due to verification delay. It is certain that this scheme cannot be used in practice without modification. We also agree that ECDSA is a very secure algorithm, it provides source authentication, message integrity and non-repudiation. However, it has the drawback of signature verification overhead. Moreover, we have seen that all the existing solutions for VANET broadcast privacy and authentication have shortcomings. Therefore, we propose a security framework for vehicular communication including an effective probabilistic ECDSA authentication strategy and conditional privacy [74].

The proposed scheme provides three-fold features:

1. Conditional privacy: We make available a privacy controlled mechanism, such that the vehicles do not pass their original identities during inter-vehicular communication. Hence, providing security against identity thefts, message forging and bogus information. However, Regional Authorities (RA) keep a record of the original vehicle identities and are responsible for disclosing them when desired by higher authorities.

2. Independent Authentication: Our proposed scheme suggest the use of elliptic curve digital signature algorithm, such that the authentication information (ECDSA signature) is sent along with the message. Both at the sender and receiver end, generation and verification of authentication information is done independently, i.e, a it does not require the aid of any other message, or any other neighbor or entity.

3. Probabilistic Verification: Our scheme assigns a probability of verification to each received broadcast, based on the practical relevance of message determined by distance and direction of the communicating vehicles with respect to each other. Message with higher relevance, such as messages received from nodes coming closer are assigned higher probability of verification, as compared to messages from nodes moving away.

We divide our solution into four sequential phases, where the first two phases account for V2I communication for establishing conditional privacy in the third phase. The fourth phase accounts for the V2V communication. The four phases are as follows:

1. System Initialization: It involves the initialization of registering authority (RA) and vehicle's OBU through the Trusted Authority (TA), the only trusted entity in the network.

2. Vehicle Registration: Each vehicle is registered and authenticated by the RA to become qualified to communicate in a vehicular ad hoc network. Since we do not assume the RA to be a trusted entity, each vehicle authenticates the RA as well.

3. Pseudonym Generation: To maintain privacy, the vehicles do not use their unique identifiers in communications, instead pseudonyms are used. The achieved privacy is conditional and the RA is able to retrieve the original identity of each communicating vehicle.

4. Message Authentication: Source validation and data integrity are accomplished by assigning digital signatures (ECDSA) to messages. It includes two processes: signature generation and mobility based probabilistic signature verification. The aim of this scheme is to reduce the number of packet losses occurring due to delay in verification queue and maximizing the number of relevant verifications. The proposed scheme does not incur any additional communication overhead because it uses the data included in the broadcasts.

6.2 Detailed Solution

Below we give a detailed description of our four-phase security framework for VANET communication such that it provides conditional privacy to the communicating vehicles.

Table 6.1 represents frequently used parameters.

Table 6.1: Table of Notations

Notation	Description
TA	Trusted Authority
RA	Regional Authority
OBU^i	Wireless On-Board Unit of i^{th} vehicle
ID^i_{OBU}	Unique identifier of i^{th} OBU
AID^i_{OBU}	Alias of i^{th} OBU for registration
ID_{TA}	Unique Identifier of TA
R^i_{OBU}	Random Number of i^{th} OBU
R^i_{RA}	Random Number of RA for i^{th} OBU
SK	secret encryption key of TA
$h()$	hash function
$\|\|$	concatenation operator
θ	Direction of motion of vehicle
RK	Encryption key given to OBUs by RA
x, y	location coordinates of OBU
V	Velocity of OBU
$P^{i,j}_{OBU}$	Pseudonym of i^{th} OBU for j^{th} message
$ts\_$	timestamp

6.2.1 System Initialization

The first process is the initialization of the entities forming the vehicular network. The entities are RA and each vehicle's OBU. System initialization is performed by the TA, it issues the system parameters to RA, hash function, *h()*, and TA's secret encryption key, SK. While each OBU is initialized with a unique secret and other parameters for communication

with the RA.

OBU^i communicates with the TA by providing it's original unique identifier (ID^i_{OBU}).

For each OBU^i, TA generates

- *a random key,* x^i
- *registration parameters*

$$\begin{cases} p^i = h(x^i) \\ seal^i = E_{SK}(ID_{TA} \,||\, p^i) \\ secret^i = h(ID^i_{OBU} \,||\, p^i) \end{cases}$$

On the whole, TA provides parameters ($h()$, p^i, $seal^i$, $secret^i$) along with the ECDSA public-private key pair and a public key certificate to each OBU^i.

6.2.2 Vehicle Registration and Mutual Authentication

In this phase, each vehicle's OBU registers with the RA. Moreover, it involves a mutual authentication process in which each OBU and RA validate each other. This design requires the vehicles joining different VANETs to go through the vehicle registration process again. The purpose of this phase is to allow vehicles to use pseudonyms instead of their original identifiers when communicating with other vehicles on the road, maintaining privacy.

The privacy is, however, conditional because the RA will have the information about the vehicles in their area and in case of accident or any other mishandling of information, the RA can reveal the original identity of the vehicle doing so. Inspired by the work [75], we use symmetric cryptography for registering vehicles with the RA. The complete process is shown in Figure 6.1. Further, we elaborate on each step that takes place in the vehicle registration and mutual authentication.

Step 1: Firstly, OBU^i selects a random number R^i_{OBU}, and concatenates it with p^i (given to it by the TA). Then, the concatenated value is hashed and XORed with the original ID of the OBU to obtain an alias.

$$AID^i_{OBU} = h(p^i \mid\mid R^i_{OBU}) \oplus ID^i_{OBU}$$

Secondly, the OBU hashes the concatenation of the random number and $secret^i$ to attain the value of $code^i$

$$code^i = h(secret^i \mid\mid R^i_{OBU})$$

Finally, it creates a stamp by encrypting the concatenation of random number and $code^i$, where secret parameter p^i is used as the encryption key.

$$stamp^i = E_{p^i}(code^i \mid\mid R^i_{OBU})$$

Step 2: Vehicle transmits the following to RA for authentication and registration: alias, $stamp^i$ and $seal^i$

$$\{AID^i_{OBU}, stamp^i, seal^i\}$$

Step 3: After RA receives the authentication message (along with authentication values) from a vehicle, firstly, it decrypts the values of ID_{TA} and p^i from $seal^i$, using the TA's secret encryption key, SK.

Further, RA retrieves the original ID of OBU by XORing the AID^i_{OBU} with the hash of p^i and R^i_{OBU}.

$$ID^i_{OBU} = AID^i_{OBU} \oplus h(p^i \mid\mid R^i_{OBU})$$

Next, RA matches the ID^i_{OBU} with the Certificate Revocation List (CRL). If a match is found, it discards further requests processing for ID^i_{OBU}.

Otherwise, since all the individual components leading to $secret^i$ are recovered. RA computes

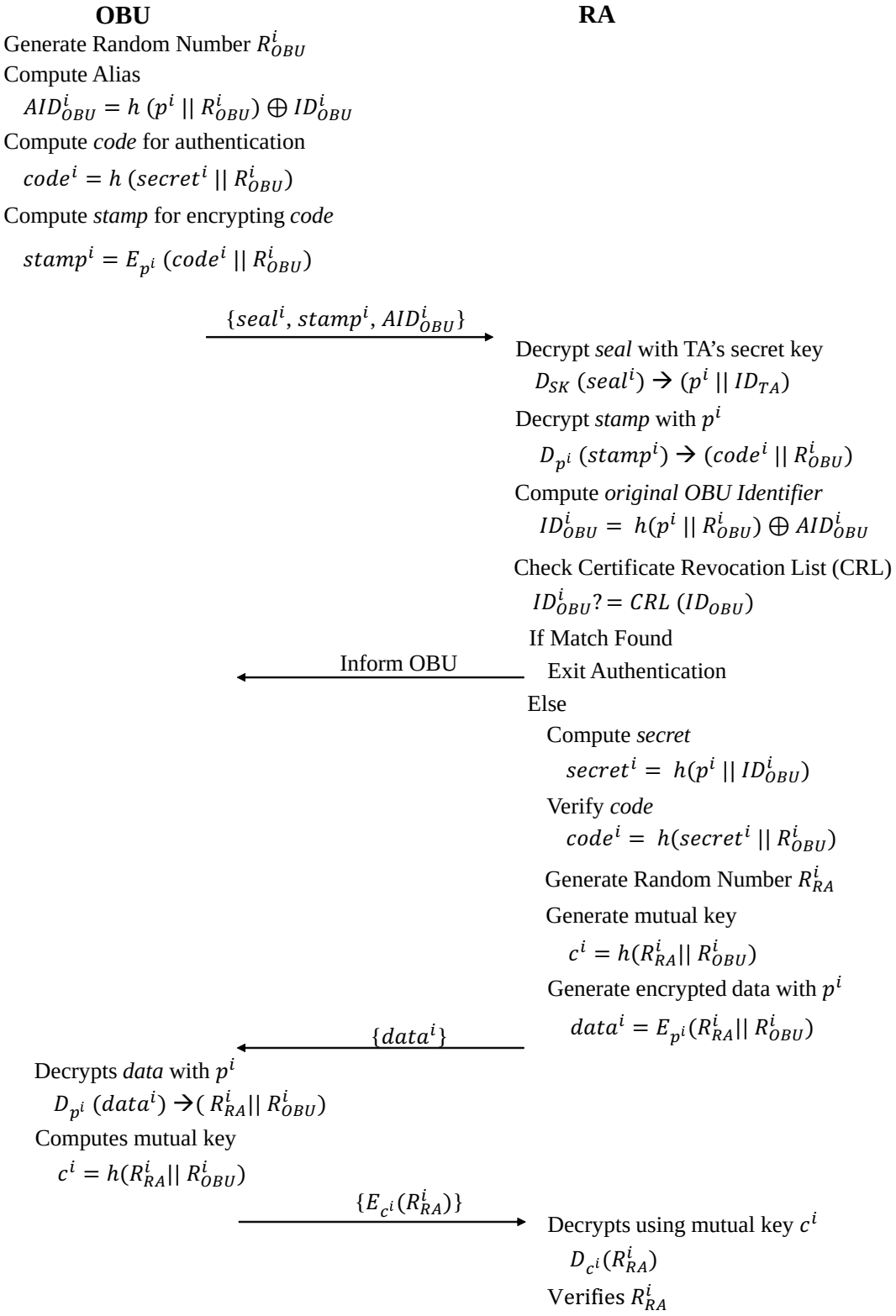

Figure 6.1: Vehicle Registration and Mutual Authentication between RA and Vehicle's OBU

the $secret^i$ using ID^i_{OBU} and R^i_{OBU}; and lastly verifies the value of $code^i$ accomplishing vehicle registration process.

RA stores each vehicle's corresponding $code^i$ along with their original identifiers ID^i_{OBU}.

For mutual authentication process, RA generates a random number R^i_{RA} for OBU^i, and a mutual key, c^i

$$c^i = h(R^i_{RA} \mid\mid R^i_{OBU})$$

Step 4: RA transmits the concatenated random numbers encrypted with secret parameter p^i, to the vehicle's OBU.

$$\{E_{p^i}(R^i_{RA}||R^i_{OBU})\}$$

Step 5: The vehicle's OBU decrypts the received value to discover R^i_{RA} and works out the value of mutual key c^i.

$$Calculates\ (ck^i)$$

Step 6: The vehicle then sends RA's random number back to it, encrypted over the key, ck^i

$$\{E_{ck^i}(R^i_{RA})\}$$

Step 7: RA solves the encryption and verifies the random number

$$Verifies\ (R^i_{RA})$$

Successful verification completes the vehicle registration and mutual authentication process.

6.2.3 Pseudonym Generation

Each OBU then creates a pseudonym, P^i_{OBU}, at its own end for communicating with the other OBUs in the network using the values $code^i$ and ID^i_{OBU}, known to RA.

$$P^{i,j}_{OBU} = code^i \oplus ID^i_{OBU}$$

This allows the RA to have control over obtaining the original identity of the OBU. TA the same time, no third party can track or access the broadcasting OBUs real identity.

6.2.4 Message Authentication

In this process, we use Elliptic Curve Digital Signature Algorithm (ECDSA) for authentication of messages. The authentication process comprises of signature generation at each sender end and signature verification at all the receivers.

Signature Generation

According to the DSRC standard, each periodic BSM broadcast transmission includes the physical location coordinates of the vehicle, on which the OBU is installed, at the time of generation of the message, velocity of the vehicle on which the OBU is installed, and warning information. We append the OBU's pseudonym and timestamp, $ts\_$ (the time at which message was generated).

$$message = \{x_i, y_i, V_i, info, P^i_{OBU}, ts\_\}$$

Creation of the message is followed by the signature generation. This message is then signed using the sender's ECDSA private key. We use ECDSA signature generation procedure as described in Section 4.1. Each signature generation take 4 ms on a 400 Mhz processor [12]. The public key certificates accompany the message and signature in the broadcast packet.

Signature Verification

Signature verification is done using a probabilistic strategy, Mobility-Based Probabilistic Verification (MBPV). We carefully consider the method for computing the probability of verification. The factors used to estimate probability are whether the communicating vehicles are coming close or moving farther away from each other, and relative direction of motion

of the communicating vehicles. Mathematical computation of probability of verification is based on i) difference between the present and future distances of sender and receiver, and i) relative direction of motion between the sender and receiver.

For predicting the direction of motion of vehicles, we develop a mobility prediction model based on past track history. We estimate the future position of the vehicle, and computes the relative direction of motion. Further, the difference in distance between the vehicles at present and in future determines if the vehicles are coming closer or moving away. Moreover, the relative direction of motion between sender and receiver is computed using the coordinates contained in the BSM broadcast. Hence, we carry no additional information with the broadcast for our computations.

Below, we describe our procedure in four sequential steps.

Step 1: Mobility Prediction

Mobility prediction can be done by various methods, such as tracking movement history, physical topology and logical topology [76]. Here, we define two methods for computing a vehicle's direction or mobility. The first method uses past track history. We accomplish this task by logging the physical coordinates of each vehicle in its database after a regular interval. The interval found to be most appropriate is 500 milliseconds.

The direction of motion can be predicted using the past track history, i.e., from the previous location coordinates and present location coordinates. For this method, it is required that each vehicle has a location management module, which keeps a track of its previous locations by logging its position information. Using the history of past locations, we compute the direction of motion as follows:

$$\theta_{t(i)} = tan^{-1}(\frac{y_{t(i)} - y_{t(i-1)}}{x_{t(i)} - x_{t(i-1)}})$$

$\theta_{t(i)}$ is the direction of motion of vehicle, $x_{t(i)}, y_{t(i)}$ are the position coordinates of vehicle at time t_i, and $x_{t(i-1)}, y_{t(i-1)}$ are the position coordinates of vehicle at time t_{i-1}. Initially

the vehicles do not have a past track history and can use their velocity information for computation.

However, at the beginning, vehicles do not have a past track history and cannot use the above method for computation of direction of motion. Therefore, we suggest the use of a second technique using the velocity components. *Given* $V = Velocity\ of\ Vehicle$

It is known that

$$\begin{cases} V_x = V cos\ \theta \\ V_y = V sin\ \theta \end{cases}$$

Therefore, we obtain:

$$\theta_{t(i)} = tan^{-1}(\frac{V_y}{V_x})$$

This method is of temporary usage for the initial stages. This is because the vehicle may experience various stops in the later stages such as red lights or due to the car following model. Once, the vehicle starts logging, it can effectively use its past track history.

Step 2: Estimation of future coordinates

The second step, following mobility prediction is the estimation of future coordinates. Once the direction of motion, θ, of the vehicle is obtained, from the above step, we estimate the future coordinates using the following equations:

$$x_{t(i+1)} = x_{t(i)} + V \times cos\ \theta \times \Delta t$$

$$y_{t(i+1)} = y_{t(i)} + V \times sin\ \theta \times \Delta t$$

where $x_{t(i)}, y_{t(i)}$ are coordinates of the vehicle at time t_i, and $x_{t(i+1)}, y_{t(i+1)}$ are physical location coordinates of the vehicle at time t_{i+1}. Time interval after which the future coordinates are estimated is given by Δt, such that $\Delta t = t_{i+1} - t_i$. For our simulations, we take Δt as 1000 milliseconds.

Step 3: Distance Computation

We estimate if the vehicles (receiver and sender) are moving away or close to each other. To

do so, we compute the distance between the receiver and sender at present and in future, using the present and future coordinates of the vehicles.

$$Distance_{t(i)} = \sqrt{(x^r_{t(i)} - x^s_{t(i)})^2 + (y^r_{t(i)} - y^s_{t(i)})^2}$$

$x^r_{t(i)}, y^r_{t(i)}$ are the position coordinates of the *receiver* at time t_i, and $x^s_{t(i)}, y^s_{t(i)}$ are the position coordinates of the *sender* at time t_i.

$$Distance_{t(i+1)} = \sqrt{(x^r_{t(i+1)} - x^s_{t(i+1)})^2 + (y^r_{t(i+1)} - y^s_{t(i+1)})^2}$$

$x^r_{t(i+1)}, y^r_{t(i+1)}$ are the position coordinates of the *receiver* at time t_{i+1}, and $x^s_{t(i+1)}, y^s_{t(i+1)}$ are the position coordinates of the *sender* at time t_{i+1}.

The quantity that is important for our calculations is the difference between these two given distances.

Step 4: Probabilistic Verification

Our probabilistic verification, as given in Algorithm 2, is dependent on the difference between the present distance and estimated future distance of sender and receiver such that the relative direction of motion between the sender and receiver is a scaling factor for the value of probability.

Algorithm 2 Probabilistic Verification

if $Distance_{t(i+1)} - Distance_{t(i)} < 0$ **then** /*Vehicles are coming close*/
  if $f_1(\theta) > 0$ **then**
    $P_v = max(f_1(\theta).f_2(d), f_2(d))$
      ▷ where $f_1(\theta) = \frac{|\theta^r_{t(i)} - \theta^s_{t(i)}|}{180}$ and $f_2(d) = 1 - e^{(Distance_{t(i+1)} - Distance_{t(i)})}$
  end if
end if
if $Distance_{t(i+1)} - Distance_{t(i)} > 0$ **then** /*Vehicles are moving away*/
  if $f_1^{'}(\theta) > 0$ **then**
    $P_v = min(f_1^{'}(\theta).f_2^{'}(d), f_2^{'}(d))$
      ▷ where $f_1^{'}(\theta) = 1 - \frac{|\theta^r_{t(i)} - \theta^s_{t(i)}|}{180}$ and $f_2^{'}(d) = e^{-(Distance_{t(i+1)} - Distance_{t(i)})}$
  end if
end if

We use continuous exponential distribution for computing the probability of verification. This is because exponential distribution has the property of generating results independent of previous occurrences. The complete set of equations is given in Algorithm 1. Probability of verification is denoted by P_v, function of distance is denoted by $f_2(d)$ and $f_2^{'}(d)$ and function of direction of motion is denoted by $f_1(\theta)$ and $f_1^{'}(\theta)$.

Function of distance is computed using exponential distribution while the function of direction of motion is a ratio. The decision of formula selection of probability computation is primarily based on the difference in distance between the communicating vehicles. Besides, the relative direction of motion between the communicating vehicles is used as a scaling factor.

If the distance between the communicating vehicles is decreasing, P_v is the *maximum* of $f_2(d)$ and product of $f_2(d)$ with $f_1(\theta)$, only if function of direction of motion has a positive value. On the other hand, if the distance between the communicating vehicles is increasing, P_v is the *minimum* of $f_2^{'}(d)$ and product of $f_2^{'}(d)$ with $f_1^{'}(\theta)$, only if function of direction of motion has a positive value.

Using the above algorithm for probabilistic verification of messages, we aim to achieve higher probability of verification for messages which are practically more relevant, i.e., coming from vehicles which are growing closer in distance and moving in relatively approaching directions.

Chapter 7

Results and Discussion

In this chapter, we compare the simulation results of our scheme, MBPV, with the above discussed IEEE 1609.2 and two widely used algorithms for broadcast authentication in ad hoc networks, Signature Amortization and TESLA. We discuss simulation setting and assumptions, followed by a discussion on the results and security analysis.

7.1 Simulations Settings

In this section, we analyze four algorithms, IEEE Std. 1609.2, Signature Amortization (SigAmor), TESLA (mTESLA) and our solution, MBPV. We implement the standard IEEE 1609.2 as discussed above in Chapter 4; Signature Amortization is implemented by putting a single signature for every *five* packets; TESLA is implemented such that the key of previous message is piggy-backed with the next message, this is done to preserve the communication bandwidth; hence we use the name mTESLA (modified TESLA). We consider the following assumptions for this work: i) all vehicles are deployed in a 2D area, ii) all vehicles are equipped with GPS, and iii) each vehicle has either its previous position or is aware of its present velocity, i.e., speed and direction vector. The simulations are done in similar settings as in Section 4.2 to maintain consistency in the network settings. Our results are an average of five simulation runs.

7.2 Results

Figure 7.1, 7.2 and 7.3 represents the comparison of message loss ratio of IEEE 1609.2, SigAmor, mTESLA and our scheme MBPV, as the broadcast interval of messages is varied. These results are from *Scenario 1.* In all figures, we indicate the 95% confidence level of each

data point in our scheme. We observe that SigAmor and mTESLA will always have some message loss, even if the vehicle density is sparse. The reason is that at high mobility of the vehicles, the receivers are unable to receive the authentication packets sent at a later point in time. However, IEEE 1609.2 and MBPV have negligible message loss for sparse networks, depending on the frequency at which messages are broadcasted.

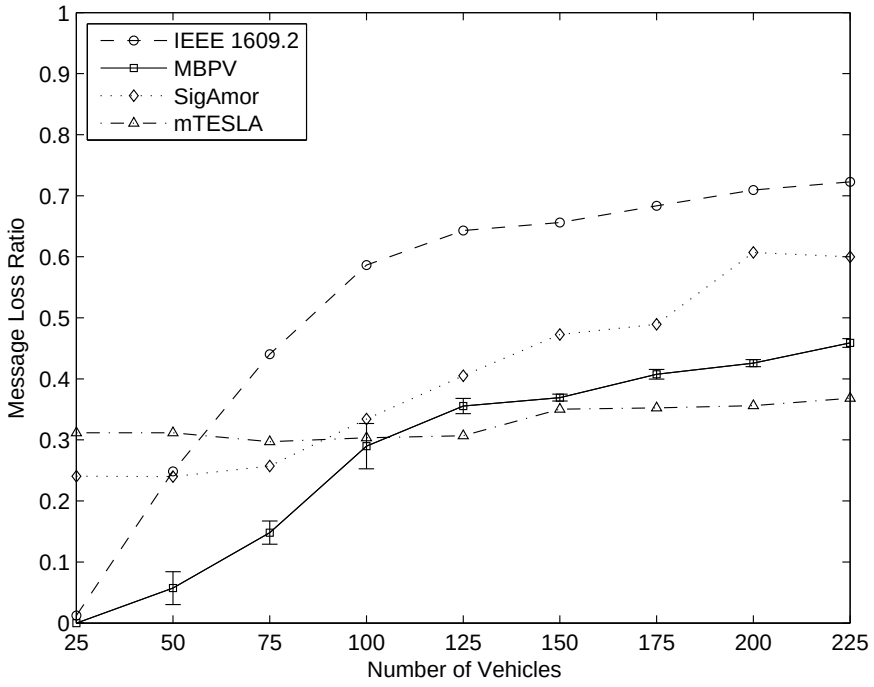

Figure 7.1: Message Loss caused by verification queue delay for 100ms broadcast interval.

When the broadcast interval for messages is set to 100 ms, IEEE 1609.2 is seen to have a very high message loss ratio. SigAmor looses between $25 - 60\%$ messages, while mTESLA looses about one-third of its messages in verification queue. MBPV proves to be better than all till the vehicle density is below 100, above which mTESLA wins over it. However, we would like to re-enumerate the drawbacks of mTESLA, which are non-repudiation, immediate authentication and time-synchronization, which make it impractical to be used in VANET applications involving human-life.

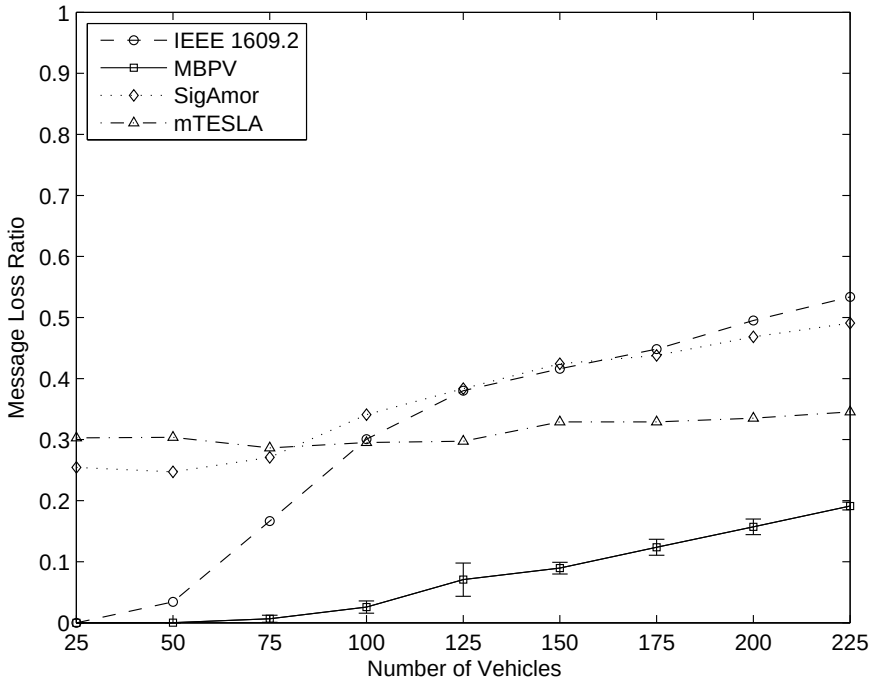

Figure 7.2: Message Loss caused by verification queue delay for 200ms broadcast interval.

For 200 ms and 300 ms broadcast interval, the following observations are made. IEEE 1609.2 has almost negligible message loss ratio when number of vehicles is < 50 for 200 ms broadcast interval and when number of vehicles < 75 for 300 ms broadcast interval. SigAmor losses between $25 - 50\%$ messages with 200 ms broadcast interval and $38 - 53\%$ with 300 ms broadcast interval. In both the cases, mTESLA looses about one-third of its messages in the verification queue, similar to the above case.

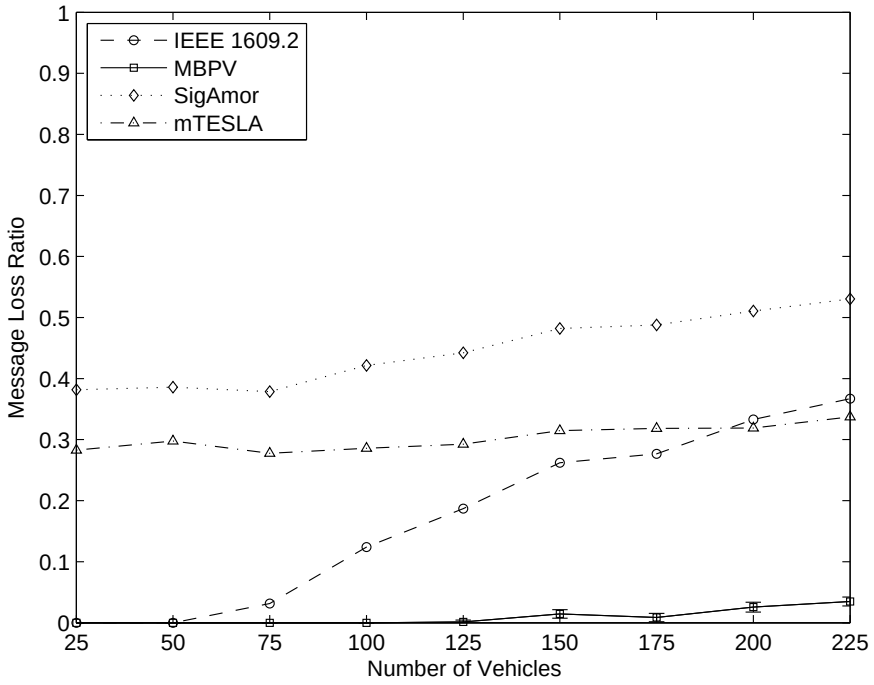

Figure 7.3: Message Loss caused by verification queue delay for 300ms broadcast interval.

MBPV proves to be the best algorithm in both cases, 200 ms and 300 ms broadcast interval. MBPV shows an average improvement (considering all broadcast interval cases) of 68% over IEEE 1609.2, 73% over SigAmor and 64% over mTESLA; with the average minimum improvement (considering one broadcast interval case at a time) being 55%, 37% and 17% against IEEE 1609.2, SigAmor and mTESLA, respectively.

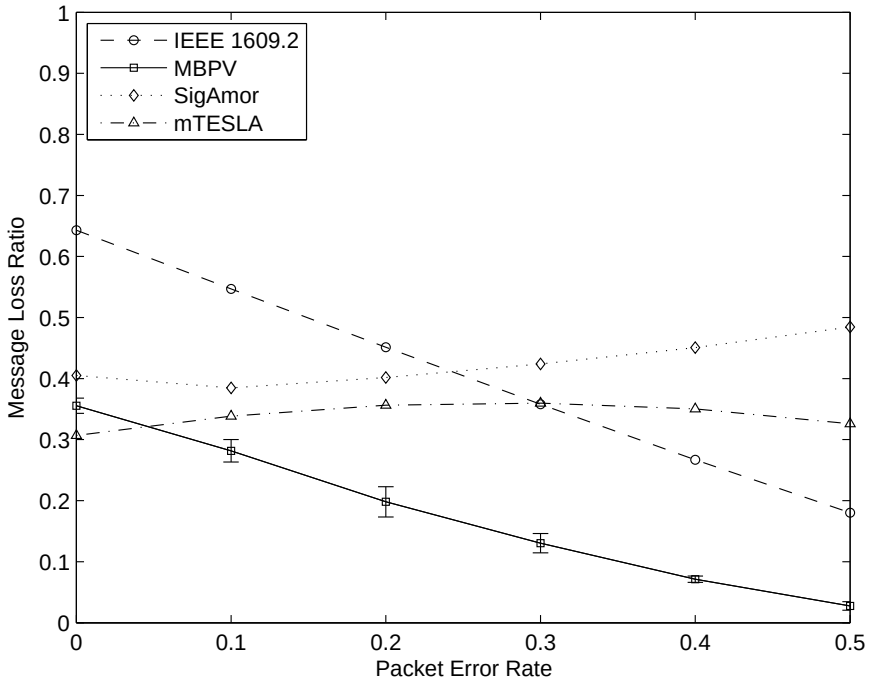

Figure 7.4: Effect of Packet Error Rate on Message Loss for 100ms broadcast interval.

Further, we introduce packet error rate in the network to inspect its effect on the message loss ratio. The total number of vehicles considered in this scenario is 125. Here, we study how the packet error rate of the network affects message loss caused by the verification queue delay, as the message broadcast interval is varied. Figure 7.4, 7.5 and 7.6 demonstrate the results for broadcast intervals 100, 200 and 300 ms, respectively.

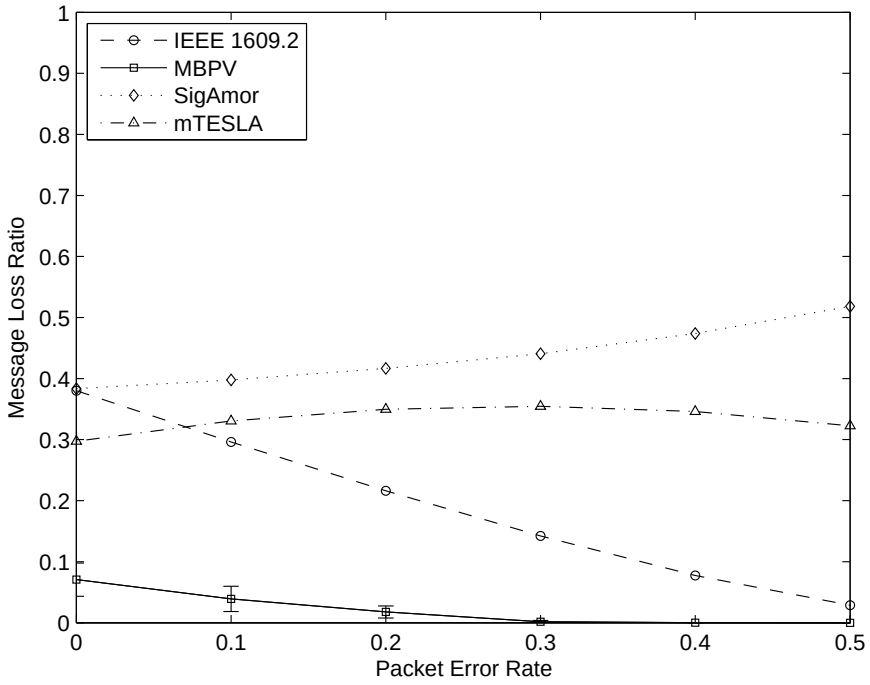

Figure 7.5: Effect of Packet Error Rate on Message Loss for 200ms broadcast interval.

Since more number of packets are dropped with an increase in packet error rate, IEEE 1609.2 gets to verify fewer messages hence increasing the verification rate with the increase in packet error rate. On the other hand, SigAmor has dependent verification, therefore, its verification rate decreases with increase in packet error rate. mTESLA maintains its loss ratio of one-third packets. MBPV behaves like IEEE 1609.2, since it does independent verification of messages, and proves to be the best algorithm.

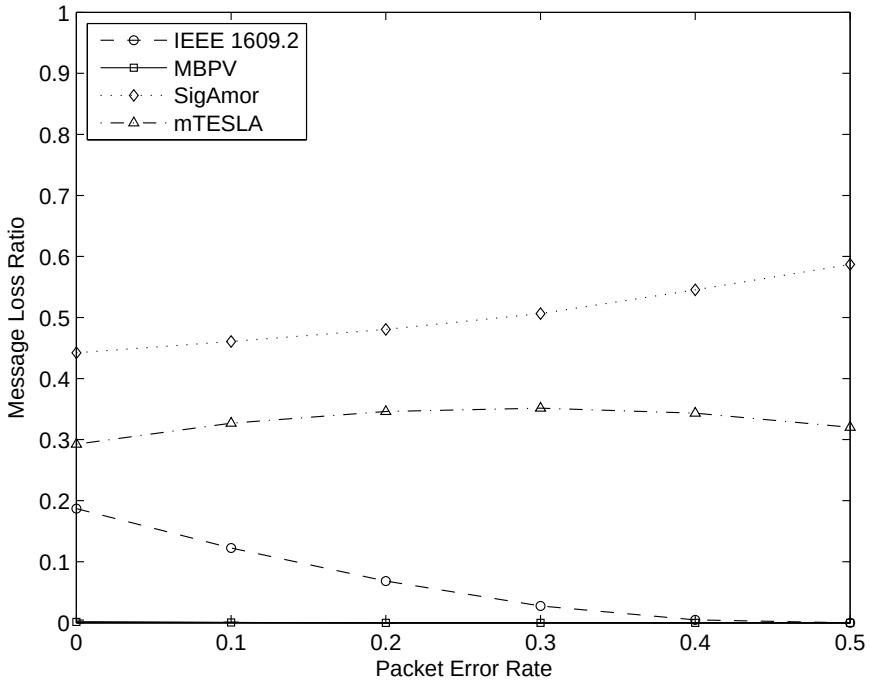

Figure 7.6: Effect of Packet Error Rate on Message Loss for 300ms broadcast interval.

Next, we use *Scenario 2* for the computation of packet processed ratio as the number of vehicles in the transmission range of a single vehicle is increased. All the in-range vehicles in this scenario are taken to be in communication range of the receiver. Figures 7.7, 7.8 and 7.9 displays our results.

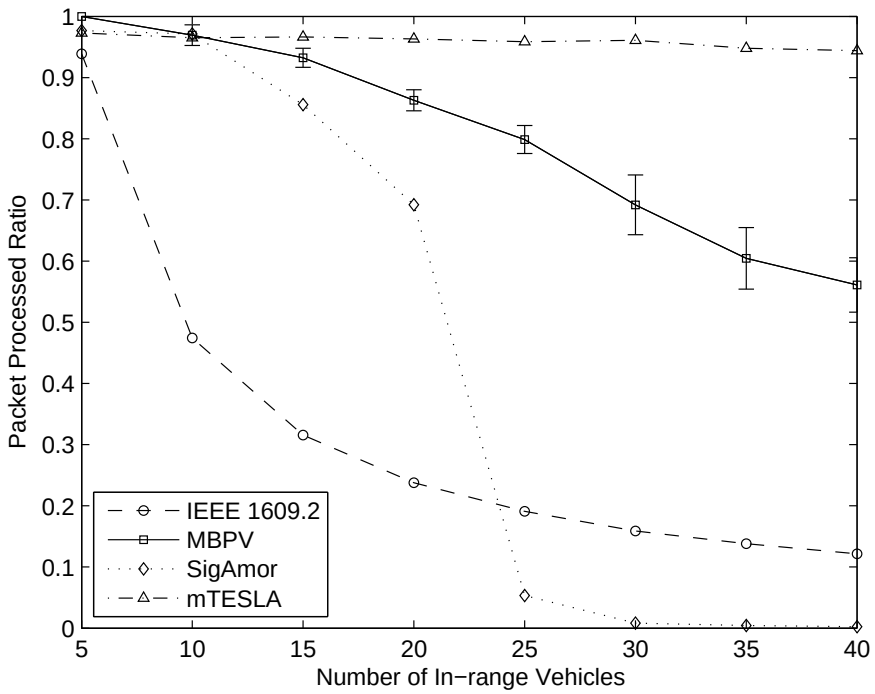

Figure 7.7: Packet Processed Rate vs Number of In-range Vehicles for 100ms broadcast interval.

With 100 ms broadcast interval, we observe a rapid decline in the packet processed ratio for IEEE 1609.2 when the in-range vehicles is between $5 - 20$. It then decreases slowly. While SigAmor shows a steep curve when the number of vehicles increases from 20 to 25, and becomes negligible further on. mTESLA shows a very high packet processed ratio, since all the vehicles are in communication range, therefore, the receiver is always able to obtain the authentication packet which arrives after the original message. In addition mTESLA has negligible computation overhead. MBPV shows a slow decline as the number of in-range vehicles is increased because it uses computationally expensive ECDSA for verification.

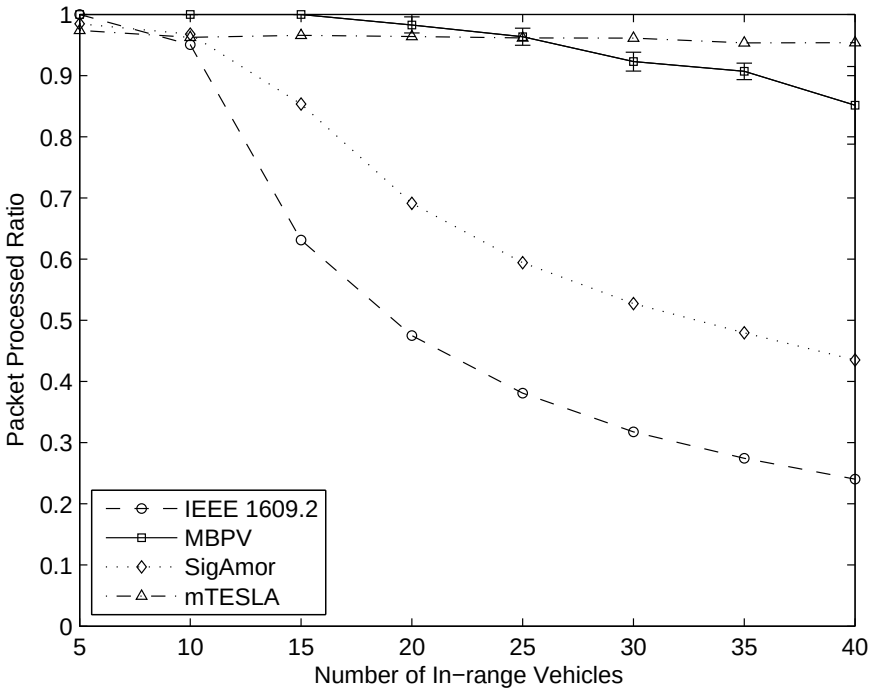

Figure 7.8: Packet Processed Rate vs Number of In-range Vehicles for 200ms broadcast interval.

Similarly for 200 and 300 ms interval, IEEE 1609.2 shows a smaller decline in the packet processed ratio as compared to the 100 ms broadcast interval. SigAmor shows an almost similar curve to IEEE 1609.2, such that its packet processed ratio is more than IEEE 1609.2 as the number of in-range vehicles is increased, for 200 ms broadcast interval. While for 300 ms broadcast interval SigAmor has packet processed ratio lesser than IEEE 1609. when number of vehicles is between $5 - 25$.

mTESLA has the highest packet processed ratio in all cases, while MBPV matches to it for 200 ms broadcast interval when in-range vehicles is < 30. For 300 ms broadcast interval MBPV overcomes the mTESLA packet processed ratio. Furthermore, MBPV has the advantage of immediate authentication, non-repudiation and independent authentication over mTESLA.

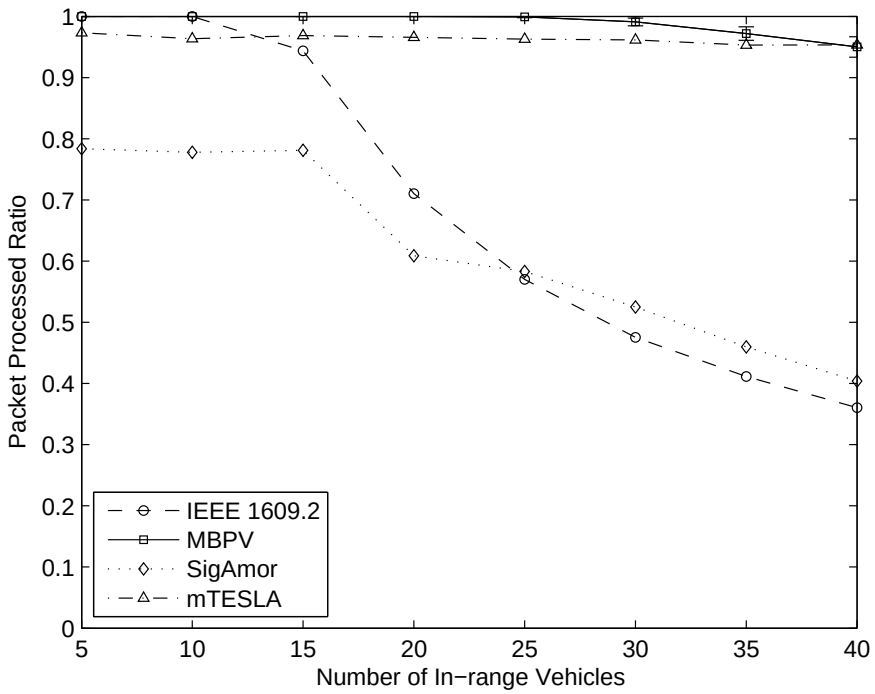

Figure 7.9: Packet Processed Rate vs Number of In-range Vehicles for 300ms broadcast interval.

7.3 Discussion

To understand the results, we study the delay in authentication for one message at the receiver end for each of the four algorithms analyzed, in Table 7.1. IEEE Std. 1609.2 and MBPV consider verifying a message immediately after it is received. However, in mTESLA and SigAmor, message verification is dependent on future packets and is not immediate as mentioned in Section 5.3.

In mTESLA, the authentication key is sent after a certain time delay of broadcasting the corresponding message, to create asymmetry. Similarly, in SigAmor a single signature is created for a group of 'n' messages. All the $n-1$ messages have to wait for the n^{th} message to complete the authentication process, such that the first message has to wait for *n times broadcast interval* at the least. Hence, it experiences the maximum authentication

Table 7.1: Comparison of Delay in Authentication

Scheme	Authentication Delay
IEEE 1609.2	0 ms
MBPV	0 ms
TESLA	100 ms
Signature Amortization	>> 100 ms

delay of all algorithms compared, starting at 100 ms for the $n-1$ message number in case of highest frequency (100 ms broadcast interval) and goes upto *n times broadcast interval*. Since SigAmor uses ECDSA, it authenticates 'n' messages in 22 ms only. However, when the network density increases, more messages are received from increased number of neighbors but due the storage queue constraint some of the packets are dropped.

In VANETs, the vehicles are highly dynamic, such that they move in and out of each other's transmission range very frequently. When mTESLA and SigAmor techniques are applied to highly dynamic vehicles, some packets, including authentication packets of previously received messages, might not reach the receiver. Hence, previously received messages are dropped due to missing authentication information leading to time out of these messages. Therefore, advantages of our algorithm are i) independent authentication and ii) probabilistic verification. MBPV does an independent authentication because the sender performs independent creation of the signature and correspondingly each receiver does independent verification without the aid of any other future message or neighbor or RSU. Further, with probabilistic verification we reduce the number of packet losses caused by verification delay by assigning higher probability of verification to messages which are practically more relevant, in terms of distance and direction between the communicating vehicles. The disadvantage of our algorithm is similar to IEEE 1609.2, i.e., the message overhead as compared to SigAmor and mTESLA, since we make use of highly secure ECDSA for authentication.

7.3.1 Feature comparison with Existing Schemes

We compare the features of our scheme, MBPV, with the latest schemes (published in year $2013 - 2014$), in Table 7.2. We first list the protocol used for key agreement in V2I communication. Then, we compare the features for mutual authentication and Certificate Revocation List (CRL) check, in V2I. Further, we compare if the V2V authentication is independent or not. Finally, we compare the adaptation of identity anonymity in both V2I as well as V2V communication.

7.3.2 Vehicle Registration Overhead Comparison

We provide an analysis of vehicle registration overhead for our scheme and its comparison to the latest schemes in Table 7.3. In our analysis, we represent cost of random number generation using symbol C_{ran}; cost of symmetric encryption/decryption using symbol C_{sym}; cost of asymmetric encryption/decryption using symbol C_{asym}; cost of signature verification using symbol C_{ver}; cost of signature generation using symbol C_{sign}; cost of one-way hash operation using symbol C_h; cost of executing XOR operation using symbol C_{XOR}; cost of elliptic curve multiplication using symbol C_{ECmul}; cost of modular operation using symbol C_{mod}; cost of performing bilinear pairing using symbol C_{Bpar}.

We elaborate on the costly operation of the schemes compared to the our efficient vehicle registration, in Table 7.3. EPAS [66] scheme uses asymmetric operations and signature verification increasing the computation cost of vehicle registration. Cross Layer [67] and Group Communication [72] uses elliptic curve multiplication and modular operations which elevate the computation cost of their operations. Privacy Preserving [77] scheme uses bilinear pairing operations which in addition to being costly make impractical assumptions according to [65]. However, our scheme MBPV is efficient in computation cost of the vehicle registration because it uses only symmetric cryptography with hash and XOR operations.

Table 7.2: Feature Comparison with other schemes

Scheme	Key Agreement Protocol (V2I)	Mutual Authentication (V2I)	CRL Check (V2I)	Identity Anonymity (V2I)	Independent Authentication (V2V)	Identity Anonymity (V2V)
EPAS [66]	Modified Diffie Hellman	Y	Y	Y	N	Y
Cross Layer [67]	Group Key	N	N	-	N	Y
Privacy Preserving [77]	Bilniear pairing	N	N	-	N	Y
Group Communication [72]	Elliptic Curve Arithmetic	N	N	-	N	Y
MBPV	Symmetric key	Y	Y	Y	Y	Y

Table 7.3: Vehicle Registration Overhead

Scheme	OBU operations cost	Registration Entity operations cost
EPAS	$C_{ran} + 3 \quad C_{sym} + 2\, C_{ver}$	$C_{ran} + 3\ C_{asym} + C_{ver} + C_{sign}$
Cross Layer	-	$2 \quad C_{ran} + n \quad C_{ECmul} + n\, C_{mod} + C_h$
Privacy Preserving	C_{Bpar}	$2\, C_{ran}$+$2\, C_{asym}$+$3\, C_{cyc}$+ C_h
Group Communication	$2\, C_{ECmul} + C_h$	-
MBPV	$C_{ran} + 3 \quad C_{sym} + 3\, C_h + C_{XOR}$	$C_{ran} + 4\ C_{sym} + 4\ C_h + C_{XOR}$

7.3.3 Mobility Prediction Model Validation

We validate our mobility prediction model in the following way. At time t_i, each vehicle records the predicted location coordinates for time t_{i+1}. Then, at time t_{i+1}, it computes the error in estimation of the location. Though the vehicular mobility is quite predictable, there might be times when the vehicle stops when a destination is reached or at traffic lights. Vehicles might slow down as they act in accordance with the Krauss car following model. These circumstances might lead to an error in position estimation.

The error in estimation using our mobility prediction model is shown in Figure 7.10. The error is in meters, which varies between $2.1 - 2.4$ m. However, accuracy of location prediction is not the purpose of our algorithm; our focus is to design a broadcast authentication scheme.

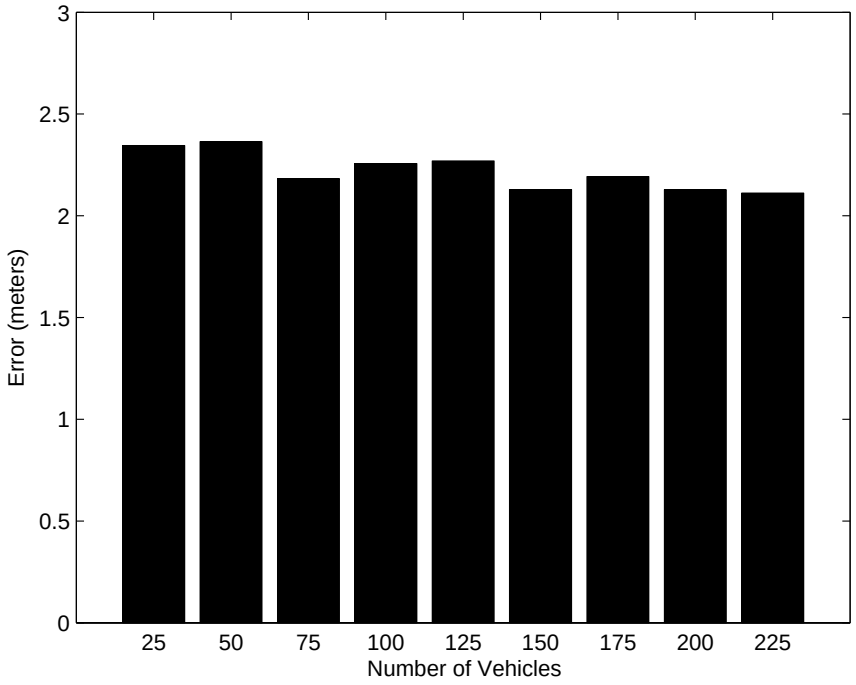

Figure 7.10: Error in Location Estimation using our Mobility Prediction Model

Error Analysis for Mobility Prediction

As we have seen above, the error in our mobility prediction is approximately 2 meters. Further, we analyze this error by comparing it with the ideal case where the mobility prediction is exact and the error in prediction is *zero*. We also use other error values, 6 meters and 10 meters; and compare them with the ideal case.

For error analysis, we use *Scenario 2*. *Scenario 2* is used to study the effect of change in number of broadcasting vehicles within the transmission range of a single receiving vehicle. We vary the number of in-range vehicles from 5 to 40. We perform the simulation with broadcast intervals of 100, 200 and 300 ms.

For each broadcast interval, firstly, we take the ideal case where mobility prediction is exact and record the verification probability values generated for this case. Next, we take the error case, where the mobility prediction is erroneous (2 meters, 6 meters, and 10 meters)

and compute the percentage change of probability values for each error case with respect to the ideal case.

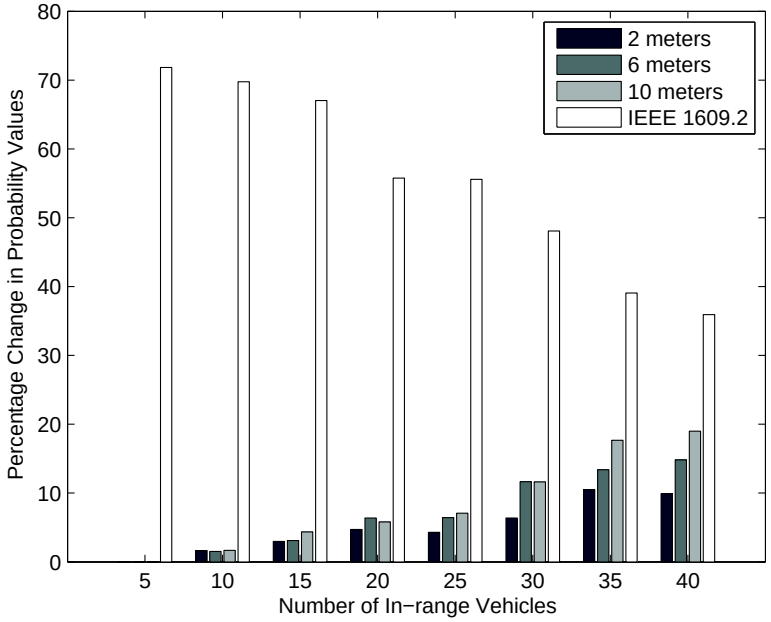

Figure 7.11: Percentage Change of Probability Values vs. Number of In-range Vehicles – 100ms broadcast interval

In Figure 7.11, for 100 ms broadcast interval, we represent the percentage change for probability values of IEEE 1609.2 and other erroneous mobility prediction with respect to the ideal case. The results here show that IEEE 1609.2 has high difference in the probability values that are generated as compared to the ideal case. This is because IEEE 1609.2 uses a probability of 1 to verify all messages. However, the message is considered for verification, only if the message has not expired when taken out of the verification queue. Therefore, we observe that with the increase in number of in-range broadcasting vehicles, the percentage change of probability values for IEEE 1609.2 decreases. This is because more messages are being dropped, as seen in Figure 7.13.

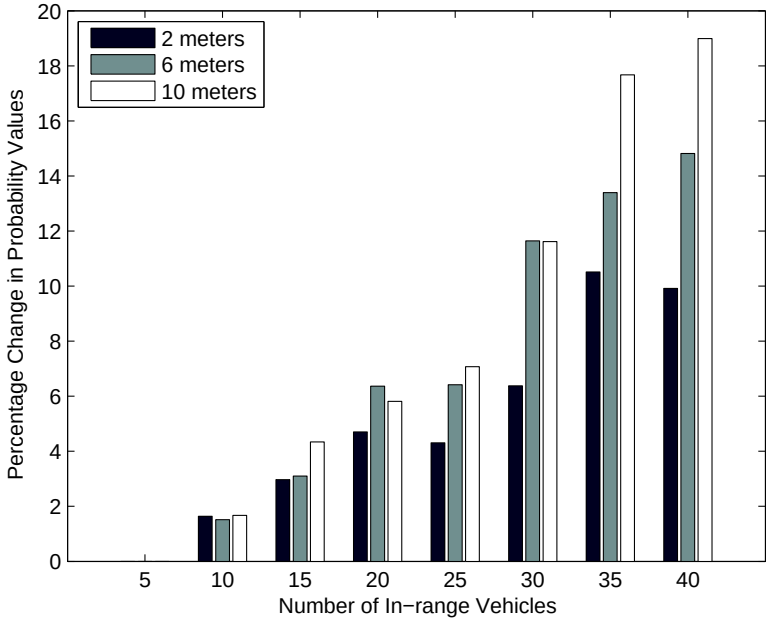

Figure 7.12: Percentage Change of Probability Values for error in mobility prediction – 100ms broadcast interval

Further in Figure 7.12, we present an enlarged view of the percentage change for probability values of erroneous mobility prediction (2 meters, 6 meters and 10 meters) with respect to the ideal case; for 100 ms broadcast interval. It is observed that with the increase in error in mobility prediction the percentage change of probability values increases. Likewise, with an increase in the number of in-range broadcasting vehicles, the percentage change in probability values increases such that it goes up to a maximum of 10% with 2 meters error and 19% with 10 meters error.

Below, we analyze the message loss ratio of all the cases, including the ideal case, 2 meters, 6 meters and 10 meters error cases and IEEE 1609.2 in Figure 7.13, for 100 ms broadcast interval. IEEE 1609.2 has the highest message loss ratio while the message loss ratio for all cases of probabilistic verification have less variation as compared to the ideal case.

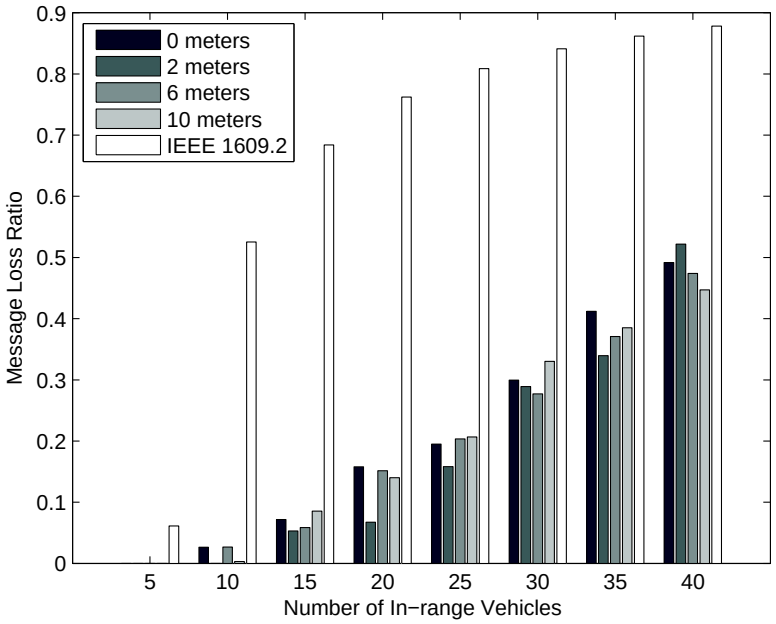

Figure 7.13: Message Loss Ratio for error in mobility prediction – 100ms broadcast interval

After studying the 100 ms broadcast interval results, we further look into the 200 ms broadcast interval and 300 ms broadcast interval. We observe that a pattern similar to 100 ms results is seen in 200 ms broadcast interval in Figures 7.14, 7.15, 7.16 and in 300 ms broadcast interval results in Figures 7.17, 7.18, 7.19.

In Figures 7.14 and 7.15, percentage change in probability values is shown for 200 ms broadcast interval. Here, the percentage change for various erroneous mobility prediction cases is less, as compared to the ones in 100 ms broadcast interval. The values for 200 ms broadcast interval are a maximum of 2.4% with 2 meters error case and 8.4% with a 10 meters error case. The reason for this decrease in percentage change is the decreased frequency of broadcast messages.

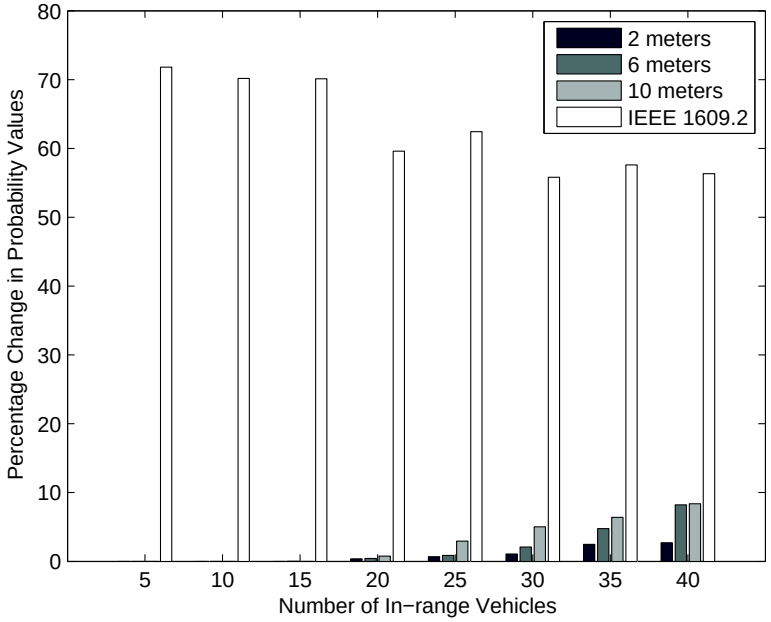

Figure 7.14: Percentage Change of Probability Values vs. Number of In-range Vehicles for 200ms broadcast interval

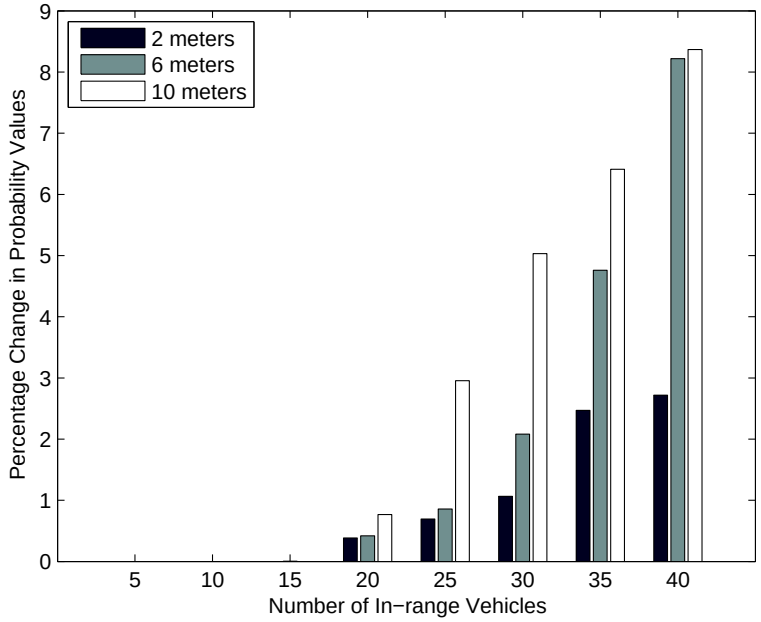

Figure 7.15: Percentage Change of Probability Values for error in mobility prediction – 200ms broadcast interval

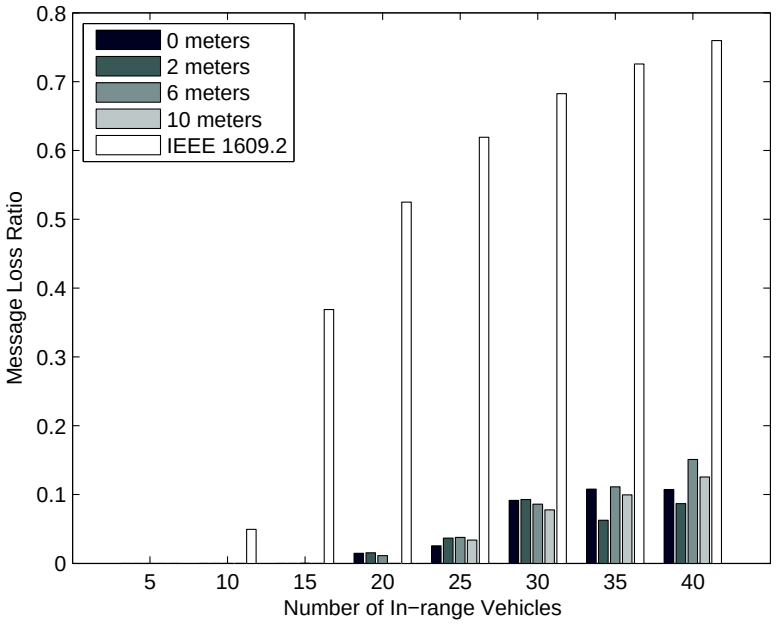

Figure 7.16: Message Loss Ratio for error in mobility prediction – 200ms broadcast interval

Message loss ratio for 200 ms broadcast interval as shown in Figure 7.16, is less than or equal to 15% in all error cases of mobility prediction, whereas the loss is very high for IEEE 1609.2.

Below, we examine the 300 ms broadcast interval in Figures 7.17, 7.18. Here the values for percentage change are maximum of 2% with 2 meters error case and 3.4% with a 10 meters error case. Moreover, for a fewer initial readings with less number of in-range vehicles, the percentage change is negligible. Therefore, we conclude that with the decrease in broadcast frequency, the percentage change of probability values becomes smaller.

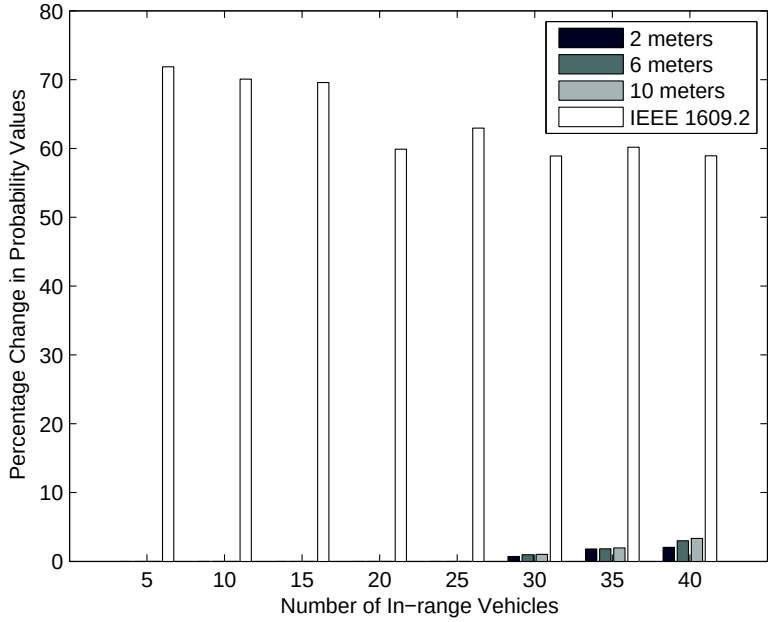

Figure 7.17: Percentage Change of Probability Values vs. Number of In-range Vehicles for 300ms broadcast interval

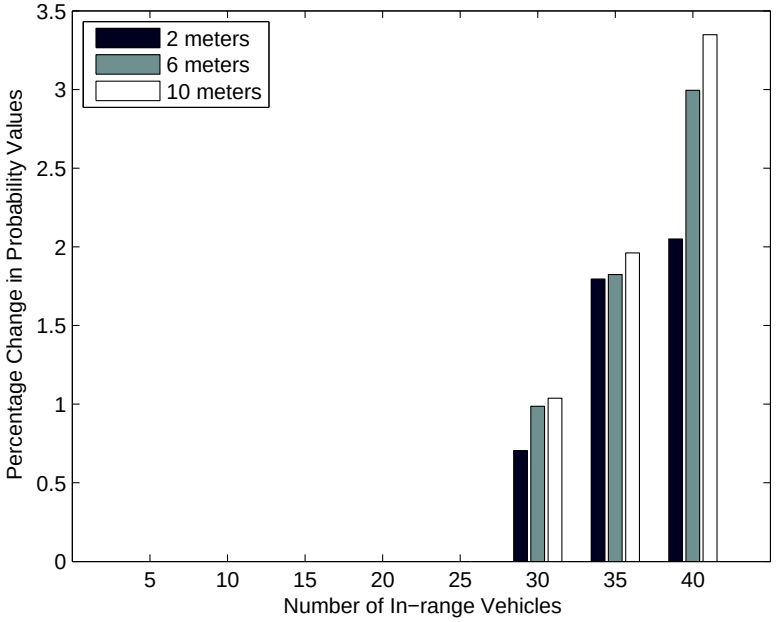

Figure 7.18: Percentage Change of Probability Values for error in mobility prediction – 300ms broadcast interval

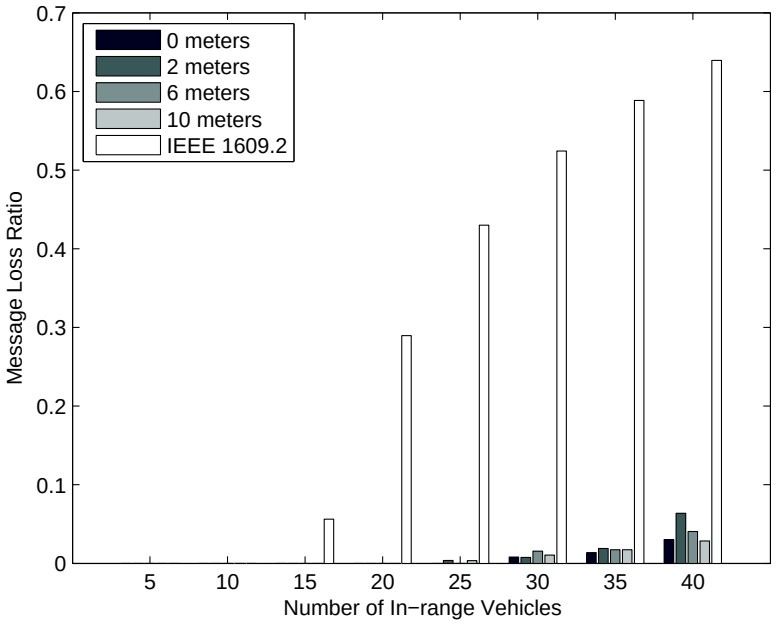

Figure 7.19: Message Loss Ratio for error in mobility prediction – 300ms broadcast interval

Analysis for Past Track History

In our mobility prediction, we use one previous location coordinate for the vehicles, to compute their direction of motion. Below, we will analyze the results by considering 5, 10 and 15 previous locations, for broadcast intervals 100 ms, 200 ms and 300 ms.

Figures 7.20, 7.21 and 7.22 show the results for message loss ratio for cases when 1, 5, 10 and 15 previous location(s) are used for computing the direction of motion of the vehicle. We observe that message loss ratio is higher when more than 1 previous locations are used. This is because an average length of a road segment between two intersections is approximately 50 meters and the maximum speed of vehicles is 20 meter/second while previous locations are logged twice every second, therefore, for one segment of road a maximum of 5 locations can be recorded.

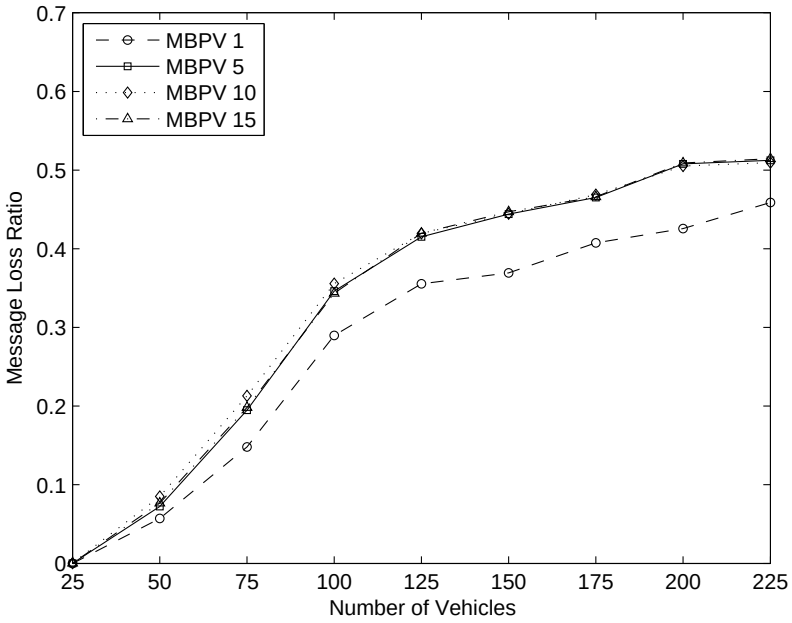

Figure 7.20: Message Loss Ratio for Past Track History – 100ms broadcast interval

Let us consider a scenario here when a vehicle moving straight takes a 90 degree at the next intersection, such that 4 previous locations were recorded before the turn while the last one was recorded after the turn. Now, if we use previous 5 locations for computing the direction of motion of vehicle, it will give us an inappropriate value. Since our purpose is to compute the accurate direction of motion of vehicles communicating with each other, therefore, we only consider one previous location in the computation of direction of motion of vehicles.

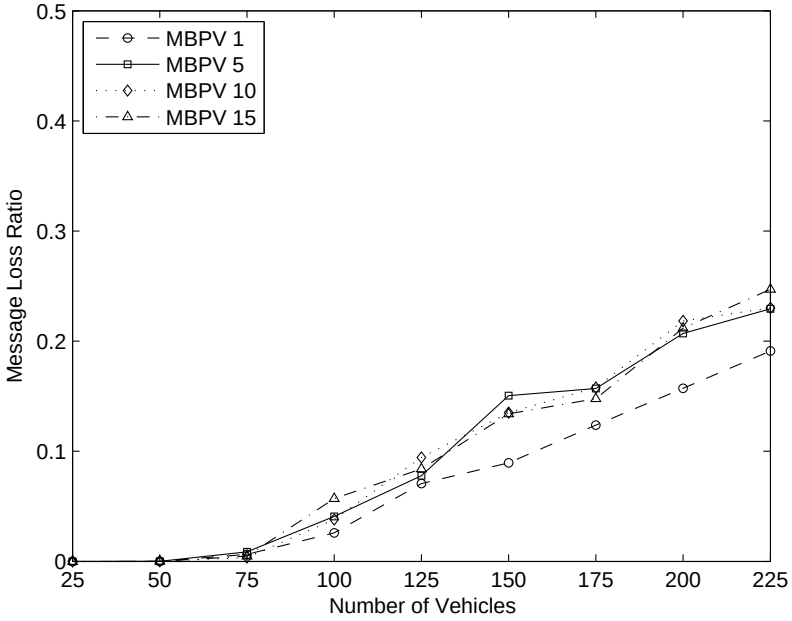

Figure 7.21: Message Loss Ratio for Past Track History – 200ms broadcast interval

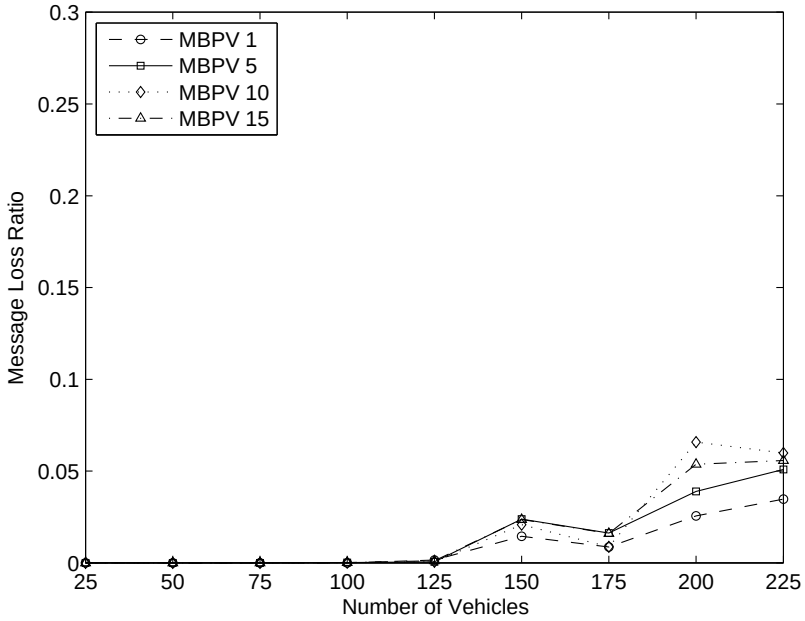

Figure 7.22: Message Loss Ratio for Past Track History – 300ms broadcast interval

7.3.4 Probability Estimation Model Validation

We also validate our probability estimation model. We consider a scenario for this validation, as shown in Figure 7.23. In this scenario vehicle A is moving from west to east and vehicle B is moving from south to north such that they will cross a common intersection point (traffic lights). Note that these vehicles are taken from a real simulation scenario of SUMO, therefore, there will be more vehicles in scenario. However, we consider the communication only between vehicle A and B, for validating our probability estimation model.

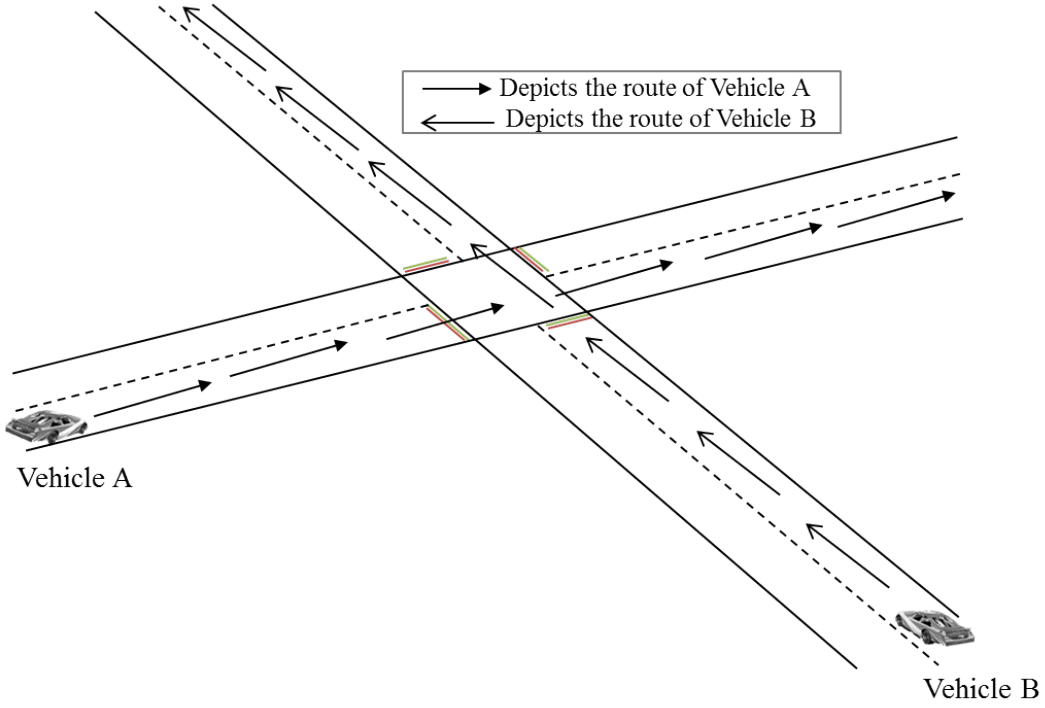

Figure 7.23: SUMO scenario for probability estimation validation

Throughout the validation test, vehicle A and B are in communication range of each other. Both the vehicles comply with the Krauss car following model and adhere to the traffic lights. Acceleration and deceleration abilities of vehicles are 0.8 m/s and 4.5 m/s, respectively; along with a maximum allowed speed of 20 m/s. Vehicle length is 5 m and the minimum gap between vehicles is taken to be 2.5 m

We divide the route of the vehicles into two parts to study the nature of the probability estimation associated with the message verification. The first division of route is considered when the vehicles are coming close to each other such that it ends at the intersection point. The second division of route is considered when the vehicles start at the intersection point and are moving away from each other.

We then study the verification probability values computed at the communicating nodes. These values are displayed with respect to the difference in estimated future distance and calculated present distance. Present distance between the communicating vehicles is calculated, while the future distance is estimated as given in Section 6.2.4. As in our Algorithm 1, we use the difference of these distances for the computation of probability.

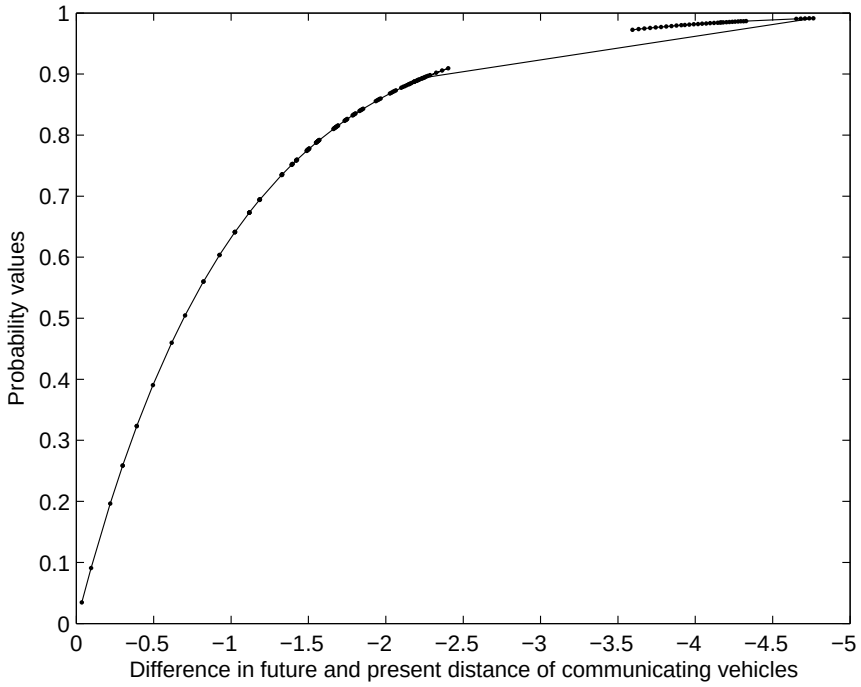

Figure 7.24: Probabilitiy estimation when vehicles are coming close

For the first division of the route, when the vehicles are coming close to each other, we obtain the probability values shown in Figure 7.24. We observe that the initial probability

values start above zero and grow rapidly as the vehicles move towards each other, keeping the highest value as 1.

On the other hand, for the second division of route, when the vehicles are moving away from each other, probability values are displayed in Figure 7.25. In this case, similar to the previous, a rapid change (decrease in probability) is seen in the initial stages followed by a slow decline towards the later stages.

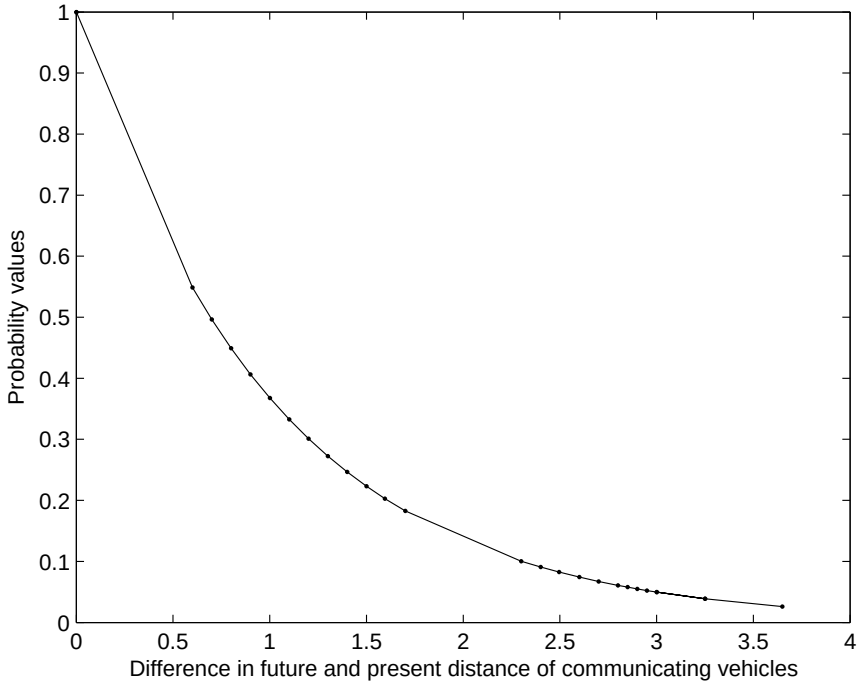

Figure 7.25: Probabilitiy estimation when vehicles are moving away

Therefore, we see that our probability estimation model exhibits a growth of probability values when the communicating vehicles are coming close to each other and a decline in the probability values when the communicating vehicles are moving away from each other.

7.4 Security Analysis

7.4.1 Identity Attack

The original identity of an OBU is never transmitted over the wireless channel; in both V2V and V2I communication, an alias or pseudonym of the vehicle's identifier is sent. The creation of the alias or pseudonym always use a random number. In V2V communication, a random number is concatenated with a parameter assigned by the TA and then hashed. Finally, it is XORed with the code created with parameters from the TA, i.e. $P^{i,j}_{OBU} = code^i \oplus ID^i_{OBU}$. Similar operations take place in V2I alias creation: $AID^i_{OBU} = h(p^i \mid\mid R^i_{OBU}) \oplus ID^i_{OBU}$. This make it impossible for the attacker to obtain the original identity of a legitimate vehicle.

7.4.2 Bogus Information Attack

Each vehicle obtains its ECDSA public-private key pair from the Trusted Authority, which are saved in a tamper-proof device in the vehicle's OBU. Because the attackers are not aware of the secret key components of the broadcasting vehicles, it is not possible for them to construct a valid signature. In V2I communication, an OBU sending bogus information will not be able to successfully complete the registration and mutual authentication process because the trusted entity TA assigns hash function and secret encryption key to RA and correspondingly created registration parameters to OBU.

7.4.3 Message Forgery Attack

ECDSA signature with 224 or 256 size key are used (as defined by IEEE 1609.2 standard). Since the attacker is not aware of the key components forming a valid signature, it cannot modify the message, thus keeping the signature valid. This is because L_n leftmost bits of the message hash is used in the signature generation, where L_n is the bit length of

group order n of an elliptic curve base point. In V2I, each registration is done by verifying $code^i = h(secret^i \mid\mid R^i_{OBU})$ which includes a secret random key assigned by the TA, $secret^i = h(ID^i_{OBU} \mid\mid p^i)$, which is difficult to create. Therefore, if any of the message is modified, the verifications at the RA will not be equivalent to the originally computed values.

7.4.4 Replay Attack

Each V2V message contains a timestamp value which is the time of creation of the message. Hence, if the adversary replays the message, the receiver can detect it from the timestamp value. In V2I, if an attacker replays a message, either from RA or OBU, the other entity can immediately reject the message because the random number in the message will be invalid.

7.4.5 Location Traceability

In all V2V communication, first of all we have used pseudonyms created from random number and secret, such that they cannot be traced. Secondly, we have encrypted the location information using a key distributed by the RA with which an OBU successfully registered. Similarly, in V2I, due to the addition of random number in each alias, it becomes impossible for an adversary to trace the path of a legitimate vehicle.

7.4.6 DoS Jamming Attack

We recommend the use of channel hopping countermeasure as resilience to jamming. We present a reactive and adaptive channel hopping strategy, once jamming has been detected in the network. Jamming detection can be done by using packet delivery ratio along with bit error rate. After the jammed channel has been identified, weight based roulette wheel method can be used to select the next communication channel.

Chapter 8

Conclusions and Future Work

8.1 Conclusion

VANET will eventually become an important part of our lives because of the benefits of traffic safety that they offer. Actual implementations of VANETs are awaiting the finalizing of security primitives for practical scenarios that ensure communications are secure. In this work, we propose an efficient and practical authentication mechanism that complies with IEEE 1609.2 in the usage of ECDSA signature generation. However, we modify the verification with the use of a smart probabilistic strategy that accounts for the relative movement and distances of the communicating nodes. We also maintain conditional privacy with the inclusion of a lightweight mutual authentication mechanism between the infrastructure and vehicles. Additionally, we develop a weight based channel hopping anti-jamming strategy for the DoS jamming attack in VANETs.

VANET standard recommends sending out broadcast at a rate of $100 - 300$ ms by all vehicles involved in the vehicular networking. These broadcasts are time-bound and expire after a given period from the time of creation. Besides, the ECDSA algorithm used for broadcast authentication has high verification overhead, i.e., 22 ms on a 400 Mhz processor while the generation of ECDSA signature takes 4 milliseconds on the same processor. Due to the verification overhead many messages expire in the verification queue while waiting to be considered for verification.

This paper presented a practical and effective *mobility-based* probabilistic signature verification solution to the above issue, particularly with the dynamically changing topology of highly mobile vehicles. We also provide conditional privacy for broadcasting vehicles

through a lightweight V2I mutual authentication process using a symmetric key cryptography. Therefore, our approach integrates security and privacy of vehicles in VANET, such that the vehicles cannot be compromised by an outsider and messages cannot be forged by an attacker. We implemented and compared the two most widely used broadcast authentication algorithms, Signature Amortization and TESLA. Our results show that MBPV has an average of 68% reduction in message loss that can be caused by delay in the verification queue.

The solution that we have proposed for jamming attacks is dynamically switching channels in the event that an attack is detected. Our algorithm proposes a channel hopping resilience from jamming attacks. The channel with the highest-weight value occupies the largest interval on the wheel and hence has more probability of being selected. Each interval spans a range between 0 and 1. Each range corresponds to a particular channel. The number of intervals matches the number of channels in the network. The ranges of the intervals keep changing as the weights of the channels are incremented and decremented. So, once the jamming is detected, this scheme can be used as an effective resilience technique.

8.2 Future Work

Here we enlist some of the research guidelines that can be pursued to refine this work.

- *Prevent Message Suppression Attack* - In this work, we have discussed six harmful attacks that can take place in vehicular networks and have developed a prevention against them. However, another serious attack that needs to be considered is message suppression attack. In this attacker drop packets from the network either randomly or by selecting messages to be dropped. These messages might contain important warning information. Therefore, it is important to find a solution to this attack.

- *Universal Jamming Detection Scheme* - We have developed a countermeasure to jamming by using a weight based channel selection technique, which is reactive and adaptive in nature. There are many existing jamming detection methods, however, there is no Universal anti-jamming technology which can detect all types of jammers. It can be an extended line of work to look into all jammer detection possibility.

- *Variants of ECDSA* - ECDSA is considered the most secure broadcast authentication protocol with the drawback of verification overhead. Elliptic Curve Cryptography (ECC) has many variants depending on the size of the parameters chosen to construct the signature. There are ranges for varying the parameters of ECC that can be used for various types of messages depending on the priority of the messages, such as warning messages will have higher priority than traffic related information messages.

- *Implementation in real world* - Since VANETs involve human life, in our work, we have presented our results through ns-2 simulations. There is a need to create a model which will use the privacy, authentication and anti-jamming methods on the real devices in actual vehicles. This implementation is necessary to measure their performance of the developed techniques in the real world.

Bibliography

[1] Federal Communications Commission, "Dedicated Short Range Communications (DSRC) service," http://wireless.fcc.gov/services/index.htm?job=about&id=dedicated_src.

[2] Q. Yang, A. Lim, S. Li, J. Fang, and P. Agrawal, "ACAR: Adaptive Connectivity Aware Routing Protocol for Vehicular Ad Hoc Networks," in *Proceedings of 17th International Conference on Computer Communications and Networks*, 2008, pp. 1–6.

[3] N. H. T. S. ADministration, "2012 motor vehicle crashes: Overview," http://www-nrd.nhtsa.dot.gov/Pubs/811856.pdf.

[4] K. Matheus, R. Morich, I. Paulus, C. Menig, A. Lübke, B. Rech, and W. Specks, "Car-to-car communication-market introduction and success factors," in *5th European Congress and Exhibition on Intelligent Transport Systems and Services (European ITS 2005).*, 2005.

[5] P. Samuel, "Of sticker tags and 5.9 ghz," *ITS International*, 2004.

[6] F. Ahmed-Zaid, F. Bai, S. Bai, C. Basnayake, B. Bellur, S. Brovold, G. Brown, L. Caminiti, D. Cunningham, H. Elzein *et al.*, "Vehicle Safety Communications–Applications VSC-A Second Annual Report." 2011.

[7] Cisco Systems, Inc., "Authentication types for wireless devices," 170 West Tasman Drive, San Jose, CA 95134-1706 USA, Tech. Rep., 2009.

[8] K. Grover and A. Lim, "A survey of broadcast authentication schemes for wireless networks," *Ad Hoc Networks Journal (Elsevier)*, 2014, Accepted.

[9] SAE Std. J2735, "Dedicated Short Range Communications, Message Set Dictionary." Nov. 2009.

[10] IEEE Std 1609.2, "(Revision of IEEE Std 1609.2-2006) IEEE Standard for Wireless Access in Vehicular Environments Security Services for Applications and Management Messages," pp. 1–289., 2013.

[11] D. Johnson, A. Menezes, and S. Vanstone, "The Elliptic Curve Digital Signature Algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63., 2001.

[12] H.-C. Hsiao, A. Studer, C. Chen, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "Flooding-Resilient Broadcast Authentication for VANETs," in *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 2011, pp. 193–204.

[13] "IEEE Standard for Wireless Access in Vehicular Environments (WAVE) - Networking Services," *IEEE Std 1609.3-2010 (Revision of IEEE Std 1609.3-2007)*, pp. 1–144.

[14] J. Kenney, "Dedicated Short-Range Communications (DSRC) Standards in the United States," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[15] M. Raya and J. Hubaux, "The security of vehicular ad hoc networks," in *Security of Ad Hoc and Sensor Networks (SASN)*, Alexandria, Virginia., November 2005.

[16] B. Parno and A. Perrig, "Challenges in securing vehicular networks," in *Workshop on hot topics in networks (HotNets-IV)*, College Park, Maryland., 2005.

[17] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The tesla broadcast authentication protocol," *RSA CryptoBytes*, vol. 5., Summer 2002.

[18] S. Mccanne, S. Floyd, and K. Fall, "ns2 (network simulator 2)," http://www-nrg.ee.lbl.gov/ns/., 1997.

[19] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of IEEE 802.11 modeling and simulation in ns-2 (802.11ext)." 2008.

[20] J. Harri, F. Filali, and C. Bonnet, "Mobility models for vehicular ad hoc networks: a survey and taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 4, pp. 19–41., 2009.

[21] A. Mahajan, N. Potnis, K. Gopalan, and A. Wang, "Urban mobility models for vanets," in *2nd IEEE International Workshop on Next Generation Wireless Networks.*, 2006.

[22] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A survey and comparative study of simulators for vehicular ad hoc networks (vanets)," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 813–828, 2011.

[23] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO - Simulation of Urban MObility," *International Journal On Advances in Systems and Measurements*, vol. 5, no. 3&4, pp. 128–138., December 2012.

[24] D. Krajzewicz, G. Hertkorn, P. Wagner, and C. Rossel, "An example of microscopic car models validation using the open source traffic simulation SUMO," in *Proceedings of Simulation in Industry, 14th European Simulation Symposium*, 2002, pp. 318–322.

[25] S. Krauss, P. Wagner, and C. Gawron, "Metastable states in a microscopic model of traffic flow," *Physical Review E.*, 1997.

[26] M. Nakagami, "The m-distribution-a general formula of intensity distribution of rapid fading," *Statistical Method of Radio Propagation.*, 1960.

[27] W. Zhang and N. Moayeri, "Classification of statistical channel models for local multipoint distribution service using antenna height and directivity," *IEEE 802.16 working group contribution IEEE 802. 16.1 pc-00*, vol. 7., 2000.

[28] V. Taliwal, D. Jiang, H. Mangold, C. Chen, and R. Sengupta, "Empirical determination of channel characteristics for dsrc vehicle-to-vehicle communication," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks*, Philadelphia, PA., 2004.

[29] K. Grover, A. Lim, and Q. Yang, "Jamming and anti-jamming techniques in wireless networks: A survey," *International Journal of Ad Hoc and Ubiquitous Computing, Indersceince*, 2013, Accepted.

[30] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2005, pp. 46–57.

[31] K. Pelechrinis, M. Iliofotou, and S. Krishnamurthy, "Denial of service attacks in wireless networks: The case of jammers," *IEEE Communications Surveys Tutorials*, vol. 13, no. 2, pp. 245 –257, quarter 2011.

[32] A. Mpitziopoulos, D. Gavalas, G. Pantziou, and C. Konstantopoulos, "Defending wireless sensor networks from jamming attacks," in *IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, sept. 2007, pp. 1 –5.

[33] G. Alnifie and R. Simon, "MULEPRO: a multi-channel response to jamming attacks in wireless sensor networks," *Wireless Communications and Mobile Computing*, vol. 10, no. 5, pp. 704–721, May 2010.

[34] R. Muraleedharan and L. A. Osadciw, "Jamming attack detection and countermeasures in wireless sensor network using ant system," in *SPIE the International Society for Optical Engineering*, 2006.

[35] D. Williams, *Probability with martingales*. Cambridge, UK: Cambridge university press, 1991.

[36] L. Lazos, S. Liu, and M. Krunz, "Mitigating control-channel jamming attacks in multichannel ad hoc networks," in *Proceedings of the 2nd ACM Conference on Wireless Network Security*, 2009, pp. 169–180.

[37] I. Broustis, K. Pelechrinis, D. Syrivelis, S. V. Krishnamurthy, and L. Tassiulas, "FIJI: Fighting implicit jamming in 802.11 WLANs," *Security and Privacy in Communication Networks*, vol. 19, pp. 21–40, Oct. 2009.

[38] P. Tague, D. Slater, R. Poovendran, and G. Noubir, "Linear programming models for jamming attacks on network traffic flows," *6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops*, pp. 207–216, April 2008.

[39] Y. W. Law, M. Palaniswami, L. V. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, p. 6, 2009.

[40] A. Wood, J. Stankovic, and S. Son, "JAM: a jammed-area mapping service for sensor networks," in *24th IEEE Real-Time Systems Symposium*, dec. 2003, pp. 286–297.

[41] S. K. Jain and K. Garg, "A hybrid model of defense techniques against base station jamming attack in wireless sensor networks," in *Proceedings of the 2009 First International Conference on Computational Intelligence, Communication Systems and Networks*, 2009, pp. 102–107.

[42] S. Misra, R. Singh, and S. V. R. Mohan, "Information warfare-worthy jamming attack detection mechanism for wireless sensor networks using a fuzzy inference system," *Sensors*, vol. 10, pp. 3444–3479, 2010.

[43] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: attack and defense strategies," *IEEE Network*, vol. 20, no. 3, pp. 41–47, may-june 2006.

[44] S. Khattab, D. Mosse, and R. Melhem, "Modeling of the channel-hopping anti-jamming defense in multi-radio wireless networks," in *Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, 2008, pp. 25:1–25:10.

[45] A. Wood, J. Stankovic, and G. Zhou, "DEEJAM: Defeating energy-efficient jamming in IEEE 802.15.4-based wireless networks," in *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, june 2007, pp. 60–69.

[46] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using channel hopping to increase 802.11 resilience to jamming attacks," in *IEEE 26th IEEE International Conference on Computer Communications*, may 2007, pp. 2526–2530.

[47] H. Wang, L. Zhang, T. Li, and J. Tugnait, "Spectrally efficient jamming mitigation based on code-controlled frequency hopping," *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 728–732, march 2011.

[48] S.-U. Yoon, R. Murawski, E. Ekici, S. Park, and Z. Mir, "Adaptive channel hopping for interference robust wireless sensor networks," in *2010 IEEE International Conference on Communications*, may 2010, pp. 1–5.

[49] K. Pelechrinis, C. Koufogiannakis, and S. V. Krishnamurthy, "Gaming the jammer: is frequency hopping effective?" in *Proceedings of the 7th International Conference on*

Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, Seoul, Korea, 2009, pp. 187–196.

[50] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.

[51] "Roulette wheel selection," http://www.edc.ncl.ac.uk/highlight/rhjanuary2007g02.php/.

[52] D. Hankerson, S. Vanstone, and A. J. Menezes, *Guide to elliptic curve cryptography.* Springer., 2004.

[53] J. Petit, "Analysis of ecdsa authentication processing in vanets," in *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on*, dec. 2009, pp. 1 –5.

[54] M. M. Haklay and P. Weber, "Openstreetmap: User-generated street maps," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[55] F. Ramm, J. Topf, and S. Chilton, *OpenStreetMap: Using and Enhancing the Free Map of the World.* Uit Cambridge Limited, 2010.

[56] U. S. Department of Transportation - National Highway Traffice Safety Administration, "Vehicle Safety Communications Project - Final Report," April 2006. [Online]. Available: http://www-nrd.nhtsa.dot.gov/pdf/surplus/nrd-12/060419-0843/Index.htm.

[57] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Network and Distributed System Security Symposium (NDSS)*, San Diego, California., Feb. 2001.

[58] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," in *IEEE Symposium on Security and Privacy*, Berkeley, California., 2000.

[59] Y. Liu, J. Li, and M. Guizani, "PKC Based Broadcast Authentication using Signature Amortization for WSNs," *IEEE Transactions on Wireless Communications*, vol. 11, no. 6, pp. 2106–2115., 2012.

[60] E. Schoch and F. Kargl, "On the efficiency of secure beaconing in vanets," in *Proceedings of the third ACM conference on Wireless network security*, Hoboken, NJ., 2010.

[61] X. Fan and G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks," *Ad Hoc Networks*, vol. 10, no. 4, pp. 723 – 736., 2012.

[62] C. Zhang, X. Lin, R. Lu, and P.-H. Ho, "RAISE: an efficient RSU-aided message authentication scheme in vehicular communication networks," in *IEEE International Conference on Communications (ICC).* Beijing, China.: IEEE, 2008.

[63] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," in *The 27th Conference on Computer Communications, IEEE (INFOCOM)*, Phoenix, Arizona., 2008.

[64] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Advances in Cryptology ASIACRYPT*. Springer Berlin Heidelberg, 2001, vol. 2248, pp. 514–532.

[65] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113 – 3121., September 2008.

[66] X. Jia, X. Yuan, L. Meng, and L. Wang, "EPAS: Efficient privacy-preserving authentication scheme for VANETs-based emergency communication," *Journal of Software*, vol. 8, no. 8, pp. 1914–1922., 2013.

[67] S. Biswas and J. Misic, "A cross-layer approach to privacy-preserving authentication in wave-enabled vanets," *IEEE Transaction on Vehicular Technology*, vol. 62, no. 5, pp. 2182 – 2192., 2013.

[68] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: a secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456., 2007.

[69] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology - CRYPTO*. Springer, 2004, pp. 41–55.

[70] J. Zhang and Y. Xu, "Privacy-preserving authentication protocols with efficient verification in vanets," *International Journal of Communication Systems*, 2013.

[71] Y. Saez, X. C. Kish, G. Pesti *et al.*, "Securing vehicle communication systems by the kljn key exchange protocol," *arXiv preprint arXiv:1404.1900*, 2014.

[72] S.-H. Kim and I.-Y. Lee, "A secure and efficient vehicle-to-vehicle communication scheme using bloom filter in vanets." *International Journal of Security & Its Applications*, vol. 8, no. 2, 2014.

[73] A. R. Thakur and J. Pimple, "Comparative analysis of performing vehicle to vehicle communication based on two tier approach with high security."

[74] K. Grover, A. Lim, and S. Lee, "Efficient authentication approach for highly dynamic vehicular ad hoc networks," *Special Issue on Dynamism and Mobility Handling in Mobile and Wireless Networking, International Journal of Ad Hoc and Ubiquitous Computing, Indersceince*, 2014, Accepted.

[75] M.-C. Chuang and J.-F. Lee, "PPAS: A privacy preservation authentication scheme for vehicle-to-infrastructure communication networks," in *International Conference on Consumer Electronics, Communications and Networks (CECNet)*, 2011, pp. 1509–1512.

[76] D. Gavalas, C. Konstantopoulos, and G. Pantziou, "Mobility Prediction in Mobile Ad Hoc Networks," *Next Generation Mobile Networks and Ubiquitous Computing*, pp. 226–240., 2010.

[77] J. Zhang and Y. Xu, "Privacy-preserving authentication protocols with efficient verification in vanets," *International Journal of Communication Systems*, 2013.