On Achieving Secure Message Authentication for Vehicular Communications

by

Chenxi Zhang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2010

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Vehicular Ad-hoc Networks (VANETs) have emerged as a new application scenario that is envisioned to revolutionize the human driving experiences, optimize traffic flow control systems, etc. Addressing security and privacy issues as the prerequisite of VANETs' development must be emphasized. To avoid any possible malicious attack and resource abuse, employing a digital signature scheme is widely recognized as the most effective approach for VANETs to achieve authentication, integrity, and validity. However, when the number of signatures received by a vehicle becomes large, a scalability problem emerges immediately, where a vehicle could be difficult to sequentially verify each received signature within 100-300 ms interval in accordance with the current Dedicated Short Range Communications (DSRC) [1] protocol. In addition, there are still some unsolved attacks in VANETs such as Denial of Service (DoS) attacks, which are not well addressed and waiting for us to solve. In this thesis, we propose the following solutions to address the above mentioned security related issues.

First of all, to address the scalability issues, we introduce a novel roadside unit (RSU) aided message authentication scheme, named RAISE, which makes RSUs responsible for verifying the authenticity of messages sent from vehicles and for notifying the results back to vehicles. In addition, RAISE adopts the *k-anonymity* property for preserving user privacy, where a message cannot be associated with a common vehicle.

Secondly, we further consider the situation that RSUs may not cover all the busy streets of a city or a highway in some situations, for example, at the beginning of a VANETs' deployment period, or due to the physical damage of some RSUs, or simply for economic considerations. Under these circumstances, we further propose an efficient identity-based

batch signature verification scheme for vehicular communications. The proposed scheme can make vehicles verify a batch of signatures once instead of one after another, and thus it efficiently increases vehicles' message verification speed. In addition, our scheme achieves conditional privacy: a distinct pseudo identity is generated along with each message, and a trust authority can trace a vehicle's real identity from its pseudo identity. In order to find invalid signatures in a batch of signatures, we adopt group testing technique which can find invalid signatures efficiently.

Lastly, we identify a DoS attack, called signature jamming attack (SJA), which could easily happen and possibly cause a profound vicious impact on the normal operations of a VANET, yet has not been well addressed in the literature. The SJA can be simply launched at an attacker by flooding a significant number of messages with invalid signatures that jam the surrounding vehicles and prevent them from timely verifying regular and legitimate messages. To countermeasure the SJA, we introduces a hash-based puzzle scheme, which serves as a light-weight filter for excluding likely false signatures before they go through relatively lengthy signature verification process. To further minimize the vicious effect of SJA, we introduce a hash recommendation mechanism, which enables vehicles to share their information so as to more efficiently thwart the SJA.

For each research solution, detailed analysis in terms of computational time, and transmission overhead, privacy preservation are performed to validate the efficiency and effectiveness of the proposed schemes.

Acknowledgments

I would like to express my deepest gratitude to my advisor Professor Pin-Han Ho. Thank him for expert guidance, kindness and encouragement; thank him for his continuing guidance and support during my four years of research; thank him for understanding and supporting my future career direction. His valuable suggestion and wise counsel on my research has been a tremendous force for the completion of this thesis. From him, I have learned so many things, including the research sense, writing papers, giving presentations, and so on. Most importantly, from him, I have learned the ability of how to do research.

I would like to express my extreme appreciation to my thesis committee members: Professor Changcheng Huang, Professor Raouf Boutaba, Professor Guang Gong and Professor Anwar Hasan. Thank them for their precious time to read my thesis and provide valuable suggestions and comments, which are great helpful for improving the quality of this thesis. Thanks to Prof. Guangzhe Fan for serving as the delegated dissertation committee member.

I would like to express my great gratitude to Professor Xuemin Shen, who has given me many precious advice and revision on collaborated research papers. Also, I would like to express great gratitude to Professor Xiaodong Lin, who gave me lots of invaluable suggestions on my research and helps on my life in Canada. I also would like to say "Thank you" to Ms. Wendy Boles who helped me a lot.

I would like to express the special gratitude to Dr. Rongxing Lu, who gave me previous advice on my research. Also, thank all my academic brothers at the ECE department for continued support and warm working atmosphere. It is indeed a great honor to work with so many great talents during my Ph.D. study at the University of Waterloo. Special thanks go to Dr. Haojin Zhu, Mr. Yanfei Fan, for day-and-night discussion and continuous support throughout my graduate study.

There are many friends whose names are not mentioned here. It does not mean that I have forgotten you or your help. My PhD life would not be a life without the many great moments that I have spent with my friends from the doctoral school. They made my academic life fruitful and colorful. Thank you!

I would never get this far without the support of my parents. Thank them for all the unconditional love and support to me. Their love and encouragement have been and will always be a great source of inspiration in my life.

Greatly thank my wife, Xiaoting Sun, who supports me with her endless love.

Contents

List of Tables

List of Figures

List of Abbreviations

AS	Authentication Server
BLS	Boneh-Lynn-Shacham Signature Scheme
CA	Certificate Authority
CDH	Computational Diffie-Hellman
CL	Camenisch-Lysyanskaya Signature Scheme
CRL	Certificate Revocation List
DSRC	Dedicated Short Range Communications
DoS	Denial of Service
DDH	Decisional Diffie-Hellman
DMJA	Dummy Message Jamming Attack
ETSI	European Telecommunications Standard institute
ECDSA	Elliptic Curve Digital Signature Algorithm
FCC	Federal Communications Commission
GBS	Generalized Binary Splitting
IBV	ID-based Batch Verification
IBC	Identity Based Cryptography
IBS	Identity Based System

IDS	Intrusion Detection System
ITS	Intelligent Transportation Systems
IVC	Inter-Vehicle Communication
LR	Loss Ratio
MAC	Media Authentication Code
MAD	Message Authentication Delay
MANET	Mobile Ad-hoc Network
MDS	Misbehavior Detection System
MITM	Man-In-The-Middle Attack
OBU	On Board Unit
PDA	Personal Digital Assistant
SJA	Signature Jamming Attack
PKI	Public Key Infrastructure
RAISE	RSU-aided Message Authentication Scheme
RSU	Roadside Unit
RVC	RSU-to-vehicle Communication
TA	Trust Authority
TESLA	Timed Efficient Stream Loss-tolerant Authentication
TLS	Transport Layer Security
TTP	Trust Third Party
V2I	Vehicle-to-infrastructure Communication
VANET	Vehicular Ad-hoc Network
V2V	Vehicle-to-vehicle Communication

VP	Verification Percentage
VSC	Vehicle Safety Communications
WSN	Wireless Sensor Network
XOR	Exclusive-OR

List of Notations

R	An RSU
V_i	The *i*-th vehicle
M_i	The message sent by V_i
K_i	A key shared between V_i and an RSU
TS_i	A timestamp that records the current time when V_i sends a message
ID_i	A pseudo identity of V_i
ID_j^i	A part of the ID_i such that $ID_i = (ID_1^i, ID_2^i)$ where $j = 1$ or 2
PK_R	The public key of R
PK_i	The public key of V_i
SK_R	The private key of R
SK_i	The private key of V_i
SK_j^i	A part of the SK_i such that $SK_i = (SK_1^i, SK_2^i)$ where $j = 1$ or 2
$Cert$	A certificate
$\{M\}_{PK}$	Encrypt M with a public key, where PK is the public key of the receiver
$\{M\}_{SK}$	Digital signature on M, where SK is the private key of the signer
$\{0,1\}^*$	A binary stream with arbitrary length
$H(.)$	A one-way hash function such as SHA-1

$H^n(M)$	$H^{(i)}(M) = H(H^{(i-1)}(M))$, $i > 0$, and $H^{(1)}(M) = H(M)$
h_i	the i-th element of a hash chain
h^{V_i}	A recommendation hash of vehicle V_i
$h(.)$	A MapToPoint hash function such as $h : \{0,1\}^* \rightarrow \mathbb{G}$
$H_n(.)$	A hash function mapping $\{0,1\}^*$ to an integer in $[0, n]$
$MAC_k(.)$	A message authentication code, which is generated with a symmetric key k
$\|$	Message concatenation operation
$\mathbb{G}$	A cyclic additive group
$\mathbb{G}_T$	A cyclic multiplicative group
$gcd(a, b)$	The greatest common divisor of a and b
$\mathbb{Z}_n^*$	$\mathbb{Z}_n^* = \{a$ is an integer, $a \in [0, n] \mid gcd(a, n) = 1\}$
P	The generator of the cyclic additive group $\mathbb{G}$
$\hat{e}$	A bilinear map: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$
q	The order of the group $\mathbb{G}$
r	A random nonce
s_i	The i-th private master key of a tamper-proof device, i is equal to 1 or 2
P_{pubi}	The i-th public key of TA, where i is equal to 1 or 2
RID	The real identity of the vehicle
PWD	A password or authentication credential
σ	A digital signature
F	A filter for mitigating signature jamming attack
Δ	A time threshold
$\oplus$	An Exclusive-OR operation

Q	A puzzle question
S	The key to a puzzle
l	The security length of a puzzle

Chapter 1

Introduction

1.1 Background

According to Traffic Safety Facts Annual Report [2] from the National Highway Traffic Safety Administration, nearly 6 million police-reported motor vehicle crashes occurred in the United States alone in 2006, leading to 1.75 million injuries and 38,588 deaths. According to the 2006 Annual Report on Traffic Congestion in the Denver Region [3], each resident on average faces about 32 hours of congestion delay per year. Travel during rush hours takes 27% longer than non-rush hours. $1.7 billion per year is lost due to the traffic delays. The above numbers indicate that the traditional traffic crash alert and traffic control systems should be meliorated in order to improve the quality of the public transportation. Fortunately, wireless communication communities and car manufactures are gearing up to develop new technologies to make cars smarter [4]. Vehicles and roadside infrastructure are equipped with wireless communication devices and constitute a vehicular ad hoc network (VANET). VANET aims at improving the road safety and avoid potential traffic accidents. It has also been envisioned to improve driving experiences by providing Internet services to the drivers and passengers and supporting possible e-commerce activities.

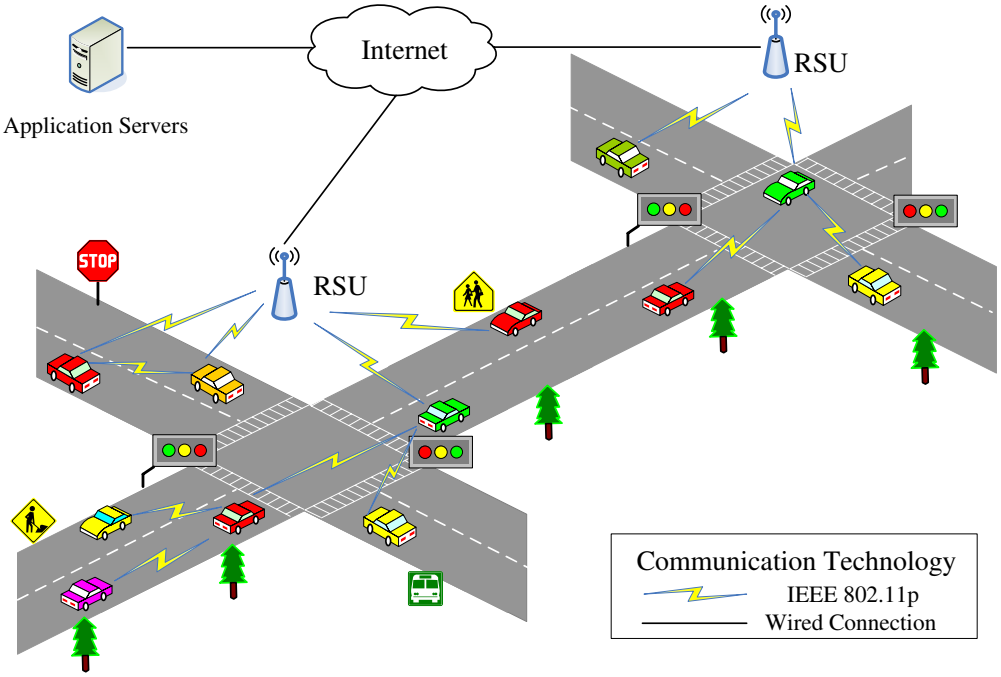


Figure 1.1: The system model

1.1.1 Vehicular Ad Hoc Networks

Fig. 1.1 shows the system model of a VANET. A VANET is composed of vehicles and roadside infrastructure units (RSUs). Vehicles are equipped with wireless communication devices, which are called On-Board Units (OBUs). The wireless communication devices enable vehicles to exchange traffic related information with each other and with RSUs.

Both vehicle-to-vehicle communication (V2V) and vehicle-to-infrastructure (V2I) communication rely on the IEEE 802.11p media access control protocol [5]. RSUs are not necessarily connected directly. They could be indirectly connected via a wired backbone network or the Internet. The backbone network or the Internet could also connect RSUs with some application servers that provide particular services for drivers on the road.

As a new type of mobile ad hoc networks (MANETs), a VANET possesses a number of unique features compared with a MANET. The unique features are presented as follows.

- *No power constraint*: Unlike laptops, personal digital assistants (PDAs), or sensors in mobile or sensor networks, vehicles are not subject to power constraint. Thus, power constraints are not considered in this research.
- *High mobility*: The speed of vehicles in cities ranges from 0 to 60 km/h and can reach up to 120 km/h on highway. In addition, unlike MANETs, the mobility of vehicles is not random. Instead it is in accordance with road directions.
- *Highly dynamic topology*: Due to the fast mobility of vehicles, VANETs have a highly dynamic topology. However, for vehicles driving on the same direction, their relative positions change slowly.
- *High density*: The density of vehicles could be very high at some locations such as intersections or highways where there are traffic jams.
- *Additional information*: Sensors equipped by vehicles can collect environmental information such as road surface temperatures, emergent braking, and vehicle speeds. Thus, more promising applications could be explored by taking advantage of VANETs.
- *Roadside unit assistance*: There are Roadside units (RSUs)located at important locations, such as intersections and exits of highways, to broadcast critical information for drivers. Compared with vehicles, RSUs are stationary and have higher computational power.

Some of these unique features in VANETs may assist us in designing security-related schemes. We integrate some of these features into our designs that will be shown in the following sections.

1.1.2 VANET Applications

The applications of vehicular networks are classified into two categories: safety-related and infotainment-related applications. The safety-related application is the main objective of

VANET , which is expected to improve the driving safety. According to the safety natures, the safety-related messages are further categorized into three groups: warning messages, assisting messages, informing messages.

- *Warning messages*: Both RSUs and vehicles can send warning messages. For example in the scenario I of Fig. 1.2, two running vehicles are approaching towards a sharp-turn intersection. RSU broadcasts warning messages to alert coming vehicles to be aware of the sharp turn and slow down. Another example is shown in the scenario III of Fig. 1.2, two vehicles are running on the road with a high speed, and the front vehicle suddenly makes a emergent brake. The front vehicle immediately broadcasts the brake alert message to its neighbors. Therefore, drivers in following vehicles can notice the emergent brake in time and might avoid a potential collision.

- *Assisting messages*: Vehicles send assisting messages when potential collision would happen. For example, when the inter-vehicle distance of two vehicles is too short, an assisting alert message will be sent by the front car to the next car. The driver in the next car could choose to slow down or change lanes subsequently.

- *Informing messages*: RSUs will broadcast some informing messages when an abnormal traffic flow occurs. For example, in the scenario II of Fig. 1.2, there is a car collision, which incurs a serious traffic jam. In this case, RSUs spread the traffic jam messages to vehicles within certain miles to let them be aware of the traffic jam and choose other less busy roads if possible. Enlightened by this example, we can take advantage of VANETs to control traffic flow specially during the rush hour in a city. In addition, if emergency vehicles such as ambulances, police cars, and fire trucks are approaching, RSUs could notify the vehicles that are far away from the emergency vehicles to make way for them in advance.

In addition to providing safety-related applications, VANETs can also provide a bunch of infotainment-related applications. All these applications should not interfere with safety-

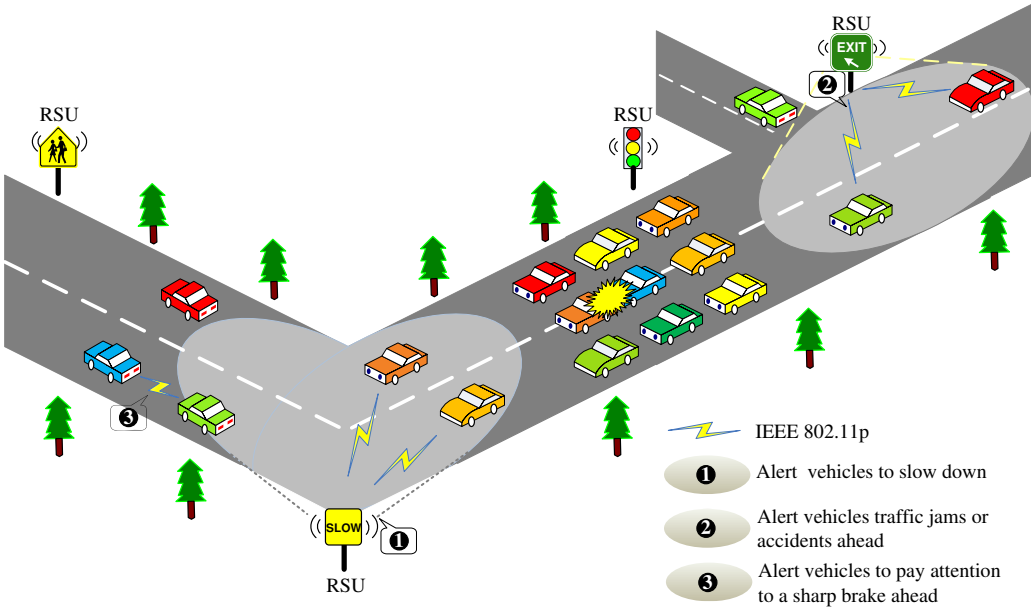


Figure 1.2: VANET Applications

related applications. Some examples are presented below.

- RSUs can assist drivers in finding surrounding interesting places, such as the closest restaurant, coffee shop, plaza, and gas station. When vehicles encounter an RSU, they send their requests to the RSU which will then searches its database, and responses vehicles with their interests. RSUs can also be located at the entrance of a parking lot, and tell whether the parking lot is full or not. If it is not full, RSUs inside the parking lot are able to guild the vehicle to find a parking spot [6].

- RSUs can serve as gateways for Internet access. For instance, RSUs are able to assist vehicles to upload/download mp3 or small-size videos. Passengers in the vehicles can also send/receive emails, browse web sites, and play online games. These internet services are transmitted through RSUs.

- RSUs can be used for business purposes. RSUs could help some stores to broadcast advertisements, such as weekend special offers, weekly flyers, and movie ticket coupons.

Drivers are able to buy some e-tickets directly from RSUs.

- VANETs can be used to collect the environmental information [7]. The sensors on the vehicles could be used to collect data such as weather and humidity information. The data is sent to RSUs, which serve as data collectors.

1.1.3 Dedicated Short Range Communications

Dedicated Short Range Communications (DSRC) [1] is a short to medium-range wireless channels, which is particularly designed for V2I and V2V communication use. In the USA, the Federal Communications Commission (FCC) allocated 75MHz of spectrum in central band 5.9GHz for DSRC in 1999. In Europe, the European Telecommunications Standard Institute (ETSI) allocated 30MHz of spectrum in central band 5.9GHz for DSRC in 2008. Similar activities also undergo in Japan, where 5.8GHz band is used instead. DSRC is exclusively used for Intelligent Transportation Systems (ITS). The decision to use the high frequency spectrum in 5GHz range is because it can achieve high data rate transmissions for long distances even under bad weather conditions, which is suitable for the vehicular communication environment. The DSRC radio technology is a variant of the IEEE 802.11 technology family, which provides a high data transfer rate of up to 27 Mb/s. The transmission range of DSRC is adjustable, and the maximum transmission range is up to 1000 meter. To make DSRC more practical and extendable, IEEE P1609 Working Group have been working on the standardization of the IEEE 802.11p, which is used in physical and medium access control layers for DSRC. The IEEE P1609 Trial-Use Standards Suite includes applications, securities, and management services over DSRC as well.

To make vehicles capable to talk with each other on the road and cooperatively avoid collisions as shown in the scenario III of Fig. 1.2, DSRC requires vehicles periodically broadcast information such as their position, speed, direction of travel, acceleration/deceleration, etc. Messages are sent every 100-300 ms. As such, each vehicle is well aware of its driving environment. Therefore the potential collision or abnormal situations could be alerted and

probably avoided under the assistance of these frequently broadcasted warning messages.

1.2 Research Motivations and Contributions

1.2.1 Motivations

VANETs provide many promising applications that could tremendously benefit our driving experience as mentioned above. However, it is a double-edged sword. VANETs raise many security and privacy concerns at the same time. Malicious cars can take advantage of VANET and disturb the whole system. For example, a compromised vehicle that forges messages to masquerade as an emergency vehicle could mislead other vehicles to pull over, slow down, and yield; a tampered vehicle who masquerades as an RSU could mislead other vehicles to a particular location and cause traffic jam there; if an attacker keeps jamming a significant amount of bogus messages into a VANET, normal message reception and verification will be delayed; the private information such as the driving route of legitimate drivers could be leaked if an attacker passively eavesdrops the traffic related messages in a certain area.

Clearly, these attacks can cause serious problems and probably lead to traffic accidents. Therefore, before VANET applications are deployed and put into commercial use, it is imperative to design security and privacy mechanisms to guard against those malicious attacks.

Since a VANET is a variant of a MANET, some attacks in VANETs also exist in MANETs. These attacks can be addressed using the same approaches as employed in MANETs. However, compared with a MANET, a VANET has unique security and privacy characteristics and challenges, such as fast authentication, conditional privacy preserving, large scale certificate revocation, etc. Therefore, traditional security protocols and mechanisms in MANETs are unsuitable for VANETs any more. This fact motivates us to explore new security protocols that should be dedicated to VANETs.

The new security protocols should take the efficiency problem into consideration. Firstly, a traffic related message should be verified as soon as it is received because a delay to ver-

ify crucial alert messages might cause wrong perception and incur severe road accidents. Second, the security protocol should not generate too much overhead, i.e., communication overhead. Due to the high density of vehicles in VANETs, a high communication overhead may cause wireless channel congestion. Third, a qualified security protocol should be scalable. The desired protocol should be capable to authenticate messages in a timely manner even when the number of vehicles within the communication range is large.

Moreover, a denial of service (DoS) attack ubiquitously exists in any kind of networks, e.g., MANETs, sensor networks, delay tolerant networks, as well as VANET. DoS in each network has a different form of expression. A robust VANET security protocol requires that DoS attack be considered.

1.2.2 Contributions

This thesis is dedicated to developing efficient solutions to address security and privacy issues in VANETs, including message verification, conditional privacy preservation, and DoS attack mitigation. The major contributions of this thesis are summarized as follows:

- We proposed an RSU-aided message authentication scheme, called RAISE, to achieve fast message authentication and conditional privacy preservation. RAISE taking advantage of the assistance of RSUs enables a vehicle to verify traffic related messages very fast, and thus reduce message authentication delay. Meanwhile, RAISE is capable of tracing the real identities of malicious cars.

- In case of the absence of RSUs, we further proposed an efficient ID-based batch verification (IBV) scheme for message authentication in VANETs. IBV works as a supplementary approach to RAISE, and does not need the assistance of RSUs. IBV can achieve the same security and conditional privacy level as RAISE does. In IBV, vehicles verify a batch of signatures once instead of verifying them in a one-by-one manner. The average authentication delay on a batch of signatures can be dramatically reduced.

- We defined a DoS attack which delays normal message authentication in VANETs. We call it invalid signature jamming attack (SJA). The potential threats of SJA are identified. To thwart such an attack, an authentication filtering mechanism is proposed, with which vehicles are able to exclude invalid signatures in the pre-authentication stage before they start the normal signature verification process. The proposed filtering scheme has very low verification overhead. It does not produce any real time verification delay either at the sender side or the receiver side.

1.3 Outline of This Thesis

The remainder of this thesis is organized as follows:

Chapter 2 first introduces the state-of-the-art research about the security and privacy preservation issues in VANETs. The general security requirements, existing attacks and unique security characteristics of VANETs are also discussed in this chapter. Chapter 3 introduces an efficient RSU-aided message authentication scheme (RAISE). Chapter 4 presents an efficient identity-based batch verification scheme (IBV) in the case of no RSU existing in VANETs. Group testing technique is adopted to find invalid signatures within a batch of signatures. Chapter 5 introduces a signature jamming attack (SJA), and employs an authentication filtering mechanism to thwart JSA. Finally, conclusions and future work are given in Chapter 6.

Chapter 2

The Security of Vehicular Ad Hoc Networks

2.1 Introduction

Nowadays, vehicles have been equipped with more and more high-technology devices, e.g., GPS navigators, radars, and OBUs. Thanks to wireless technology, these wireless-enabled devices make vehicles intelligent and be able to "talk" with each other, and thereby form a self-organized VANET. With the assistance of V2V and V2I communications, potentially fatal road accidents can be avoided; dangerous driving behaviors can be alerted; city traffic flows can be optimized; traffic jams can be alleviated.

However, even though VANETs bring tremendous benefits to us, VANETs raise many research challenges as well. One of these challenges is security concerns. In VANETs, malicious vehicles may modify or insert fake information in the network, which could incur life-endangering accidents. In a word, if the security mechanism in VANETs is not carefully designed, misbehavior and malicious attacks may ruin the original intention of VANETs. Therefore, prior to putting VANETs into the practice, it is important to have a robust and efficient security mechanism on board.

Related researches are listed in Section 2.2. Security threats and requirements in VANET are described in Section 2.3 and 2.4. 2.5 presents the Security architecture.

2.2 Related Work

Because of the importance of secure communication in VANETs, recently both industry and academia have paid extensive attentions to addressing security and privacy issues.

US DOT first initiated a Vehicle Safety Communications (VSC) project in 2002. VSC is conducted to evaluate the safety benefits of VANET applications, develops DSRC testing system, and evaluates DSRC communication functionalities for potential vehicle safety implementations. The VSC project investigates the DSRC security, and constructs a proposed security architecture and protocol, supporting message integrity/origin authentication, correctness, privacy, and robustness under attack. In the security system, hierarchical Public Key Infrastructure (PKI) is presented to ensure the authentication between RSUs and OBUs. Considering the privacy of OBUs, the VSC project adopts a short-lived anonymous certificate approach, where a pseudonym is included in a certificate for identification and a new certificate is periodically updated in a short time. If a comprised vehicle is detected, its certificate must be revoked by putting the certificate into a Certificate Revocation List (CRL). This revocation approach is inefficient for certificate validity inquiring once the CRL grows quickly and becomes long.

IEEE working group drafted IEEE 1609.2 [8] standard for Wireless Access in Vehicular Environments - Security Services in 2006. IEEE 1609.2 trial-use standard specifies a range of security services for use in the WAVE environment. It can be used to protect messages from attacks such as spoofing, eavesdropping and alteration. Also, with this standard, vehicles can send encrypted messages to each other or to roadside infrastructures. Nevertheless, as presented in the last section, the communication overhead caused by the security mechanism is large. Although this standard mentions the necessity to respect the drivers' privacy, such

as not leaking personal, identifying, or linkable information to unauthorized parties, it does not provide detailed approaches to achieve this privacy requirement.

In addition to the industrial efforts on security architecture design for VANETs, many researches [9–19] in academic have been conducted to address security and privacy issues. J. Hubaux *et al.* [9, 10] first identified security and privacy issues of VANETs by claiming that an appropriate public key infrastructure must be well devised to protect the transmitted information and to do mutual authentication among network entities. To address the privacy issue, they suggested to rely on temporary pseudonyms to achieve anonymity. To achieve both message authentication and anonymity, Raya *et al.* in [11, 12] proposed that each vehicle should be pre-loaded with a large number of anonymous public and private key pairs together with the corresponding public key certificates. All traffic related messages are signed with a public key based scheme. To achieve privacy, each public and private key pair is used in a short life time and a pseudo ID is used in each public key certificate. Moreover, the authors computed a safe time interval, in which each vehicle should change its pseudo ID at least once so that two consecutive pseudo IDs of the vehicle cannot be linked by an adversary. Clearly, this scheme to protect privacy is straightforward and efficient. But it requires a large storage capacity to store these security information in each vehicle. Further more, on the side of a Trust Authority (TA), it should keep the record of all pseudo IDs and their corresponding key pairs of all vehicles. This is not only inconvenient for the TA to find the real identity of an abusing vehicle, but is also inconvenient to manage these pseudo identities.

In order to overcome the weakness of the above issues, Lin *et al.* in [14] developed a group signature based scheme. With this scheme, vehicles do not require any identities at all. All vehicles within the same group share the same public key, while their private keys are different. When a vehicle receives a signed message, the vehicle verifies it with the group public key. The verifier only knows whether the signer is a legitimate group member or not, but the verifier does not know who the signer exactly is. In this way, the identity privacy is well protected. In case that an dispute happens, TA working as the group manager is capable

to trace the real identity of the sender by using TA's secret key. For example, a group member (as an attacker) broadcasts a malicious bogus message with his/her group private key. Notice that the signature of the message is valid, but the content is fake. Suppose a neighbor of the attacker finds out that the message is bogus. The neighbor reports the message along with the corresponding message signature to TA. The TA using its private key, can compute the private key of the attacker from the signed signature. Then, by looking up the table in which a private key maps to a real world identity of a user, eventually TA can trace the real identity of the attacker. The disadvantage of the group signature based scheme relied in its inefficiency of revocation. If TA revokes a private key of a vehicle, TA has to update the entire security keys of the whole group. The key materials of all group members have to be renewed. In addition, although the group signature based scheme achieves the conditional privacy preservation, the computational cost of verifying a group signature is high, compared with the traditional PKI based signature scheme such as ECDSA. Therefore, the group signature based scheme could result in high message loss ratio under a high traffic density scenario.

Aiming to alleviate the high computational overhead of the group signature based security scheme, Calandriello *et al*. [15] introduced a hybrid scheme that integrates the traditional PKI based scheme and the group signature based scheme. Similar to the above group signature based scheme, each vehicle is assigned a private key and a group public key. The group public key is the same for all group members, and each member holds a distinct private key. Unlike the above scheme of Lin *et al*. in [14], the private key is not used for signing messages. Instead, a signer uses the private key to generate temporary public key certificates. In particular, vehicles generate multiple private key and public key pairs. Each pair has a public key certificate, which mainly contains a pseudo identity and a lifetime, as well as a signature. The signature in the certificate is signed using its group private key by each vehicle itself instead of the trust authority. Here, the lifetime in the certificate indicates how long the certificate is valid. The lifetime should be short such that an adversary cannot link two distinct pseudo identities. The temporary public key and private key pairs work the same

as in [12]. They are used to sign traffic related messages. The revocation process is similar to the group signature scheme in [14]. In case that a dispute happens, TA can trace the real identity of the attacker through the public key certificate because the certificate is signed by vehicles using their group private key. The hybrid scheme makes a tradeoff between the traditional PKI based scheme and the group signature based scheme. Although the hybrid scheme has less computational overhead than the group signature based scheme, it still has higher computation overhead than the traditional PKI based scheme. Therefore, this hybrid scheme cannot address our scalability issues.

The presence of roadside infrastructure units (RSUs) is one of the unique characteristics of VANETs. Some related works take advantage of this feature to achieve privacy requirements. 1) Lu *et al.* in [16] developed a conditional privacy preservation scheme, called ECPP, which divides privacy requirements into three levels. The first level is anticipated by TA. TA is capable to trace the real identity of vehicles from a signed safety message. From the users' perspective, no privacy is defined in the first level. In the second level, each safety message is anonymously authenticated, but an adversary can trace a vehicle by collecting messages. The third level is the strongest privacy level. An individual cannot be traced by collecting messages of the vehicles. In ECPP, RSUs play an important role to achieve the three privacy levels. RSUs are responsible to issue a temporary public key certificate, which vehicles use to sign safety messages. The temporary certificate does not reveal the real identity of a vehicle. Vehicles determine the lifetime of a certificate. When vehicle requests a certificate from an RSU, the vehicle indicates the lifetime of the certificate. Then the RSU issues the requested certificate. Before an issued certificate is expired, a vehicle should request a new certificate from a nearby RSU. The unfixed lifetime is to prevent an adversary from linking multiply certificates of an individual vehicle. RSUs sign each temporary certificate using a group signature scheme, which prevents an adversary from having any knowledge of location information during the use of a certificate. However, TA has the highest authority, which is not only able to recover the real identity of a vehicle from its message signature, but is also

able to reveal the real identity of an issuer (RSU) from a temporary public key certificate. The abilities of TA is used in case of a dispute. 2) J. Freudiger *et al.* in [17] introduced a mix-zone scheme to protect the location privacy for vehicles. An RSU manages a mix-zone, in which vehicles change their pseudo ID and corresponding public key. An adversary cannot link two pseudo IDs from the same vehicle when the vehicle passes through a mix-zone. In the mix-zone scheme, RSUs are located at intersections. Vehicles that go through an intersection process mutual authentication with the RSU, and then obtain a secret key from the RSU. All legitimate vehicles share the same secret key. When vehicles within an intersection send safety messages, vehicles first sign them with their temporary public key and then encrypt the whole message with the secret key. An adversary without the secret key cannot see the content of the message including the used public certificates, and thus the adversary cannot link two pseudo IDs used before and after a vehicle going through a mix-zone. However, the secret key cannot stop a legitimate vehicle from link such two IDs because any legitimate vehicle has the same secret key. Therefore, the mix-zone scheme cannot thwart an internal attack.

Aiming at minimizing the communication overhead and initiate a tradeoff between the security and efficiency, Raya *et al.* in [18] proposed a secure traffic aggregation scheme. Under their design, firstly, cells are defined and predetermined according to the physical road constructions. When vehicles are located in a cell, the vehicle that is physically closest to the center of the cell is automatically taken as the group leader of the vehicles in the cell, which is delegated to aggregate messages for the whole group when the message is going to be relayed to the leader of the neighboring groups. The aggregation of messages can achieve a significant reduction in the overhead for vehicle to vehicle communications. However, the vehicle closest to the center of a cell could change frequently, leading to a frequent update of the group leader of a cell (e.g., once in a few seconds), which indicates that the approach can be further improved in terms of its efficiency and practical applicability.

To evict misbehaving and faulty nodes, Raya *et al.* in [19] introduced a revocation

scheme. The scheme consists of three parties: revocation of trust components, misbehavior detection system, and local eviction protocol. (1) The revocation of trust component scheme is based on using a tamper-proof device, which is embedded in each vehicle and works as an absolutely trust component. Once a Certificate Authority (CA) intends to revoke a vehicle, CA will initiate a protocol with the trust component of a vehicle through roadside infrastructure units in range. The trust component faithfully remove the cryptographic materials (e.g., keys) from the vehicle. Without cryptographic materials, the revoked vehicle cannot generate message signatures any more; (2) The Misbehavior Detection System (MDS) is similar to a traditional Intrusion Detection System (IDS), which can detect anomalies by monitoring network traffic patterns. MDS uses entropy, as a measure of information, to represent the normal and anomalous behaviors of vehicles and then compare them. With MDS, a vehicle can classify messages from its neighbors as faulty or correct. The vehicle sending faulty message is therefore identified as a misbehaving vehicle. Nevertheless, MDS can only work based on locations and time stamps. Under applications without location information, MDS is difficult to detect misbehavior; (3) The local eviction protocol works as a backup of the revocation of trust components because CA may not revoke a vehicle in time or because the communication between CA and a vehicle may be interrupted. To evict misbehaving vehicle locally, vehicles broadcast a warning message indicating a suspect vehicle's identity and position when they believe that there is an attacker in range. The warning message is triggered by MDSs. When the number of warning message regarding a suspect vehicle reaches a predefined threshold, the suspect vehicle is evicted locally. As such, neighboring vehicles ignore all messages sent by the suspect vehicle. Clearly, this scheme is based on voting evaluation, and thus the scheme is not robust enough when the traffic density is low.

Even though the previous works addressed some security and privacy issues for various applications, unfortunately few of them put the emphasis on the scalability issue that is caused by the security mechanism. Further more, none of previous works addressed both the scalability issue and conditional privacy preservation.

2.3 VANET Security Threats

In VANETs, there are several possible security threats, which are summarized as follows:

- *False information attack:* Attackers diffuse false information in VANETs. For example, an attacker sends a false message telling its neighbors that there is a road accident ahead. This false message could lead its neighboring vehicles to choose other ways and leave the road for the attacker to drive.

- *DoS attack:* Attackers attempt to disturb the normal service of a VAENT. For example, attackers continuously send a huge number of dummy messages to wireless channels aiming to bring down the channel transmission. For a sophisticated attacker, he may send a large number of messages with invalid signatures. In this case, legitimate vehicles will spend a lot time verifying invalid signatures, while delaying verifying legitimate message.

- *Replay attack:* Attackers resend the messages received from other legitimate vehicles, to disturb the traffic. The duplicated messages make a vehicle fail to know its neighbor's correct driving status, e.g., direction, position, speed, etc.

- *Impersonation attack:* For malicious purposes, an attacker actively masquerades as another vehicle by using false identities to attack and fool other vehicles. Furthermore, an attacker could pretend to be an RSU and send fake advertisements.

- *Message modification attack:* Attackers may alter their perceived information, e.g., direction, position, speed etc, to escape the consequence of a criminal accident event.

- *Privacy attack*: Attackers may abstract the real identity of a target vehicle from its traffic related messages, and further knows the vehicle location information.

- *Trajectory disclosure attack:* Attackers may globally observe trajectories of a targeted vehicle and use the trajectory information to analyze the identity of the targeted vehicle.

2.4 VANET Security Requirements

The security requirements are directly linked to the threats that are mentioned above. Generally speaking, five requirements should be met in VANETs to deal with the above mentioned threats: authentication, confidentiality, integrity, conditional privacy, and scalability. These requirements are fundamental so that every VANET application should follow.

2.4.1 Authentication

Authentication is the ability to ascertain that a user or a thing is indeed the one that it claims to be. Particularly in VANETs, authentication means verifying the identity of a vehicle and distinguishing legitimate vehicles from unauthorized vehicles. Message authentication is important in VANETs because it ensures that a received message is really sent from a legitimate and authorized vehicle in the network. For safety applications in V2V communication, the authentication requirement can deal with a masquerade attack. For commercial applications in V2I communication, authentication ensures that each user is authorized and he only has access to services that he is authorized to use. Thereby, authentication is a fundamental access control mechanism in VANETs.

2.4.2 Integrity

Due to the nature of wireless communication, the wireless channel is vulnerable to active attacks, e.g., modifications, additions, and falsification. Integrity is to assure that messages exchanged between vehicles do not suffer from these attacks, and all messages sent by vehicles are delivered unaltered. Therefore, integrity protection is a highly essential requirement for vehicular communications.

2.4.3 Conditional Privacy

Undoubtedly, the driver and the passenger benefited from the traffic related messages automatically sent by other vehicles. However, these messages include the sender's private information such as the vehicle's identity (plate license number), locations, directions, and so on. Clearly, people are not willing to expose these private information to third parties. Hence, a secure mechanism should prevent an unauthorized party from knowing the combination of the real identity and location information. On the other hand, a trust authority (e.g., police officers) has the authority to know the real identity of any individual vehicle in case of criminal chasing by law. Thereby, conditional privacy preservation is essential in VANETs.

2.4.4 Non-repudiation

Non-repudiation is the ability to prevent an authorized vehicle from denying having sent the message or denying the contents of the message sent by it. Non-repudiation is a critical requirement for vehicular communication because it can prevent an attacker from denying the attacks that he/her launched. On the other hand, the property of non-repudiation enables a trust authority to trace the identity of an internal attacker, thus it is useful to thwart internal attacks.

2.4.5 Scalability

Scalability is easily ignored when designing a security protocol in a traditional MANET because the number of users in MANETs is not big and failing to consider scalability could not lead to vital attacks. However, in VANETs scalability is an extremely important factor. Vehicles should be able to authenticate incoming messages in a timely manner even in a high density area. Otherwise, many useful messages will have to be dropped before they could be verified if the security scheme is not efficient in high density areas. Moreover, a scheme that is not scalable is vulnerable to DoS attacks.

2.5 Security Architecture

2.5.1 Public key Infrastructure

VANETs need a hierarchical PKI. In particular, Vehicle and RSUs register themselves in a province. Each province has a CA, which issues a public/private key pair and the corresponding PKC to each vehicle and RSU. On the top of each province CA is a country CA. The country CA issues a certified public/private key pair and PKC to each province CA. Vehicles registered in a country could travel to other countries. Thus, the trust relationship between different country CAs is not hierarchical but flat. Different country CAs can establish mutual trust relationship each other. Vehicles are pre-installed several country CAs' PKC that are treated as trust PKC.

To authenticate messages of a vehicle from a foreign country, vehicles first find out the foreign country PKC in their pre-installed CAs' database, use the found CA's PKC to verify the province PKC of the foreign vehicle, and then use the province PKC to verify the vehicle's PKC. A single failure of verifying any certificate in the above steps leads to the failure of the message authentication.

2.5.2 Key distribution

In VANETs, secure key materials should be distributed among four entities: key manager, vehicle, RSU, and TA.

- *Key manager:* The key manager could be a government office, which is responsible for generating, distributing, and managing private/public key pairs and certificates to vehicles and RSUs. The key manager records all key materials and real information of every vehicle and RSU.

- *Vehicle:* Vehicles obtain key materials from the key manager. Vehicles store their private key in a safety place, for example a tamper-proof device, which is a special

hardware and secure against any compromise attempt in any circumstance [9, 12]. For privacy purpose, vehicles could have several valid private/public key pairs and anonymous certificates.

- *RSU:* Similar to vehicles, RSUs acquire key material from the key manager. It is unnecessary to protect the privacy of RSUs, and thus RSUs do not need to have multiple private/public key pairs and anonymous certificates.

- *TA:* TA as a cental authority has the authorization to reveal the real identity of any vehicle in a network. Therefore, the key manager distributes TA special key materials, which enables TA to recover a specific vehicle's identity from its anonymous certificate.

2.6 Summary

In this section, we have reviewed the related work regarding to the security issues in VANET, and enumerated several security attacks that could potentially be conducted in VANETs. In order to design a robust security system, we have presented essential security requirements, and have clarified the basic security architecture.

Chapter 3

An Efficient RSU-aided Message Authentication Scheme

3.1 Introduction

Since security and privacy are essentially important in VANETs, recently more and more research efforts [20–29] have been put on designing security and privacy preservation protocols. All of the existing solutions adopt asymmetric public key based signature schemes, in which a vehicle signs a message with its private key, and broadcasts the message attaching a signature. Receivers verify the signature using the sender's public key. Such an approach works properly when the traffic density is low. However, unfortunately, in the case of a high traffic scenario, this approach results in much high computation overhead for the receivers. In particular, according to DSRC [1], each vehicle should periodically broadcast a traffic related message. The time interval can varies from 100 to 300 ms. It is common that, at a certain metropolitan area, there are 50-150 vehicles in a vehicle's communication range, which vary from 300 to 1000 m in accordance with DSRC as well. As such, it is required that a vehicle is capable to verify up to 1500 messages per second. Such a large number messages not only induces communication overhead in networks, but also leads to high computation overhead

for receivers. None of traditional PKI signature algorithms can be directly adopted to verify the large number of signatures in a short period.

To accelerate the verification speed and avoid the computation overhead caused by asymmetric cryptography, some researches has been conducted to find better solutions. The Timed Efficient Stream Loss-tolerant Authentication (TESLA) [30, 31] scheme, as a broadcast authentication protocol, is the most promising one of the existing solutions. TESLA needs only a few of hash function operations for message verification, and thus can achieve fast message verification. However, TESLA is not suitable for VANETs. It is not only because the position chance of vehicles is too dynamic, but also because TESLA is a delay authentication protocol while messages in VANETs are delay sensitive. The delay caused by TESLA could be life-endangered in VANETs. In addition, TESLA is also vulnerable to memory exhaustion attacks. These disadvantages imply that TESLA cannot be used in VANETs. However, TESLA enlightens us that symmetric cryptography can be used to accelerate the verification speed. Motivated by this fact, this chapter proposes an efficient RSU-aided message authentication scheme, named RAISE, for VANETs. RAISE explores an important feature of VANETs by employing RSUs to assist vehicles in authenticating messages. With RAISE, vehicles first perform mutual authentication and key agreement with an RSU. Vehicles that received safety messages do not need to verify the message through a conventional PKI-based scheme. Instead, each safety message will be attached with a short Message Authentication Code (MAC) that is generated by a sender under the secret key shared between the sender and an RSU. The RSU then verify MACs and disseminate the results of the authenticity of the safety messages to vehicles in its communication range. The verification of the message can be performed in an extremely fast manner due to the nature of the MAC authentication which just needs fast hash operations. The major contributions of RAISE are to improve the authentication efficiency and reduces the communication overhead.

The rest of the chapter is organized as follows. In Section 3.2, preliminaries including the system model, problem statements, and design objectives are briefly introduced. Sec-

tion 3.3 presents the proposed message authentication scheme in detail and explains how the proposed scheme can ensure security and privacy without incurring high overhead and scalability concerns. Section 3.4 analyzes the performance of the proposed schemes through extensive simulations. Section 3.5 analyzes the security of the proposed schemes. Finally, we give the summary in Section 3.6.

3.2 Preliminaries

3.2.1 System Model

A vehicular communication network hierarchically consists of two layers. The upper layer is composed of Application Servers (ASs) and RSUs. The ASs can be connected with RSUs through secure channels, such as Transport Layer Security (TLS) protocol with either wired or wireless connections. The ASs provide application data for RSUs, and RSUs work as gateways to deliver data to the lower layer, which is composed of vehicles. All vehicles and RSUs keep time synchronization. Vehicles can communicate with each other and with RSUs. In this chapter, we aim at addressing the security issues in the lower layer.

In general, RSUs have higher computation capability than vehicles, and are trusted since it is not easy for RSUs to be compromised. According to DSRC, the communication range of an RSU is adjustable, and thus it can be larger than that of the vehicles, such that some vehicles can hear from the RSU while the RSU may not hear from the vehicles. The locations where the density of vehicles is high will be allocated with an RSU, such as an intersection and any possible traffic bottleneck. Notice that only IVC message authentication is considered when any RSU is available. For those areas with sparse vehicle distribution, the scalability issue will not be a problem, and a conventional PKI-based authentication scheme can sufficiently work well.

3.2.2 Problem Statement

The current IEEE Trial-Use standard [8] for VANET security provides detailed documentation including the choice of cryptosystems. To authenticate a message's sender and guarantee the message's integrity, OBUs or RSUs should sign messages with their private keys before the messages are sent. Fig. 3.1 shows the format of a signed message [8], where a 125-byte certificate and a 56-byte ECDSA signature have to be attached for each 69-byte IVC message. Clearly, the cryptographic overhead (the certificate and the signature) takes up a significant portion of the total packet size.

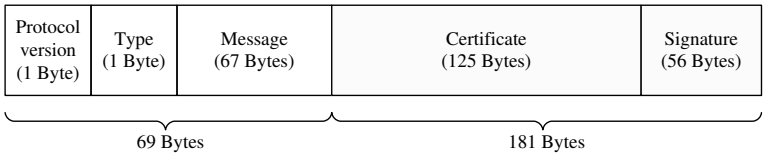


Figure 3.1: The format of the signed message

Cryptographic operations also lead to high computation cost for receivers to verify these messages. According to DSRC [1], a vehicle sends each message within a time interval of 100 ms to 300 ms. Generating a signature every 100 ms is not an issue for current public key based signature schemes. However, in the case that 50-150 vehicles are within the communication range, the receiver needs to verify up to 1500 messages per second. The traditional digital signature algorithm cannot enable a vehicle to verify all of the messages every second. We evaluate the verification speed digital signature schemes, i.e., DSA (2048 bits) and ECDSA (224 bits), on an Intel Pentium IV 3.0GHz desktop with 1GB RAM running Fedora Core 4 based on cryptographic library MIRACL [32]. The verification speed (verifications per second) of DSA and ECDSA are respectively 62 and 258 per second. In addition to message signatures, public key certificates have to be verified as well. Thereby, DSA and ECDSA cannot achieve the desired verification speed.

Furthermore, in addition to safety related messages, non-safety related messages, such as

inquiry information, emails, media data, peer-to-peer forwarding data [25], routing information [33, 34], traffic management information [35], etc., also take up a significant percentage in vehicular communications. These messages can significantly increase as the traffic density increases, and thus the security-related scalability issue for non-safety related messages also needs to be addressed.

To address such an issue, a desired security scheme should be scalable to data traffic in VANETs. Therefore, a verification algorithm is required to be very fast such that received messages by vehicles can be processed timely. Unfortunately, none of traditional digital signature schemes is directly suitable for this time requirement in VANETs.

Embedding a special hardware and increasing the number of computing processors could be an obvious solution to address the scalability issue. However, these approaches certainly increase the cost of a vehicle. In contrast, software solutions are highly desired.

3.2.3 Design Objectives

In this chapter, we aim at achieving the following security objectives:

- *Message integrity and source authentication:* All accepted messages should be delivered unaltered, and the origin of the messages should be authenticated to guard against the impersonation attack.

- *Low communication overhead and fast verification:* The security scheme should be efficient in terms of small communication overhead and acceptable processing latency. A large number of message signatures should be verified in a short interval.

- *Conditional privacy preservation:* The identities of vehicles should be hidden from a normal message receiver during the authentication process in order to protect the senders' private information, such as the driver's identity and any personal information. On the other hand, the authorities should be able to trace the sender of a message by revealing its identity in case of any exceptional case such as liability investigation.

- *Prevention of internal attack:* Different from the study in [36], a normal vehicle holding its own keying material cannot obtain other vehicles' keying materials. Furthermore, even if a vehicle is compromised, an adversary cannot use the compromised vehicle to obtain other vehicles' important information.

3.3 RSU-aided Message Authentication Scheme

3.3.1 Scheme Overview

In this section, we propose an RSU-aided message authentication scheme, named RAISE. Toward a better understanding of RAISE, we first present an overview of RAISE, which includes the following four steps, also corresponding to the steps in Fig. 3.2.

- *Registration:* In the first step, any vehicle equipped with an OBU needs to resister itself to a key manager, who is responsible for distributing a pair of private/public (SK/PK) key, and PKC to each vehicle. In addition, the key manager also plays a role as a security authority, who is capable of tracing their real identities of vehicles form their certificates. This role is important and necessary specially when criminal events happen. Thereby, during the registration, vehicles/drivers must provide the key manager with the real identity information. The registration process needs to be done before V2V and V2I communications.

- *Symmetric Key Establishment:* When vehicles enter the communication range of an RSU, vehicles initiate a mutual authentication process with an RSU. An RSU authenticates vehicles by verifying their signatures. Vehicles compute message signatures with their private keys, and RSUs verify the signatures and their corresponding public key certificates. A valid signature means that the signer of the signature is a legitimate user in VANETs. In a similar way, vehicles can also authenticate an RSU. Meanwhile, the messages that have been signed by vehicles and RSUs include secret credentials,

which can be used to compute a shared key. Here, Diffie-Hellman key agreement [37] could be adopted to establish the shared symmetric *key*. It is worth noticing that different vehicles share different keys with an RSU. Vehicles do not know the key shared between an RSU and other vehicles.

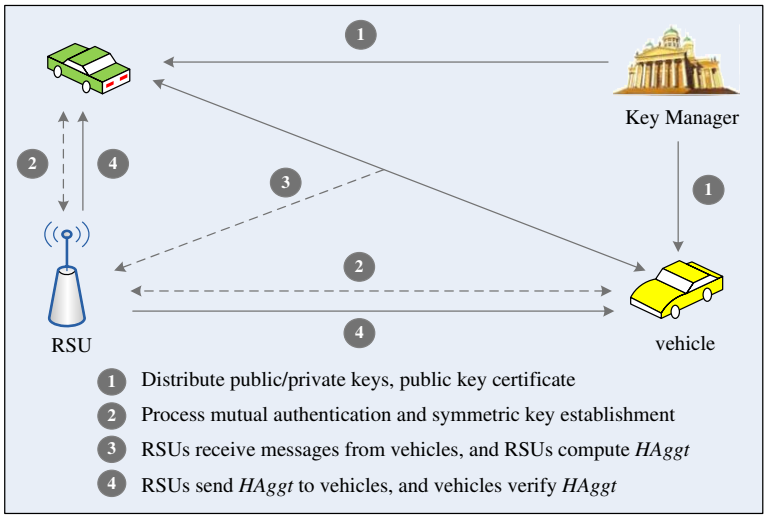


Figure 3.2: The illustration of the message authentication using RAISE

- *Hash Aggregation:* Having the shared key, each vehicle can use it to compute a MAC of a message. Then, each vehicle one-hop broadcast a tuple (pseudo ID, message, MAC) every 300 ms. Neighboring vehicles within the coverage of a vehicle can receive such a message, but they cannot verify the MAC. It is because they do not have the shared *key*. In this case, the neighbors just first buffer the received (ID, message)s for a while. On the other hand, an RSU also receives the message. Since the RSU has the shared *key*, it can verify the MAC. Thereby, in our scheme, RSUs are responsible for verifying MACs. After an RSU verifies all received MACs, it computes hash(ID, message) corresponding to each MAC. Then, it concatenates all computed hashes, which we call *HAggt*. At last, the RSU sends *HAggt* back to vehicles. In addition, the signature on hash(*HAggt*) also needs to be sent with *HAggt*. The period of sending a *HAggt* is

shorter than 300 ms, e.g., 30 ms. This period is adjustable, and it determines message authentication delay.

- *Verification:* Receiving *HAggt* and its signature, vehicles first use an RSU's public key to verify the signature. Then, they hash each (ID, message) which they received within previous 300 ms, and check whether hash(ID, message)s within *HAggt*. If a hash(ID, message) is in *HAggt*, the buffered (ID, message) is verified as a valid message.

Since the first registration step is simple and straightforward, the more explanation of the registration step is skipped. The left three steps, which are symmetric key establishment, hash aggregation, and verification, will be presented in more detail in the following sections.

3.3.2 Symmetric Key Establishment

When a vehicle V_i detects the existence of an RSU R (e.g., through a Hello message from the R), the V_i initiates anonymous mutual authentication and establishes a shared secret key with the R. This can be achieved by adopting the Diffie-Hellman key establishment protocol secured with signature scheme [38]. The mutual authentication and key establishment processes are shown as follows:

$$
\begin{aligned}
V_i \longrightarrow R: &\quad \{aP \mid Cert_{V_i}\}_{PK_R}. \\
R \longrightarrow V_i: &\quad ID_i \mid bP \mid \{ID_i \mid aP \mid bP\}_{SK_R}. \\
V_i \longrightarrow R: &\quad \{ID_R \mid bP \mid aP\}_{SK_{V_i}}.
\end{aligned}
$$

where aP and bP ($a, b \in \mathbb{Z}_q^*$, P is a generator of an addition group $\mathbb{G}$) are random elements of the Diffie-Hellman key establishment protocol, and the shared session key between the R and V_i is $K_i \leftarrow abP$. When receiving the first message from the V_i, the R decrypts $\{aP \mid Cert_{V_i}\}_{PK_R}$ ($\mid$ as a concatenation operation) with its private key SK_R, and then verifies the V_i's public key PK_{V_i} in the anonymous certificate $Cert_{V_i}$. Then, the R sends $ID_i \mid bP \mid \{ID_i \mid aP \mid bP\}_{SK_R}$ to the V_i. The V_i verifies the signature $\{ID_i \mid aP \mid bP\}_{SK_R}$ on $ID_i \mid aP \mid bP$. At last, the V_i sends back the signature $\{ID_R \mid bP \mid aP\}_{SK_{V_i}}$, where

ID_R is the identity of the R, and the R verifies the signature. If the above three steps are completed correctly, the mutual authentication succeeds. Note that the mutual authentication in the protocol is provably secure (refer to [38] for more details). The pseudo identity ID_i that the R sends to the vehicle V_i in the second flow is uniquely linked with K_i[1]. With ID_i, R can know which vehicle sends the message, and can further verify the authenticity of the message with their shared symmetric key. Therefore, R maintains an ID-Key table in its local database, as shown in Fig. 3.4(a). Vehicles update their anonymous certificates once they get out of the radio range of an RSU. For instance, vehicles choose a new public/private key pair [12] to sign messages. In Fig. 3.4(a), T_i denotes the time when R receives the latest message from V_i. T_i is used to determine the freshness of a record. If the interval between the current time of R and T_i exceeds a pre-defined threshold, the record corresponding to T_i will be removed from the ID-Key table and stored into the trace evidence table as shown in Fig. 3.4(b), which will be used for the purpose of traceability. The LT_i in Fig. 3.4(b) is used to control how long trace evidence is kept. In reality, it is decided by the authority, and is much larger than the T_i in Fig. 3.4(a). The details of the trace process will be discussed in the following section.

3.3.3 Hash Aggregation

Once the vehicle V_i establishes the symmetric key K_i with an RSU R, V_i can use K_i to compute the message authentication code $MAC_{K_i}(ID_i \mid M_i \mid TS_i)$ on $ID_i \mid M_i \mid TS_i$, where ID_i is V_i's pseudo identity assigned by R, M_i is the message to be sent, and TS_i is a timestamp that records the current time when sending the message M_i. TS_i is used to thwart the replay attack. Then, V_i one-hop broadcasts $\langle ID_i \mid M_i \mid TS_i \mid MAC_{K_i}(ID_i \mid M_i \mid TS_i)\rangle$. Because K_i is only known by R in addition to V_i itself, only R can verify M_i. Thus, to make other vehicles

[1] In order to protect the privacy, it is necessary that vehicles do not have unique pseudo IDs. This case will be discussed in Chapter 3.3.5. For ease of representation, we explain the scheme with the assumption that vehicles are allocated with unique pseudo ID in this subsection

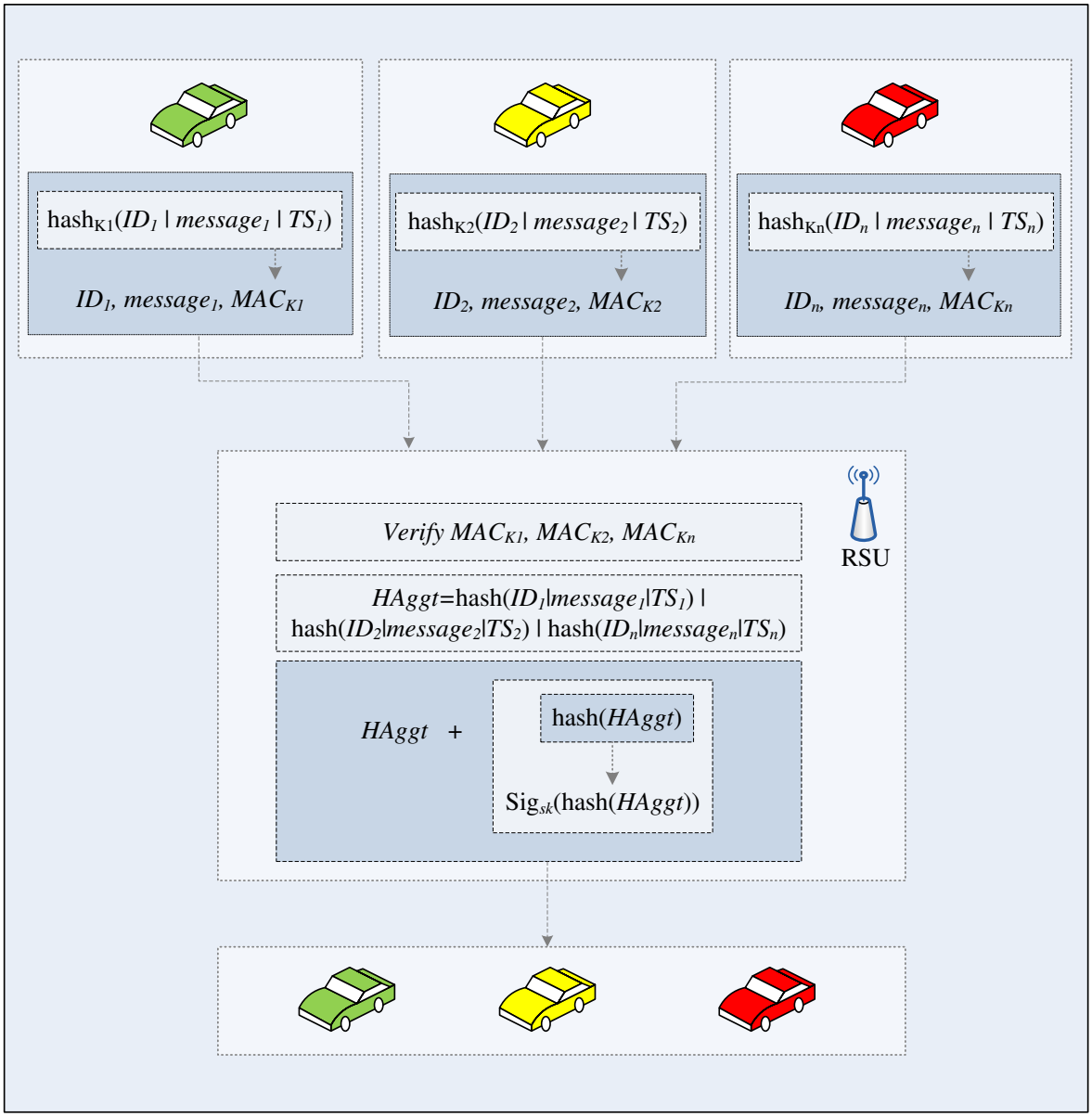


Figure 3.3: The illustration of the message authentication process at an RSU

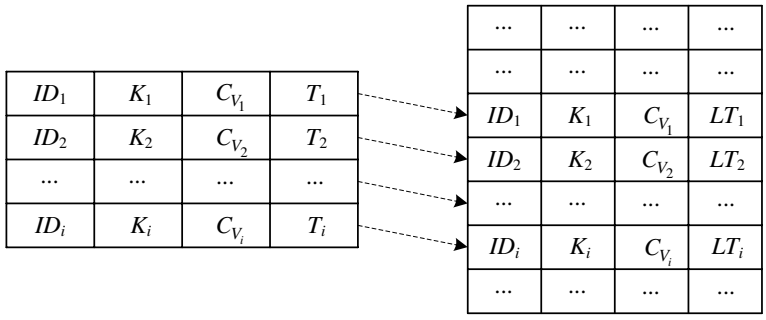


Figure 3.4: (a) The ID-Key table; (b) The trace evidence table

be able to verify the authenticity of M_i, and at the same time to reduce communication overhead, the RSU R is responsible to aggregate multiple authenticated messages in a single packet and send it out. Fig. 3.3 abstractly illustrates the message authentication process at an RSU. The detailed process is given as follows:

1. R checks if the time interval between the current time and the time when R sent the last message authenticity notification packet is less than a predefined threshold Δt. If so, go to Step 2. Otherwise, go to Step 4.

2. When R receives a message, $\langle ID_i \mid M_i \mid TS_i \mid MAC_{K_i}(ID_i \mid M_i \mid TS_i)\rangle$ sent by the vehicle V_i, R first checks whether ID_i is in R's ID-Key table. If yes, go to Step 3. Otherwise, go to Step 4.

3. R uses ID_i's K_i to verify $MAC_{K_i}(ID_i \mid M_i \mid TS_i)$. If it is valid, R computes $H(ID_i \mid M_i \mid TS_i)$. Otherwise, drop the packet. Go to Step 1.

4. R aggregates all hashes generated at Step 3, i.e., $HAggt = H(ID_1 \mid M_1 \mid TS_1) \mid H(ID_2 \mid M_2 \mid TS_2) \mid ... \mid H(ID_n \mid M_n \mid TS_n)$, and signs it with its private key SK_R. Then, R one-hop broadcasts $\langle HAggt \mid \{HAggt\}_{SK_R}\rangle$ to vehicles within its communication range.

The predefined threshold Δt in the above algorithm can affect message authentication delay, which will be further discussed in Chapter 3.4.2. In addition, the above algorithm supports the identity traceability property. Since there is a one-to-one mapping between the key K_i and the certificate $Cert_{V_i}$ in the trace evidence table, the RSU can distinguish the unique sender of a message. Thus, in case that a malicious vehicle sends a bogus message (e.g., the context of the message is found to be fake after a while), the RSU can trace back to the message sender by finding out its certificate. The RSU could also report the certificate to a trusted authority for further investigation.

3.3.4 Verification

When a vehicle receives messages sent by other vehicles, it only buffers the received messages in its local database without verifying them immediately. The buffered record has the following format: M_i, ID_i, TS_i, $H(ID_i \mid M_i \mid TS_i)$ (notice that $H(ID_i \mid M_i \mid TS_i)$ is computed by the receiver). Once vehicles obtain the signed packet $\langle HAggt \mid \{HAggt\}_{SK_R} \rangle$ from the RSU, they are able to verify the buffered messages one by one. First of all, vehicles use the RSU's public key PK_R to verify the signature $\{HAggt\}_{SK_R}$. If it is valid, vehicles will check the validity of the previously received messages buffered in the record in the local database. This is done by comparing whether there is a match between the buffered record with the de-aggregated message. For example, V_i checks if $H(ID_i \mid M_i \mid TS_i)$ coming in *HAggt* has been buffered in any record before. If so, M_i is consumed. Otherwise, V_i waits to see if M_i will be in the next *HAggt* packet. If $H(ID_i \mid M_i \mid TS_i)$ does not appear in two[2] successive aggregated *HAggt* packets, M_i is regarded as invalid. The reason that $H(ID_i \mid M_i \mid TS_i)$ is double checked is because the RSU may have not aggregated the message M_i yet when V_i receives the first *HAggt* packet from the RSU. In addition, a vehicle has to be capable of verifying all

[2]Suppose a vehicle V_i receives a message M_j sent by V_j, and then immediately receives an aggregate *HAggt* sent by an RSU R. Since the R could not receive M_j at all before the R sends the *HAggt*, $H(M_j)$ will appear at the following *HAggt*.

incoming messages sent by neighboring vehicles, which means all messages received by the vehicle can be received by its corresponding RSU as well. However, if the communications between the RSU and a vehicle (or RSU-to-vehicle Communications (RVC)) has the same distance limit as that of IVC, a vehicle will lose the messages sent by the vehicles that have not been in the eligible distance with the RSU. Fig. 3.5 shows the illustration. Let the distance limit of RVC be r. The RSU can communicate with vehicles V_1 and V_2. Since V_3 has not associated with the RSU, V_2 cannot verify messages from V_3 although the two vehicles are supposed to be communicable. To overcome this problem, we require the distance limit for RVC to be two times longer than that for IVC. The distance between vehicles and vehicles and between vehicles and RSUs can be derived from GPS coordinates, because the GPS coordinates can be contained in the messages of vehicles.

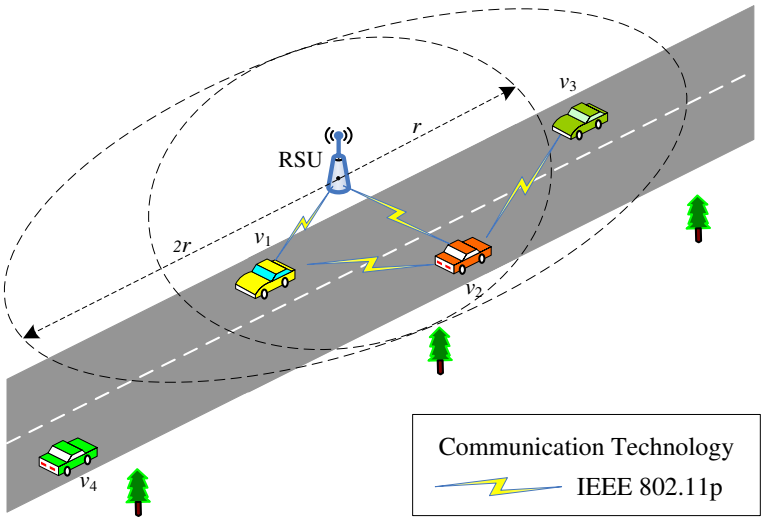


Figure 3.5: The radio range of an RSU

3.3.5 Enhancement of User Privacy

With RAISE, if a vehicle does not change its pseudo ID all the time during the association period, an adversary can trace the vehicle movement trajectory according to the vehicle's unchanged ID. Therefore, the vehicle's trace privacy is violated during the small time duration.

To preserve the trace privacy, we employ the concept of *k-anonymity* (k entities are not distinguishable) [39] in the proposed RAISE scheme to mix k vehicles. With RAISE, RSUs assign a common pseudo ID to k vehicles, where the k vehicles (as a group) will take the same pseudo ID when communicating with the RSU. When an adversary intends to trace a specific vehicle through the pseudo ID, he/she will easily get lost after the group of vehicles passes through an intersection (where an RSU is allocated). In other words, the route of a specific vehicle cannot be identified. The biggest value of k would be the total number of vehicles within the coverage range of an RSU, in which all vehicles' messages are mixed and cannot be distinguished. Notice that such a scenario is equivalent to the case that vehicles have no identity at all.

In the *k-anonymity* RAISE, RSUs can still identify a vehicle by finding the symmetric key shared with the vehicle, and each pseudo ID corresponds to k unique symmetric keys. Suppose a vehicle V_i sends $\langle ID \mid M_i \mid TS_i \mid MAC_{K_i}(ID \mid M_i \mid TS_i)\rangle$ to RSU R. R first finds out k possible keys corresponding to the pseudo identity *ID*. Then, R sequently checks whether $MAC_{K_i}(ID \mid M_i \mid TS_i)$ is equal to $MAC_K(ID \mid M_i \mid TS_i)'$ that is generated by one of the k symmetric keys. If there is a match, the message is considered valid. Since a vehicle holds a distinct key shared with the RSU, the key that makes the above comparison can be used to find the message sender's anonymous certificate that was used during the first mutual authentication process. This can be done by looking up the RSU's local ID-Key table. Being able to find out the anonymous certificate used during the mutual authentication process is to support the future ID traceability property.

However, if there is still no match with the two MAC values after R has tried all possible k keys, the message is considered as invalid and will be dropped. After this process, R can

continue the message aggregation process as presented in Chapter 3.3.3.

With the adoption of *k-anonymity*, the verification process remains the same as before. Vehicles compare whether there is a match between the de-aggregated $H(ID_i \mid M_i \mid TS_i)$ from *HAggt* and the buffered $H(ID_i \mid M_i \mid TS_i)$ value in any record. Here, the cost of comparison computation can be neglected compared with message verification of the PKI-based scheme in [12].

3.4 Performance Evaluation

In this section, we use the ns-2 simulator [40] to evaluate the performance of RAISE in terms of the message loss radio, the message end-to-end delay, and the communication overhead, respectively, compared with the group signature based scheme in [14] and the standard PKI-based ECDSA signature scheme in [8]. We simulate a traffic scenario with high vehicle density. An RSU is located at an intersection, and 30-200 vehicles can associate with the RSU. Notice that an RSU is used only in RAISE, and other schemes do not need an RSU's help. The inter-vehicular distance varies from 7.5 m to 15 m to simulate the scenarios with different traffic densities. The distance limits for IVC and RVC are 300 m and 600 m, respectively. Inter-vehicle messages are sent every 300 ms at each vehicle. IEEE 802.11a is used to simulate the transmission protocol in medium access control layer. The bandwidth of the channel is 6 Mb/s. The group signature verification delay is 11 ms [42]. The ECDSA signature verification delay is 3.87 ms [42] [3]. All possible cryptographic operations in the simulation are considered to have same simulation delay.

[3] In [42], the 224 bits ECDSA cryptographic delays are quoted from MIRACL cryptographic lib [32] with the 3GHz Pentium IV system.

3.4.1 Message Loss Ratio

Average message Loss Ratio (LR) is defined in Eq. 3.1, where N represents the total number of vehicles in the simulation. For the group signature based and PKI-based signature schemes, M^i_{mac} represents the total number of messages received by the *i*-th vehicle in the medium access control layer, and M^i_{app} represents the total number of messages consumed by the *i*-th vehicle in the application layer. For RAISE, M^i_{mac} represents the total number of messages received directly from other vehicles in the medium access control layer; M^i_{app} represents the total number of $H(ID_i \mid M_i \mid TS_i)$s that are sent by the RSU, and are consumed by the application layer. Since RAISE needs two hops communication, we consider the loss caused by wireless communications between the RSU and vehicles.

$$LR = \frac{1}{N}\sum_{i=1}^{N}(M^i_{app}/M^i_{mac}) \tag{3.1}$$

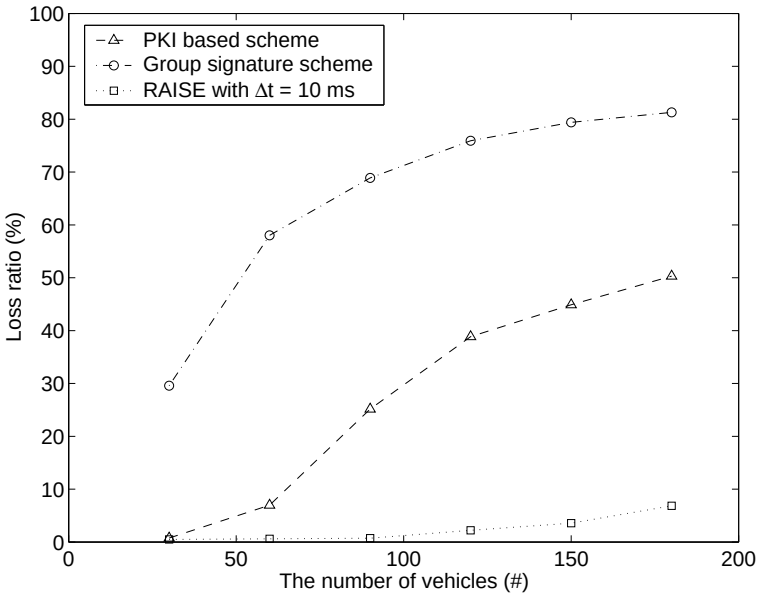


Figure 3.6: Average loss ratio vs. traffic load

Fig. 3.6 shows the relationship between the message loss ratio and the traffic load. The traffic load is represented by the number of vehicles. For RAISE, the RSU periodically broadcasts an aggregation of $H(ID_i \mid M_i \mid TS_i)$s every 10 ms. From Fig. 3.6, we can see that the message loss ratio of RAISE, the group signature, and the PKI-based ECDSA signature schemes increases as the traffic load increases. The group signature based scheme has the highest loss ratio, because a group signature needs the longest time to be verified. If the number of received group signatures is beyond the number that a vehicle has the capacity to verify every 300 ms, the exceeded signatures are dropped. Since verifying a PKI-based (ECDSA) signature is shorter than a group signature, the PKI-based scheme in Fig. 3.6 ranks in the second. RAISE, on the other hand, has the lowest loss ratio. The reason is because RAISE does not require the time-consuming PKI-based signature verification, and instead RAISE only needs fast hash function operations. However, compared with the PKI-based schemes, RAISE needs more communications. From the simulation, most of the message losses of RAISE come from the two-hop wireless transmission.

3.4.2 Message Authentication Delay

Average Message Authentication Delay (MAD) is defined in Eq. 3.2, where N represents the total number of vehicles in the simulation, M is the number of messages sent by the *i*-th vehicle, and K is the number of adjacent vehicles within the *i*-th vehicle's communication range. $T_{recv}^{i,k,m}$ represents the moment that the *k*-th vehicle in the application layer receives the *m*-th message from the *i*-th vehicle. $T_{send}^{i,k,m}$ represents the moment that the *i*-th vehicle in the application layer sends the *m*-th message to the *k*-th vehicle.

$$MAD = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{MK}\sum_{m=1}^{M}\sum_{k=1}^{K}(T_{recv}^{i,k,m} - T_{send}^{i,k,m}) \tag{3.2}$$

Fig. 3.7 shows the relationship between the message authentication delay and the traffic load. Again, the group signature scheme has the highest message authentication delay. This is due to the high delay verifying a message signature. The PKI-based ECDSA scheme and

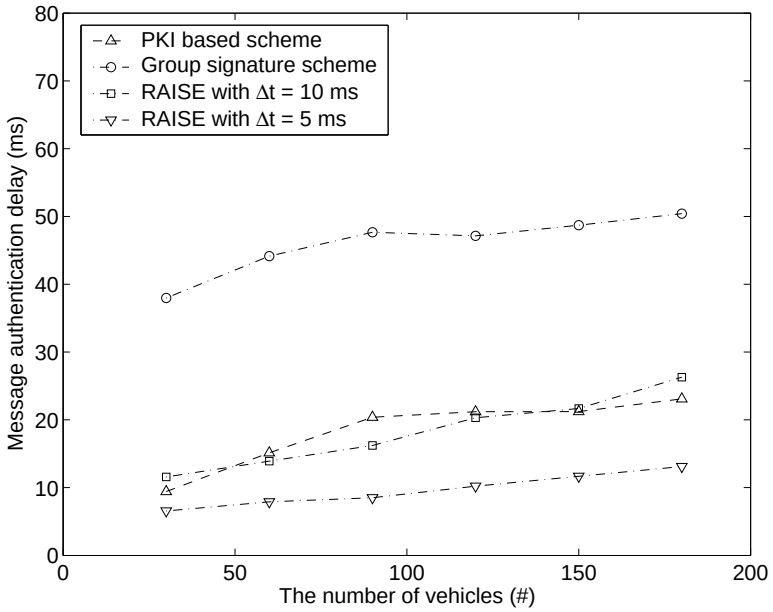


Figure 3.7: Average message authentication delay vs. traffic load

RAISE yield nearly the same message authentication delay. Since the comparison computation is very fast, the delay of RAISE is primarily determined by the packet release interval at the RSU. For example, the packet release interval Δt is 10 ms in our simulation, which serves as the main contribution of the message authentication delay. To reduce the message authentication delay, we may reduce Δt, e.g., $\Delta t = 5$ ms. However, a too small Δt brings two disadvantages in RAISE. First, a smaller Δt produces more communications, which can result in higher message loss ratio. Second, a smaller Δt increases more communication overhead, which will be further discussed in the next subsection.

3.4.3 Communication Overhead

The communication overhead of ECDSA [41], the group signature based scheme [14], and RAISE is respectively listed below. With ECDSA, each message yields 181 bytes as the additional overhead due to cryptographic operations, which includes a certificate and an ECDSA

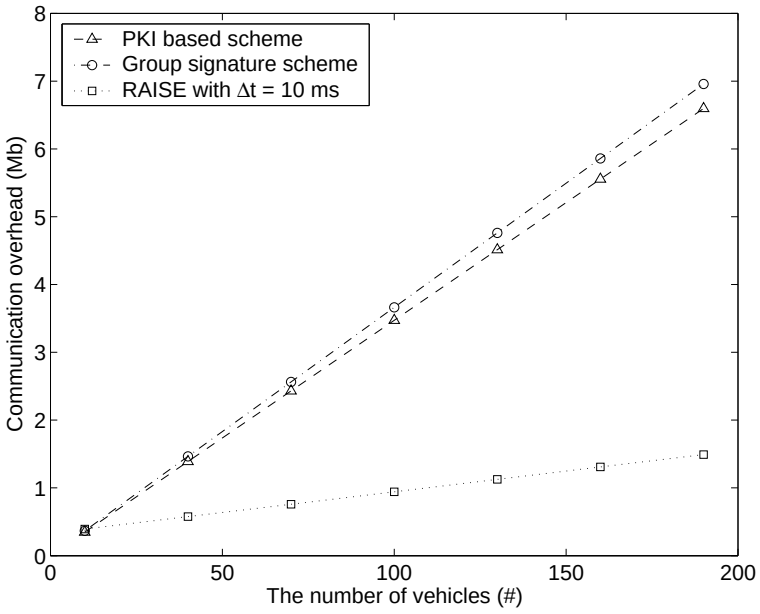


Figure 3.8: Communication overhead vs. traffic load

signature, as shown in Fig. 3.1. With the group signature based scheme, the additional communication overhead is 184 bytes [42]. With RAISE, the additional communication overhead is 128 bits + 128 bits + (56+2)/n bytes, where the first 128 represents the length of a MAC sent by a vehicle, the second 128 bits represent the length of a $H(ID_i \mid M_i \mid TS_i)$ packet that is sent by an RSU, 56 bytes are the length of an ECDSA signature [8] signed by the RSU, and 2 bytes are the length of a message header as shown in Fig. 3.1. Here, 56+2 bytes are shared by n messages, because in RAISE n messages are batched and signed once. Note that n is determined by the density of vehicles and the packet release interval for the RSU to broadcast a batched packet.

Fig. 3.8 shows the relationship between the overall communication overhead in 1 minute and the traffic load within an RSU. We can see that RAISE with the time interval of 10 ms has much lower communication overhead than that of the PKI-based ECDSA signature scheme and the group signature based scheme. Furthermore, the communication overhead of RAISE

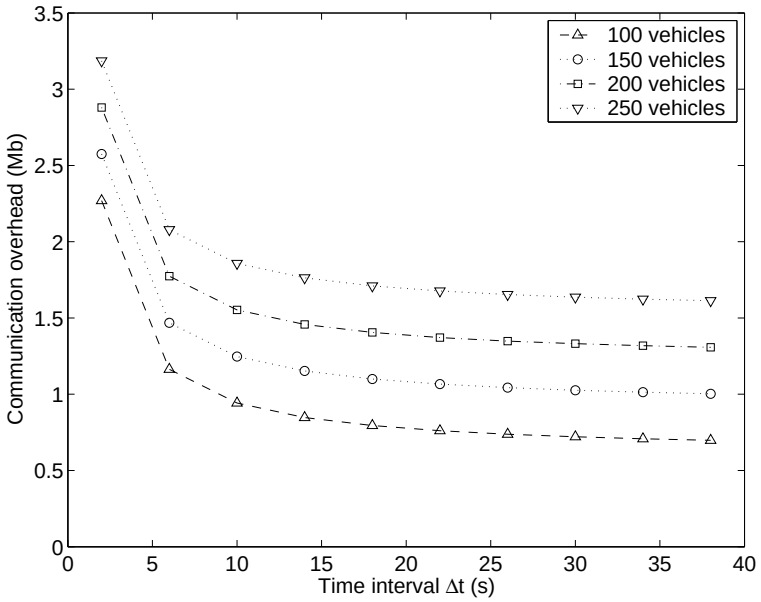


Figure 3.9: Communication overhead vs. time interval

is 24.94% of the PKI-based ECDSA signature scheme and 23.64% of the group signature based scheme.

To further illustrate the effect of the time interval on RAISE, Fig. 3.9 shows the relationship between the time interval and the overall communication overhead, caused by 100, 150, 200, and 250 vehicles, respectively, in 1 minute. Clearly, as the time interval increases, particularly from 2 ms to 10 ms, the communication overhead decreases sharply. However, when the time interval is up to 10 ms or larger, it has very little effect on the communication overhead. This is because the frequency of sending 56+2 bytes decreases as the time interval increases. From Fig. 11, we can also see that the communication overhead increases approximately 0.3 megabytes every time the number of vehicles increases by 50.

3.5 Security Analysis

In this section, we analyze the security of the proposed schemes in terms of message integrity and source authentication, prevention of internal attack, replay attack resistance, and conditional privacy preservation.

Message integrity and source authentication: With RAISE, a vehicle generates a MAC for each launched message. The MAC can only be generated by the vehicle that has the key assigned by the RSU. If an adversary tampers with a message, the RSU cannot find a responding validation key that can compute a matching MAC for the message, and therefore the intercepted message will be ignored. In addition, for each vehicle, there is a unique key stored in the ID-Key table at the RSU side. If an RSU can find out a key to verify a MAC, the RSU can know the identity of the message sender, and therefore the source is authenticated. Due to the nature of message integrity and source authentication, typical attacks such as bogus attack and impersonation attack [12] can be prevented.

Prevention of internal attack: RAISE is robust against not only the external attacks but also the internal attacks. Even if a vehicle is compromised and its symmetric secret session key shared with an RSU is exposed to an adversary, the adversary cannot trace other vehicle's movement because it cannot distinguish the vehicles that use the same pseudo ID with the compromised vehicle. Therefore, RAISE can resist key-compromise impersonation attack.

Replay attack resistance: With a replay attack, an adversary replays intercepted messages in order to impersonate a legitimate vehicle. Clearly, this impersonation cannot work with RAISE because a time stamp TS_i is attached with the corresponding M_i and all vehicles keep time synchronization. Suppose an adversary intercepts a message $\langle ID_i \mid M_i \mid TS_i \mid MAC_{K_i}(ID_i \mid M_i \mid TS_i)\rangle$, and launches a replay attack at the time TS_j. Because the time period $|TS_j - TS_i| > \Delta T$ where ΔT is a mutually agreed transmission delay, the receiver will reject the message. Therefore, RAISE is robust to resist the replay attack.

Conditional privacy preservation: RAISE makes vehicles use pseudo identities to protect their real identities. Nevertheless, RSUs are able to know the anonymous certificate corre-

sponding to a pseudo identity, and a trust authority is capable of tracing the real identity of a vehicle from its anonymous certificate. For example, a vehicle V_i sends a bogus message, which contains the pseudo identity ID_i that an RSU allocates. Once the RSU finds out that the content of the message is bogus, the RSU can know the anonymous certificate of the V_i from the trace evidence table in which the ID_i uniquely maps the anonymous certificate $Cert_{V_i}$ as shown in Fig. 3(b). Further, the RSU gives the certificate $Cert_{V_i}$ to a trust authority, which has the ability to trace the real identity of the V_i from $Cert_{V_i}$. Therefore, in RAISE, vehicles cannot tell their real identities each other, while RSUs can distinguish whether two messages are sent by the same vehicle. The trust authority and RSUs cooperate, which can trace the real identity of a message sender.

3.6 Summary

In this chapter, a novel RSU-aided message authentication scheme, named RAISE, has been proposed. With RAISE, RSUs are responsible for verifying the authenticity of messages sent by vehicles and notifying the authentication results back to all the associated vehicles. The RAISE scheme has many advantages because of its lower computation and communication overhead, which have been justified through extensive simulations. We have presented the security analysis to analyze that RAISE is able to achieve message integrity, source authentication, and prevention of internal attacks. In addition, RAISE also achieves the conditional privacy preservation: the privacy among vehicles is protected by adopting the *k-anonymity* approach; by providing anonymous certificates to TA, RSUs enable TA to trace the real identity of vehicles.

Chapter 4

An Efficient Identity-based Batch Verification Scheme

4.1 Introduction

In the last chapter, we have introduced RAISE, an RSU-aided message authentication scheme. With the assistance of RSUs, RAISE is able to improve authentication efficiency and reduce communication overhead. However, in general, RSUs may not cover all the busy streets of a city or a highway in some situations, for example, at the beginning of a VANETs' deployment period, or due to the physical damage of some RSUs, or simply for economic considerations. Under these circumstances, unfortunately RAISE cannot be used. Thereby, without the presence of RSUs, how to improve authentication efficiency and address scalability issues, and how to acheive conditional privacy preservation in VANETs still motivate us to explore the corresponding solutions.

In this chapter, we address the above issues with a novel approach. We propose an identity-based batch verification (IBV) scheme for message authentication in VANETs. The IBV scheme does not need the assistance of RSUs, and works as a supplementary approach of RAISE. IBV uses identity-based cryptography and batch verification techniques. It has

the following unparalleled features: 1) Multiple signatures can be verified at the same time instead of one after the other as that in the previously traditional PKI signature schemes. Therefore, the signature verification speed can be significantly improved such that the computational workload of each vehicle can be alleviated; 2) By generating distinct pseudo identities and the corresponding private keys for signing each message with a tamper-proof device, privacy regarding user identity and location of the vehicles can be protected; 3) The identities of the vehicles can be uniquely revealed by TA under exceptional cases; and 4) Since identity-based cryptography is employed, efforts on certificate management and the transmission overhead can be significantly reduced. In addition, to find invalid signatures in a batch of signatures, we investigate and adopt some group testing approaches which can find invalid signatures efficiently.

The rest of the chapter is organized as follows. In Section 4.2, background and preliminary knowledge related to the proposed research is given, including the system model, security requirements, identity-based cryptography, bilinear pairing technique, batch verification, and group testing technique. In Section 4.3, the proposed batch verification scheme is described in details. In Section 4.4, the security of the proposed scheme is analyzed. In Section 4.5, group testing approaches are introduced to detect invalid signatures. In Section 4.6, the performance evaluation is presented. Finally, we give the summary in Section 4.7.

4.2 Background and Preliminaries

4.2.1 System Model

In VANETs, there are four entities: TA, key manager, vehicle, and RSU. In this chapter, we investigate the V2V communication where there is no RSU presence and assistance. We only consider V2V communication and omit V2I communication. Each vehicle periodically broadcast a traffic related message. The time period is 100-300 ms. PKI is adopted in our system. Each vehicle has a signing private key and a corresponding public key. Before

sending the messages, vehicles sign messages with their private keys. Vehicles accept the messages only if their signatures are valid. In our system, each vehicle is equipped with a tamper-proof device, which is secure against any compromise attempt in any circumstance. With the tamper-proof device on vehicles, an adversary cannot extract any data stored in the device including key material, data, and code [9, 12].

4.2.2 Security Requirements

The V2V communication scenario is subject to the following three security requirements: *fast message authentication*, *identity privacy preserving*, and *traceability*, which are further discussed as below.

- *Fast message authentication:* Messages from vehicles have to be authenticated to confirm that they are indeed sent unaltered by legitimate ones. Further more, when the traffic density becomes high, the number of messages that a vehicle receives increases. To avoid any possible performance bottleneck, we require a novel signature verification scheme, in which a vehicle can verify as many signatures as possible. The verification scheme should not depend on the assistance of RSUs.

- *Identity privacy preservation:* In vehicular communication, due to its broadcasting nature, overhearing an identity-specific information could happen frequently. If the employed signature scheme is an ordinary digital signature, the signature would easily leak one's identity information [43]. Even though a pseudo identity is employed as a mask, an outside observer can also link multiple signatures to one vehicle through traffic analysis. This issue is called linkability, which may incur a location privacy violation problem [7]. Therefore, identity privacy preserving is required.

- *Traceability:* The TA should have the ability to retrieve a vehicle's real identity from its pseudo identity when the signature is in dispute or when the content of a message is

bogus. Both identity privacy preservation and traceability requirements constitute the conditional privacy preservation requirement which is presented in the last chapter.

In this chapter, we aim to address all the aforementioned security requirements without the assistance of RSUs.

4.2.3 Identity-based Cryptography

Identity Based Cryptography (IBC) is a type of public-key cryptography, in which the public key of a user is unique information identifying the user. The unique information could be a user's name, email address, and IP address. In an Identity Based System (IBS), any party can generate a public key from a well-known identity value, and a Trust Third Party (TTP) generates the corresponding private key, which is also referred to as a master key. TTP publishes a master public key and keeps the corresponding master key. Using the master public key and a known ID, any party can compute the public key corresponding to the ID. The corresponding private key of ID can be obtained from TTP, which uses its's master key to generate the private key for the identity ID.

Compared with the public key cryptosystem with certificates, IBS has some advantages. First, IBS simplifies the certificate management because an identity (a public key) of a user is well-known for the public and it is unnecessary for the public key certificate distribution. Second, since there is no certificate needed in IBS, the communication overhead caused by certificates can also been reduced.

4.2.4 Bilinear Pairing

Since bilinear pairing work as the basis of our proposed identity-based scheme in this chapter, we briefly introduce the bilinear pairing as follows.

Let $\mathbb{G}$ be a cyclic additive group generated by P, and $\mathbb{G}_T$ be a cyclic multiplicative group. $\mathbb{G}$ and $\mathbb{G}_T$ have the same prime order q, i.e., $|\mathbb{G}| = |\mathbb{G}_T| = q$. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be an

bilinear map, which satisfies the following properties:

- *Bilinear*: For all $P,\ Q,\ R \in \mathbb{G}$, and $a, b \in \mathbb{Z}_q^*$, $\hat{e}(Q, P+R) = \hat{e}(P+R, Q) = \hat{e}(P,Q) \cdot \hat{e}(R,Q)$. In particular, $\hat{e}(aP, bP) = \hat{e}(P, bP)^a = \hat{e}(aP, P)^b = \hat{e}(P,P)^{ab}$.
- *Non-degenerate*: There exist $P,\ Q \in \mathbb{G}$ such that $\hat{e}(P,Q) \neq 1_{\mathbb{G}_T}$, where $1_{\mathbb{G}_T}$ is the identity element of $\mathbb{G}_T$
- *Computable*: There is an efficient algorithm to compute $\hat{e}(P,Q)$ for any $P,\ Q \in \mathbb{G}$.

Such an bilinear map $\hat{e}$ is called an admissible pairing, and can be constructed by the modified Weil [44] or Tate pairings [45] on elliptic curves. The group that possesses such a map $\hat{e}$ is called a bilinear group, on which the Decisional Diffie-Hellman (DDH) problem is easy [46] to solve while the Computational Diffie-Hellman (CDH) problem is believed hard [47]. For example, given $P, aP, bP, cP \in \mathbb{G}$ and any $a, b, c \in \mathbb{Z}_q^*$, there exists an efficient algorithm to determine whether $ab = c \bmod q$ by checking $\hat{e}(aP, bP) \stackrel{?}{=} \hat{e}(P, cP)$, while there exists no algorithm that can compute $abP \in \mathbb{G}$ with non-negligible probability within polynomial time.

4.2.5 Batch Verification

With the pervasiveness of telecommunication applications, the demand and requirement on authentication for communication security become more stringent. The delay caused by verification of a bulk of signatures may dramatically impede transmission throughput and impair the system applicability. In order to speed up the process of verification, a batch verification scheme should be a good alternative solution since it can verify all the signatures received in a time window with rather short time compared to verify each signature one after the other. The general definition of batch verification is defined below.

Let ***Gen***, ***Sign***, ***Verify*** respectively represent the key generation, signature signing and verification algorithms of a signature scheme. ***Gen*** outputs a signing and verification key

pair (sk, pk). ***Sign***(sk, m) uses the signing key sk to compute a signature σ on the input message m. ***Verify***(pk, m, σ) outputs 1 if σ is a valid signature on m that is signed using sk corresponding to the public key pk, and 0 otherwise. Let P_1, P_2, . . . , P_n be n signers, whose corresponding key pairs are (sk_1, pk_1), (sk_2, pk_2), . . ., (sk_n, pk_n), respectively. Let Θ be a collection of n tuples of (P_i, m_i, σ_i). ***Batch***(Θ) is called a batch verification algorithm provided ***Batch***(Θ) =1 if and only if ***Verify***(pk_i, m_i, σ_i) outputs 1 for all i. The time consuming on ***Batch***(Θ) is much less than the sum of the time consuming on ***Verify***(pk_i, m_i, σ_i) n times.

The batch cryptography based on RSA was introduced by Fiat [48] in 1989. Some other batch signature schemes were proposed later [49–53]. The latest batch verification scheme proposed in [54] is based on the CL signature scheme [55], and is the first solution on batch verification without using random oracles, in which the computation efficiency can be significantly improved. With the batch verification scheme of [54], verifying a single signature requires 3 pairing operations, while verifying n signatures also takes 3 pairing operations instead of $3n$ pairing operations. In other words, the verification time of the dominant operation (i.e., paring) is independent of the number of signatures to verify. Therefore, the batch verification can dramatically decrease the time spent on verifying a large number of signatures, which can achieve much better scalability. In this chapter, we propose an efficient identity-based batch verification scheme based on the improved CL signature scheme in [54], and the proposed scheme is perfectly suitable for VANET communications.

4.2.6 Group Testing Technique

Group testing technique was first presented in World War II, and was motivated by the task to testing blood samplers of draftees to detect syphilis. In this application, each draftee was taken a blood sample. There were millions of draftees, and only a few thousand of draftees had the syphilis disease. A single test on the combination of multiple blood samples returned positive if at lease one sample was positive. A positive combination was divided and then further tested with other positive samples. On the other hand, a single test on the

combination of multiple samples returned negative if any of the samples was negative. A negative combination can save many individual tests. The group testing technique was to find an efficient strategy to combine blood samples aiming to find the positive blood samples with as the few number of tests as possible. In our application, the objective of group testing is to find invalid signatures with the minimal number of batch verification.

4.3 Batch Verification for Traffic Information Messages

In this section, we propose a novel Identity-based Batch Verification (IBV) scheme for traffic related message transmission. The proposed scheme includes the following four phases: the key generation and pre-distribution phase, the pseudo identity and private key generation phase, the message signing phase, and the batch verification phase.

4.3.1 Key Generation and Pre-distribution

In our scheme, TA is in charge of checking the vehicle's identity, and generating and pre-distributing the private master keys of the vehicles. Prior to the network deployment, the TA sets up the system parameters for each vehicle. It is worth noticing that the processes of key generation and pre-distribution in our scheme are only processed once and they do not participate in every message authentication and verification. The detail of key generation and pre-distribution are presented as follows.

- Let $\mathbb{G}$ be a cyclic additive group generated by P, $\mathbb{G}_T$ be a cyclic multiplicative group, and $\mathbb{G}$ and $\mathbb{G}_T$ have the same order q. Let $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map.
- The TA first randomly chooses $s_1, s_2 \in \mathbb{Z}_q^*$ as its two master keys, and computes $P_{pub1} = s_1 P, P_{pub2} = s_2 P$ as its public keys. These two master keys of the TA are then loaded in the vehicles' tamper-proof device.

- Each vehicle are preloaded with the public parameters $\{\mathbb{G}, \mathbb{G}_T, q, P, P_{pub1}, P_{pub2}\}$. In addition, the tamper-proof device of each vehicle is preloaded with the parameters $\{s_1, s_2\}$.

- To activate the tamper-proof device, each vehicle is assigned with a real identity, denoted as $RID \in \mathbb{G}$, and a password, denoted as *PWD*, where the *RID* uniquely identifies the vehicle, while the *PWD* is required in the authentication process by the tamper-proof device. Therefore, an adversary cannot take advantages of the tamper-proof device even if the vehicle is stolen.

4.3.2 Pseudo Identity Generation

To achieve privacy preservation, we exploit to use a tamper-proof device [9, 12, 56, 57], which is responsible for generating random pseudo identities and corresponding private keys based on identity-based cryptography [44]. The tamper-proof device is composed of three secure modules: an authentication module, a pseudo identity generation module, and a private key generation module as shown in Fig. 4.1, which are further described in details as follows.

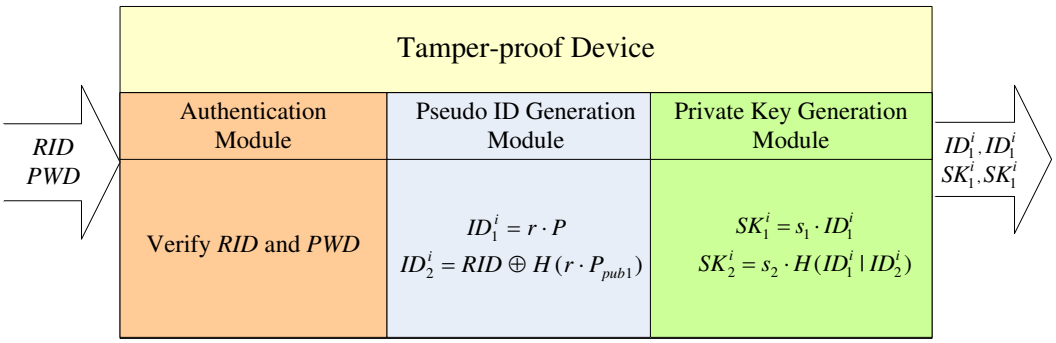


Figure 4.1: The tamper-proof device

Authentication module: The authentication module works as an access control mecha-

nism. A vehicle inputs its unique real identity *RID* and the password *PWD* to initiate the device, where the *PWD* can be the signature of the *RID* signed by the TA. If the *RID* and *PWD* successfully pass the verification of the authentication module, the *RID* is delivered to the next module, the pseudo identity generation module. Otherwise, the device denies providing services for the vehicle. Clearly, the authentication module enhances the security of the tamper-proof device since a malicious adversary cannot take advantages of it even though the tamper-proof device is physically held by the adversary.

Pseudo identity generation module: This module is responsible for generating a list of random pseudo identities from the authenticated *RID*. Each pseudo identity ID_i of vehicle V_i is composed of ID_1^i and ID_2^i. In this module, the ElGamal encryption algorithm [59] over the ECC [60] is employed to encrypt the *RID* as shown in Fig. 4.1. The two items of the cipher texts are taken as ID_1^i and ID_2^i, respectively. In other words, we have $ID_1^i = rP$, and $ID_2^i = RID \oplus H(rP_{pub1})$, where r is a random nonce. r is changed each time and guarantees the distinction of ID_1^i and ID_2^i for each pseudo *ID*. $\oplus$ is an Exclusive-OR (XOR) operation. Here, P and P_{pub1} are the public parameters preloaded by the TA. After the encryption, ID_1^i and ID_2^i are delivered to the private key generation module.

Private key generation module: In this module, identity-based cryptography [44] is employed. Since a pseudo identity has two parts (i.e., ID_1^i and ID_2^i), the private key generation module is responsible for computing a private key based on ID_1^i and ID_2^i. Thus, the resultant private key also contains two parts, which are denoted as SK_1^i and SK_2^i, respectively. As shown in Fig. 4.1, SK_1^i and SK_2^i are equal to $s_1 ID_1^i$ and $s_2 h(ID_1^i \mid ID_2^i)$, respectively.

Finally, vehicle V_i can obtain a list of pseudo identities ID_i=(ID_1^i, ID_2^i) along with the corresponding private keys SK_i=(SK_1^i, SK_2^i). Note that the pseudo identities and the private keys can be generated offline by the tamper-proof device; thus, no delay will be caused in the signing messages at the OBU side due to this process.

4.3.3 Message Signing

When vehicles are traveling on the road, they periodically broadcast traffic related information that could be extremely vital and life-critical information for neighboring vehicles. To ensure the integrity of the messages, each message sent by a vehicle should be signed and verified when being received. With the proposed IBV scheme, the message signing phase is presented as follows.

1. A vehicle, denoted by V_i, first generates the traffic related message denoted by M_i.
2. V_i picks a pseudo identity ID_i=(ID_1^i,ID_2^i) and the corresponding private key SK_i=(SK_1^i, SK_2^i) by way of the tamper-proof device.
3. With the private key SK_i=(SK_1^i,SK_2^i), V_i can compute the signature σ_i of the message M_i, where

$$\sigma_i \quad = \quad SK_1^i + H(M_i)SK_2^i \; .$$

4. Subsequently, V_i broadcasts the final message $\langle ID_i, M_i, \sigma_i \rangle$ to its neighboring vehicles.
5. These steps are repeated every 100-300 ms according to the DSRC [1].

The signature of the proposed IBV scheme has the following merits. Firstly, the signature overhead is very low. Compared with the ECDSA signature scheme of IEEE1609.2 [8], which is the current standard for VANETs, the length of a signature in the IBV scheme is a half of that of the ECDSA, e.g., $|\sigma_i| = 225$ bits ≈ 29 bytes.[1] However, the IBV scheme does not need any signature certificate to be sent along with the message due to the adoption of identity-based cryptography; instead, only a short-length pseudo identity is sent, which is of a length 58 bytes, i.e., $|ID_i| = |ID_1^i| + |ID_2^i| = 58$ bytes. In contrast, the ECDSA scheme

[1] If IBV uses an MNT curve [61] with 224-bit q, where the bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is asymmetric, $\mathbb{G}_1 \neq \mathbb{G}_2$, and elements in $\mathbb{G}_1$ are 224+1 bits long. The one extra bit indicates the coordinate (x-coordinate or y-coordinate) of a point in an elliptic curve.

has to incorporate a certificate in the message, which is 125 bytes long in the case of using the certificate presented in IEEE 1609.2 Standard [8]. We will further compare our proposed IBV scheme with the ECDSA scheme in terms of the communication overhead in Section V.

Secondly, from the perspective of signing speed, the proposed IBV scheme does not add any extra signature generation delay compared with that in ECDSA, where both of them need two multiplication operations on an elliptic curve. At last, the signature of the IBV scheme does not leak any real identity information of the vehicle because a pseudo identity is used in the scheme. Furthermore, since all the messages are signed with different pseudo identities, thus none of the two messages can be connected to a single vehicle with the IBV signature scheme, which is expected to successfully address the issue of privacy preservation in VANETs.

4.3.4 Message Batch Verification

Based on the network architecture as described in Section II, once a vehicle receives traffic related messages from other vehicles, the vehicle has to verify the signatures of the messages to ensure that the corresponding vehicles are not attempting to impersonate any other legitimate vehicles or disseminating bogus messages, which may result in tremendous impairment. For ease of presentation, we first introduce the single signature verification process, followed by the presentation on the batch verification of multiple signatures signed by distinct vehicles on different messages.

Single signature verification: Given the system public parameters $\{\mathbb{G}, \mathbb{G}_T, q, P, P_{pub1}, P_{pub2}\}$ assigned by the TA and the message $\langle ID_i, M_i, \sigma_i \rangle$ sent by the vehicle V_i, the signature σ_i is valid if $\hat{e}(\sigma_i, P) = \hat{e}(ID_1^i, P_{pub1}) \cdot \hat{e}(H(M_i)h(ID_1^i \mid ID_2^i), P_{pub2})$,as verified below.

$$
\begin{aligned}
& \hat{e}(\sigma_i, P) \\
= \ & \hat{e}(\boldsymbol{SK}_1^i + H(M_i)\boldsymbol{SK}_2^i, P) \\
= \ & \hat{e}(\boldsymbol{SK}_1^i, P)\hat{e}(H(M_i)\boldsymbol{SK}_2^i, P) \\
= \ & \hat{e}(s_1\boldsymbol{ID}_1^i, P)\hat{e}(H(M_i)s_2 h(\boldsymbol{ID}_1^i \mid \boldsymbol{ID}_2^i), P) \\
= \ & \hat{e}(\boldsymbol{ID}_1^i, s_1 P)\hat{e}(H(M_i)h(\boldsymbol{ID}_1^i \mid \boldsymbol{ID}_2^i), s_2 P) \\
= \ & \hat{e}(\boldsymbol{ID}_1^i, P_{pub1})\hat{e}(H(M_i)h(\boldsymbol{ID}_1^i \mid \boldsymbol{ID}_2^i), P_{pub2})
\end{aligned}
$$

Therefore, the computation cost by a vehicle for verifying a single signature is dominantly comprised of three pairing operations, one multiplication, one MapToPoint hash [47]. Note that the computation cost of a pairing operation is much higher than the cost of a multiplication and a MapToPoint hash operation.

Batch verification: Given n distinct messages denoted as $\langle \boldsymbol{ID}_1, M_1, \sigma_1 \rangle$, $\langle \boldsymbol{ID}_2, M_2, \sigma_2 \rangle$, ..., $\langle \boldsymbol{ID}_n, M_n, \sigma_n \rangle$, respectively, which are sent by n distinct vehicles denoted as V_1, V_2, ..., V_n, all signatures, denoted as σ_1, σ_2, ..., σ_n, are valid if $\hat{e}(\sum_{i=1}^n \sigma_i, P) = \hat{e}(\sum_{i=1}^n \boldsymbol{ID}_1^i, P_{pub1}) \cdot \hat{e}(\sum_{i=1}^n H(M_i)h(\boldsymbol{ID}_1^i \mid \boldsymbol{ID}_2^i), P_{pub2})$. Let $\boldsymbol{HID}_i$ denote $h(\boldsymbol{ID}_1^i \mid \boldsymbol{ID}_2^i)$. This batch verification equation follows since

$$
\begin{aligned}
& \hat{e}\left(\sum_{i=1}^n \sigma_i, P\right) \\
= \ & \hat{e}\left(\sum_{i=1}^n (\boldsymbol{SK}_1^i + H(M_i)\boldsymbol{SK}_2^i), P\right) \\
= \ & \hat{e}\left(\sum_{i=1}^n \boldsymbol{SK}_1^i, P\right)\hat{e}\left(\sum_{i=1}^n H(M_i)\boldsymbol{SK}_2^i, P\right) \\
= \ & \hat{e}\left(\sum_{i=1}^n s_1\boldsymbol{ID}_1^i, P\right)\hat{e}\left(\sum_{i=1}^n s_2 H(M_i)\boldsymbol{HID}_i, P\right) \\
= \ & \hat{e}\left(\sum_{i=1}^n \boldsymbol{ID}_1^i, s_1 P\right)\hat{e}\left(\sum_{i=1}^n H(M_i)\boldsymbol{HID}_i, s_2 P\right) \\
= \ & \hat{e}\left(\sum_{i=1}^n \boldsymbol{ID}_1^i, P_{pub1}\right)\hat{e}\left(\sum_{i=1}^n H(M_i)\boldsymbol{HID}_i, P_{pub2}\right).
\end{aligned}
$$

Thus, this batch verification can dramatically reduce the verification delay, particularly when

verifying a large number of signatures. From the above batch verification equation, the computation cost that a vehicle spends on verifying n signatures is comprised of 3 pairing, n multiplication, n MapToPoint hash, $3n$ addition, and n one-way hash operations. Among these operations, the pairing operation is relatively time consuming. However, the batch verification scheme uses the constant number (i.e., 3) of pairing operations regardless of the size of the batch. Thus, the time for a vehicle to verify a large number of signatures sent by the surrounding vehicles can be dramatically reduced, which can apparently reduce the message loss ratio due to the potential bottleneck of signature verification for vehicles.

Another advantage of IBV is that it can aggregate multiple signatures as one signature. This promising feature is not directly used in our safety related application, but it can be used to multi-hop non-safety related applications [25] to reduce communication overhead in VANETs. In our scheme, the aggregate signature is equal to $\sum_{i=1}^{n} \sigma_i$, given n distinct signatures, σ_1, σ_2, ..., σ_n. Further more, compared with BLS [47], our scheme does not require that n distinct messages have to be sent from the same sender.

4.4 Security Analysis

In this section, we analyze the security of the proposed batch verification scheme in terms of the following three aspects: the message authentication, the user identity privacy preservation, and the traceability by the TA.

- *Message authentication.* The message authentication is one of the basic security requirements in vehicular communications. In the proposed IBV scheme, the signature $\sigma_i = SK_1^i + H(M)SK_2^i$ is actually a one-time identity-based signature. Without knowing the private key SK_1^i and SK_2^i, it is infeasible to forge a valid signature. Because of the NP-hard computation complexity of Diffie-Hellman problem in $\mathbb{G}$, it is difficult to derive the private keys SK_1^i and SK_2^i from ID_1^i, P_{pub1}, P, and $h(ID_1^i \mid ID_2^i)$. At the same time, because $\sigma_i = SK_1^i + H(M)SK_2^i$ is a Diophantine equation, by only knowing σ and

$H(M)$, it is still difficult to get the private keys SK_1^i and SK_2^i. Therefore, the one-time identity-based signature is unforgeable, and the property of message authentication is achieved.

- *Identity privacy preserving.* In the proposed scheme, the real identity *RID* of V_i is converted into two random pseudo identities ID_1^i and ID_2^i, where $ID_1^i = rP$ and $ID_2^i = RID \oplus H(rP_{pub})$ for unknown r. Note that the pseudo identity pair (ID_1^i, ID_2^i) is actually an ElGamal-type ciphertext, which is semanticly secure under the chosen plaintext attacks. Therefore, without knowing the *master-key* (s_1, s_2), it is infeasible for anyone to tell the real identity from the pseudo identity pair. Also, the linkability does not exist because the pseudo identities (ID_1^i, ID_2^i) in each signature instance is distinct. Therefore, the identity privacy preservation can be guaranteed.

- *Traceability.* Given the pseudo identity pair ID_1^i and ID_2^i, only the TA, given the *master-key* (s_1, s_2), can trace the real identity of the vehicle by computing $ID_2^i \oplus H(s_1 ID_1^i) = RID \oplus H(rP_{pub}) \oplus H(s_1 rP) = RID$. Therefore, once a signature is in dispute, the TA has the ability to trace the vehicle from the disputed message, in which the traceability can be well satisfied.

4.5 False Signature Detection with Group Testing Technique

It is clear that the proposed batch verification scheme can significantly accelerate the overall signature verification when no false signature exists. When an error was found during the batch verification, the false signatures should be identified in the batch, and it can be simply done by sequentially verifying each signature using a single batch verification algorithm. However, sequential verification obviously causes long delay, particularly when an attacker who attempts to ruin the batch verification periodically sends a few invalid signatures. Note that an attacker can use invalid identities when sending each message in order to prevent the

TA from tracing its real identity. To improve the efficiency of false signature identification, this chapter investigates group testing techniques for this problem.

4.5.1 Employment of Group Testing Algorithm

The group testing technique, as presented in Chapter 4.2.6, was to find an efficient strategy to combine blood samples in the World War II, aiming to identify positive blood samples with as few number of tests as possible. In our application, the objective of group testing is to find invalid signatures with the minimal number of batch verification.

The task of false signature identification from a batch containing at least one false signature (or termed a "bad batch" in the following context) is formulated as a group testing problem. The batch verification equation in Chapter 4.3.4 takes 3 pairings and n operations of $H(M) \cdot \mathit{HID}$ to verify n signatures. Each of $H(M) \cdot \mathit{HID}$ takes a MapToPoint hash and a multiplication. It is worth noting that the MapToPoint hash and multiplication can be pre-computed and stored in memory for reuse. Once a batch verification on n signatures is launched, n operations of $H(M) \cdot \mathit{HID}$ are performed. If later a batch verification on any subset of the n signatures is needed, the MapToPoint hash and a multiplication on $H(M) \cdot \mathit{HID}$ do not need to be computed again. In this case, only 3 pairing computations are needed for a batch verification (test), and the time of testing the validity of multiple signatures is equal to that of testing a single signature.

Many constructions for adaptive combinatorial group testing were reported in the literature. However, it is difficult to find an optimal algorithm of group testing for a general purpose because the computational complexity of group testing has not been determined [62]. Adaptive group testing algorithms for finding invalid signatures were summarized in [63], and can be generally divided into the following four types: individual testing, binary search, generalized binary splitting (GBS) [62], Li's s-stage [62]. For each algorithm, the number of tests in the worse case is summarized in Table 4.1. Throughout this section, let n denote the number of signatures to be verified in a batch, and d be the upper bound on the estimated

number of invalid signatures.

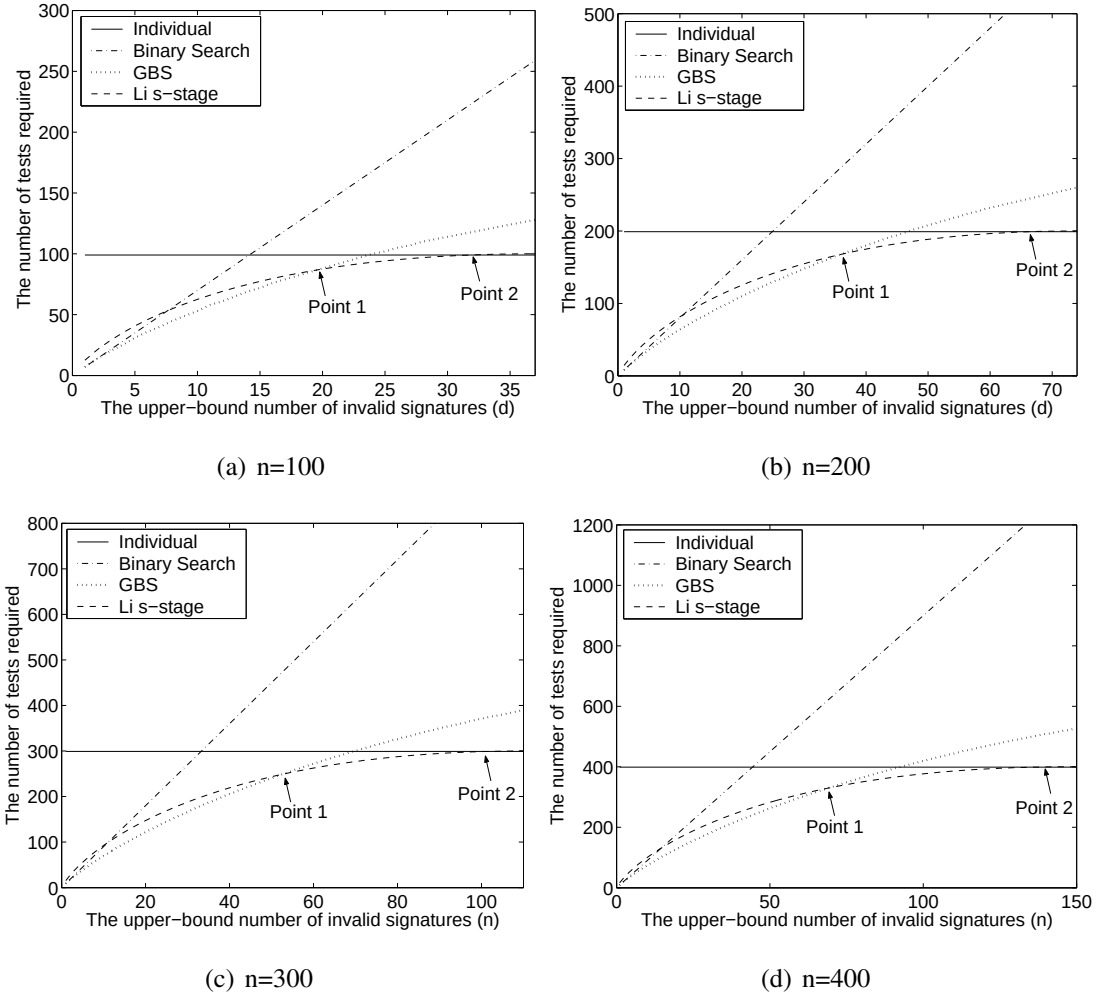
Table 4.1: Different adaptive group testing algorithm comparison [63]

Algorithm	Tests (worst case)
Individual Testing :	$n-1$
Binary Search :	$d\lceil\log(n)\rceil$
Generalized Binary Splitting :	$d-1+\lceil\log(\binom{n}{d})\rceil$
Li's s-stage :	$\frac{e}{\log(e)}\cdot d\cdot\log(\frac{n}{d})$

It is a challenging task to find an appropriate function due to the complexity of the functions $d-1+\lceil\log(\binom{n}{d})\rceil$ and $\frac{e}{\log(e)}\cdot d\cdot\log(\frac{n}{d})$. Moreover, the optimal algorithm depends on the values of n and d. Nevertheless, it is convenient to analyze the values of the above functions when a parameter (i.e., d) changes and the other parameter (i.e., n) is fixed. Fig. 4.2 shows the number of required tests (batch verifications) as d changes and n is fixed. In Fig. 4.2(a)-(d), n is equal to 100, 200, 300, and 400 respectively. It is clear that the function of Li's s-stage and the function of GBS always meet at a certain point, as represented as Point 1 in Fig. 4.2. When d is less than the *x* coordinate[1] of Point 1, GBS always has the optimal (minimal) function value. In addition, the function of Li's s-stage and the function of Individual testing always meet at another point that is represented as Point 2. When d is less than Point 1 and larger than Point 2, Li's s-stage always has the optimal function value. When d is larger than Point 2, the Individual testing always has the optimal function value.

In Fig. 4.2, n is only set to four values, i.e., 100, 200, 300, 400. For better analyzing the relationship between Point 1 (Point 2) and n, more value of n are selected. Given each n, the values of Point 1 and Point 2 are computed, as they are computed in Fig. 4.2(a)-(d). As such, a set of values of Point 1 and Point 2 can be obtained. Fig. 4.3(a) shows the set of the

[1]To ease our presentation, Point 1 and Point 2 respectively represent the *x* coordinate of Point 1 and *x* coordinate of Point 2 throughout this paper.



(a) n=100

(b) n=200

(c) n=300

(d) n=400

Figure 4.2: Show the number of tests required changing as increasing the upper bound on the number of invalid signatures (d) in four scenarios, where the number of signatures (n) equals 100, 200, 300, and 400, respectively.

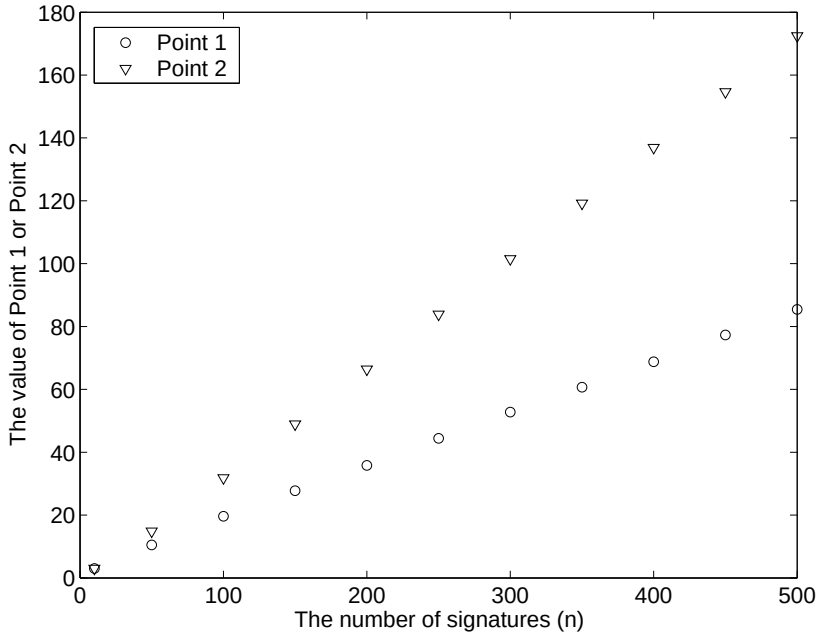
values of Point 1 and Point 2 given different values of n. As we can see, Point 1 and Point 2 increase linearly as n increases. Thus, it is reasonable to use two linear functions to represent the Point 1 set and Point 2 set. Fig. 4.3(b) shows the two fitting functions: the function $y_1 = 0.17n + 1.31$ fits the Point 1 set; the function $y_2 = 0.34n + 0.44$ fits the Point 2 set. The two lines divide the plane of Fig. 4.3(b) into three areas. Each area represents a desired group testing algorithm to be used. Therefore, given n and d, an optimal group testing algorithm can be selected. For example, given $n = 200$ and $d = 10$, the point (200,10) is in the Area 3 in Fig. 4.3(b). In this case, GBS is the optimal group testing algorithm. Given $n = 300$ and $d = 60$, the point (300,60) is in the Area 2 in Fig. 4.3(b). In this case, Li's s-stage is the optimal group testing algorithm. Given $n = 100$ and $d = 60$, the point (100,60) is in the Area 1 in Fig. 4.3(b). In this case, the Individual testing is the optimal group testing algorithm. The more generalized solution to choose the optimal algorithm is given below in accordance with the values of d and n.

$$\begin{cases} \text{GBS}, & d \leq 0.17n + 1.31 \\ \text{Li's s-stage}, & 0.34n + 0.44 \geq d > 0.17n + 1.31 \\ \text{Individual testing}, & d > 0.34n + 0.44 \end{cases}$$

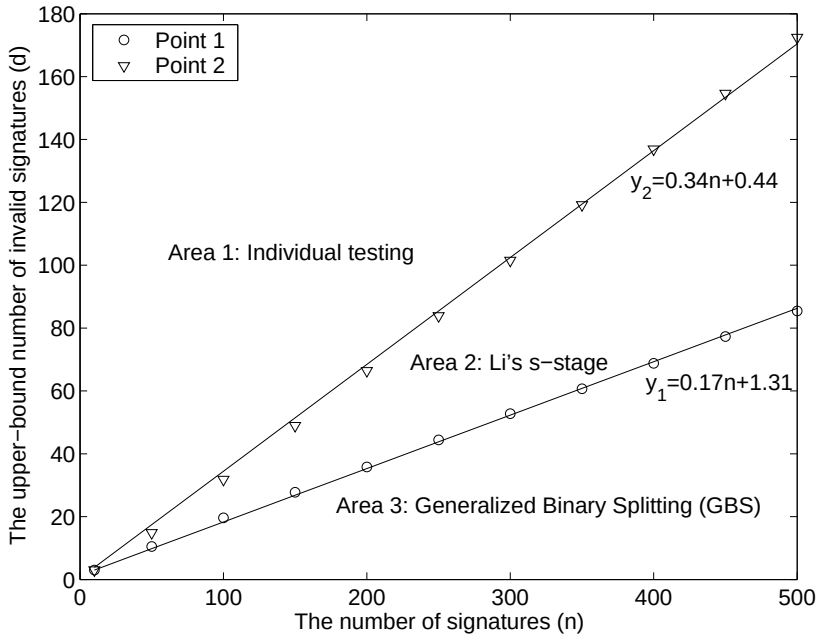
4.5.2 Generalized Binary Splitting

For batch verification, an attacker needs to send only a few number (at least one) of invalid signatures to launch a DoS attack that makes normal vehicles do group testing to find invalid signatures. In this case, the value of d that has been analyzed in the previous section is small. From the analysis in the previous section, we know that when d is small, GBS is the optimal group testing algorithm.

The GBS algorithm [62] is adopted in our scheme and presented in Algorithm 1. To use Algorithm 1, we need to estimate d, the upper bound on the number of invalid signatures. In VANETs, vehicles send the traffic related messages every 300 ms. In a normal case, the number of signatures that vehicles receive every 300 ms is equal to the number of their



(a) Show the values of Point 1 (Point 2) under different values of n



(b) Show the functions to fit the samples of Point 1 and Point 2

Figure 4.3: Show that the value sets of Point 1, Point 2 and the corresponding fitting functions.

Algorithm 1: Generalized binary splitting algorithm

Input: n signatures, where the estimated number of invalid signatures is not more than d

Result: Find out all invalid signatures

if $n \leq 2d - 2$ **then**
    a vehicle tests the n items individually;
    the group testing is done and return.
else
    compute $l = n - d + 1$, and $\alpha = \lfloor \log(l/d) \rfloor$.
end
Test a group of size 2^{α} signatures.
if *the outcome is negative* **then**
    the group of 2^{α} signatures are identified as good.
    set $n = n - 2^{\alpha}$, and go to Step 1.
else
    use binary search to identify 1 invalid signature, and an unspecified number, say m, of valid signatures.
    set $n = n - 1 - m$, $d = d - 1$, and go to Step 1.
end

neighbors, which can assist in estimating d. Such an estimating work belongs to a category of intrusion detection, and many related work [64, 65] has been conducted. For example, based on a number of ds in previous time periods, a Markov chain [66] can be used to compute the distribution of d and estimate d in the upcoming time period. We adopt the existing solution, and estimating d is beyond the scope of this thesis.

4.6 Performance Evaluation

In this section, we evaluate the performance of the IBV scheme in terms of verification delay and transmission overhead. Since the proposed scheme focuses on the signature verification process for V2V, we only consider the V2V communication in this section. In the following evaluation, we assume all the vehicles can communicate directly with each other's communication range.

4.6.1 Verification Delay

We define and compute the time cost of the cryptographic operations required in each verification by the proposed IBV scheme. Let T_{mul} denote the time to perform one point multiplication over an elliptic curve, T_{par} denote the time to perform a pairing operation, T_{mtp} denote the time to perform a MapToPoint hash function, T_{inv} denote the time to perform an inverse operation in a finite field, and T_{hash} denote the time to perform a hash function, e.g., SHA-1. Since T_{mul} and T_{par} dominate the speed of a signature verification, we only consider these operations and neglect other operations such as hash and inverse operations. We adopt the experiment in [67], which observes processing time for an MNT curve [61] of embedding degree $k = 6$ and 160-bit q, running on an Intel Pentium IV 3.0 GHZ machine. The following results are obtained: T_{mul} is 0.6 ms and T_{par} is 4.5 ms.

Next, we compare the proposed IBV scheme with ECDSA and BLS in terms of the verification delay. Here, the ECDSA scheme is the signature algorithm adopted by IEEE1609.2 standard [8], while BLS is a short signature scheme, which can also be used to perform signature aggregation. Table 4.2 shows the combination of the dominant operations of the three signature schemes in terms of verifying a single signature and n signatures, respectively. From the batch verification equation in Section III-V, we observe that the time to verify n distinct signatures is $3T_{\mathsf{par}} + nT_{\mathsf{mtp}} + nT_{\mathsf{mul}}$. According to [69], with BLS, the time spent on verifying n signatures is equal to $(n + 1)T_{\mathsf{par}} + nT_{\mathsf{mtp}}$. Verifying a ECDSA signature

requires $2T_{\mathsf{mul}} + T_{\mathsf{inv}}$. Note that using Shamir's trick [59] to compute $2T_{\mathsf{mul}}$ can be calculated faster than two point multiplications. Particularly, according to [70], $2T_{\mathsf{mul}}$ can be reduced to $1.2T_{\mathsf{mul}}$. Thus, verifying a ECDSA signature requires $1.2T_{\mathsf{mul}} + T_{\mathsf{inv}}$, and verifying distinct n signatures requires $1.2nT_{\mathsf{mul}} + nT_{\mathsf{inv}}$. Since ECDSA and BLS are not identity-based signature schemes, additional operations are needed to verify the public key's certificate. Thus, the overall message verification time for ECDSA and BLS should be doubled[2] as shown in Table 4.2.

Table 4.2: Comparisons of the speed of different verification schemes (ms)

	Verify a single signature	Verify n signatures
IBV :	$3T_{\mathsf{par}} + T_{\mathsf{mtp}} + T_{\mathsf{mul}}$	$3T_{\mathsf{par}} + nT_{\mathsf{mtp}} + nT_{\mathsf{mul}}$
BLS :	$4T_{\mathsf{par}} + 2T_{\mathsf{mtp}}$	$(2n+2)T_{\mathsf{par}} + 2nT_{\mathsf{mtp}}$
ECDSA :	$2.4T_{\mathsf{mul}} + 2T_{\mathsf{inv}}$	$2.4nT_{\mathsf{mul}} + 2nT_{\mathsf{inv}}$
RAISE :	$1.2T_{\mathsf{mul}} + T_{\mathsf{inv}} + T_{\mathsf{hash}}$	$1.2T_{\mathsf{mul}} + T_{\mathsf{inv}} + nT_{\mathsf{hash}}$

In our analysis, we assume the communication coverage of a vehicle is 300 m, and each vehicle periodically broadcasts a traffic related message every 300 ms. The traffic density is taken as the number (#) of vehicles within a vehicle's radiation range. The traffic density is also taken as the number of signatures to be verified in 300 ms. Since RAISE is an RSU-aided verification scheme, it is not in the same category with IBV, ECDSA, and BLS. Thus, here we only compare the performance by using IBV, ECDSA, and BLS to verify the signatures.

Fig. 4.4 shows the relationship between the verification delay and the number of vehicles within a certain vehicle's radiation range. The embedded small figure is a local zoom-in with the traffic load ranging from 12 to 20. From Fig. 4.4, we can observe that the verification delay by using BLS is always the largest no matter how many messages are received by

[2]With the IBV scheme, each message sent by a vehicle corresponds to a distinct identity. Thus, to achieve the same privacy level as the IBV's, the vehicle using the public key based schemes also needs to change an identity for each sending message. That is the reason why verification time for ECDSA and BLS should be doubled in this section.

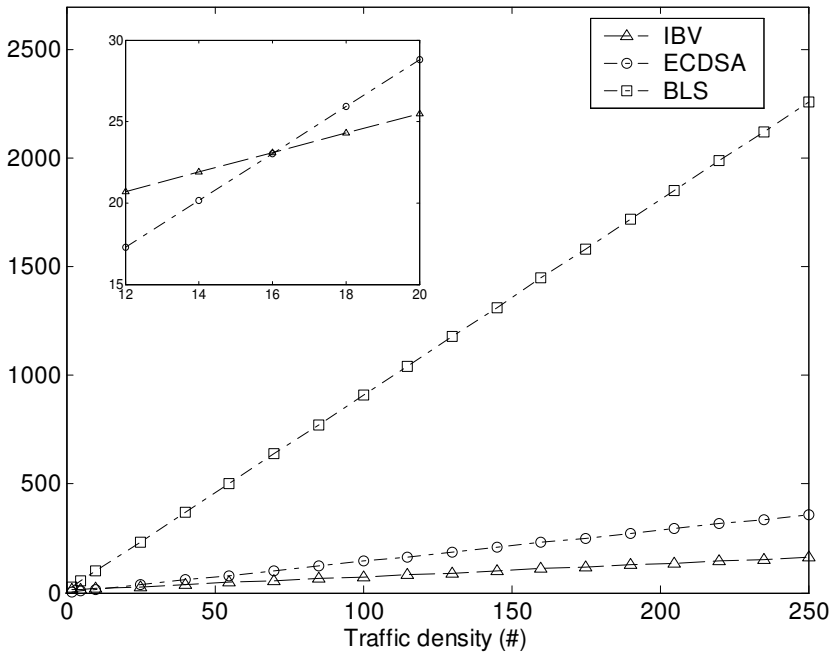


Figure 4.4: Verification delay *vs.* Traffic density

a vehicle. Another interesting result is that when the number of messages received within 300 ms is smaller than 16, the ECDSA scheme achieves the smallest message verification latency; however, when the number of messages is greater than 16, the IBV scheme yields much less verification latency. Clearly, when the traffic density is high, the IBV scheme can verify the largest number of signatures, which is observed to achieve the lowest message loss ratio when the traffic load increases.

We compare the message verification delay of these three schemes in terms of the ratio of the verification delays as shown in Fig. 4.5. We can see that the delay ratio between IBV and ECDSA approaches to a constant, which is approximately 0.651 when the number of messages in one interval is greater than 40. The delay ratio between IBV and BLS is approximately 0.112 when the number of messages is larger than 20. In other words, the speed of IBV is 34.9% faster than that of ECDSA, and is 88.8% faster than that of BLS.

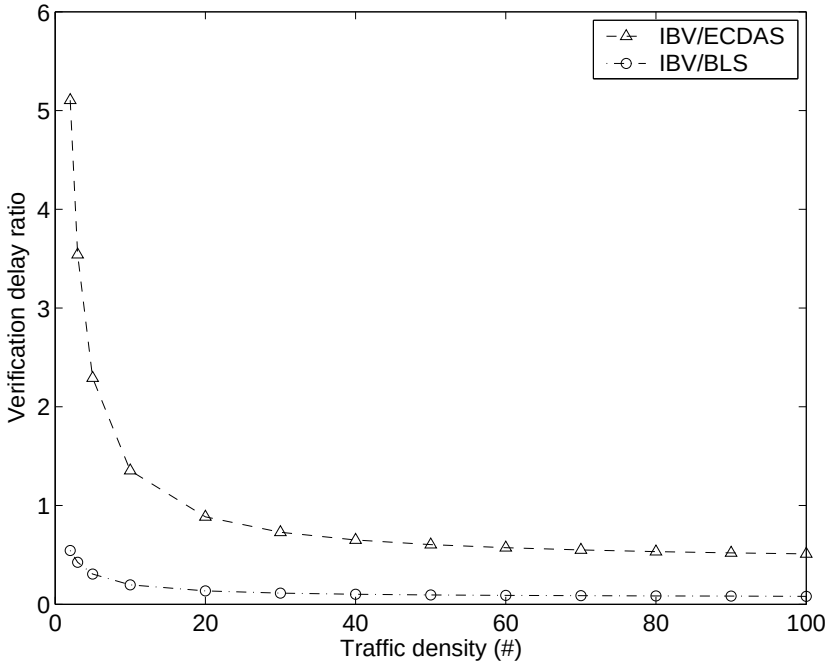


Figure 4.5: Verification delay ratio *vs.* Traffic density

4.6.2 Expected Verification Delay with False Signatures

In the previous section, the verification delay is evaluated in the situation where no false signature exists in each batch. In this section we will further analyze the verification delay where false signatures exist in a batch.

If the batch verification of IBV fails, the GBS group testing approach is used to find invalid signatures. In this case, more verification delay would occur. To properly quantify this delay, we define two probabilities. Let q denote the probability that a signature is invalid, and p denote the probability that a batch of n signatures has at least one invalid signature. Clearly, p also denotes the probability that a DoS attack happens, while $1 - p$ denotes the probability that no DoS attack happens. The relationship between p and q is presented below.

$$p = 1 - (1 - q)^n, \text{ and } q = 1 - (1 - p)^{1/n}. \tag{4.1}$$

Let T_{IBV} denote the verification delay that IBV is used to perform batch verification on

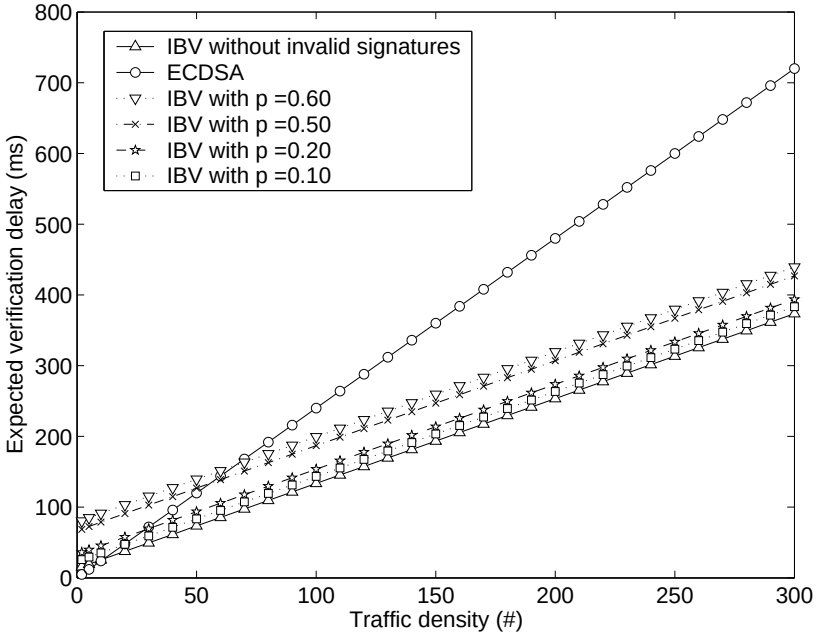


Figure 4.6: Expected verification delay *vs.* Traffic density

n signatures. As presented in Table 4.2, $T_{IBV} = 3T_{\mathsf{par}} + nT_{\mathsf{mtp}} + nT_{\mathsf{mul}}$. Let T_{GBS} denote the delay that the GBS group testing approach is used to find invalid signatures. As shown in Chapter 4.5.1, $T_{GBS} = (d - 1 + \lceil \log(\binom{n}{d}) \rceil) \cdot 3T_{par}$, where $d = n \cdot q$. Using the above parameters, we can derive the expected verification delay of IBV. We use E_{IBV} to denote the total expected verification delay, which is derived as below.

$$\begin{aligned} & E_{IBV} \\ &= T_{IBV} \cdot (1-p) + (T_{IBV} + T_{GBS}) \cdot p \\ &= T_{IBV} + T_{GBS} \cdot p \\ &= T_{IBV} + (d - 1 + \lceil \log(\tbinom{n}{d}) \rceil) \cdot 3T_{par} \cdot p \\ &= 3T_{\mathsf{par}} + nT_{\mathsf{mtp}} + nT_{\mathsf{mul}} + \{(1-(1-p)^{1/n}) \cdot n \\ &\quad -1 + \lceil \log(\tbinom{n}{\lceil (1-(1-p)^{1/n}) \cdot n \rceil}) \rceil \} \cdot 3T_{par} \cdot p \end{aligned} \tag{4.2}$$

Fig. 4.6 shows the relationship between E_{IBV} with different values of p and the number of vehicles (signatures) in a vehicle's communication range. From Fig. 4.6, we can observe

that given a fixed number of vehicles (signatures) E_{IBV} increases as p increases, but the increasing amount is not significant. If p is a small value, for example $p = 10\%$, E_{IBV} is close to the verification delay when there is no DoS attack. Compared with ECDSA, IBV with a large p still yields a lower expected verification delay especially in the scenario where the traffic density is high. As shown in Fig. 4.6, if the probability (p) that DoS happens equals to 50%, the IBV has lower verification delay than ECDSA has when the number of vehicles (signatures) is greater than 55. It is worth noticing that in reality the probability that DoS happens is far less than 50%. Therefore, we conclude that IBV is able to achieve low verification delay even though the DoS attack is taken into consideration.

4.6.3 Transmission Overhead

In this section, we compare the transmission overhead of IBV, ECDSA and BLS. The comparison is in terms of the following two aspects: the transmission overhead in V2V communication and the overhead in non-safety application. Here, the transmission overhead includes a signature and a certificate appended to the original message, while the message itself is not counted.

Table 4.3: Comparisons of transmission overhead of three schemes (ms)

	Send a single message	Send n messages
IBV :	29+58 bytes	29+58n bytes
BLS :	29+125 bytes	29+125n bytes
ECDSA :	56+125 bytes	56n+125n bytes

If an elliptic curve with 224-bit q is used, the length of a signature for IBV and BLS is 29 bytes, and the length for ECDSA is 56 bytes. When we use BLS or ECDSA, a certificate must be transmitted along with a signature. If we use the certificate presented in IEEE 1609.2 Standard [8], which has 125 bytes in length, the total transmission overhead of the BLS and ECDSA scheme is 29+125 bytes and 56+125 bytes, respectively, as shown in Table

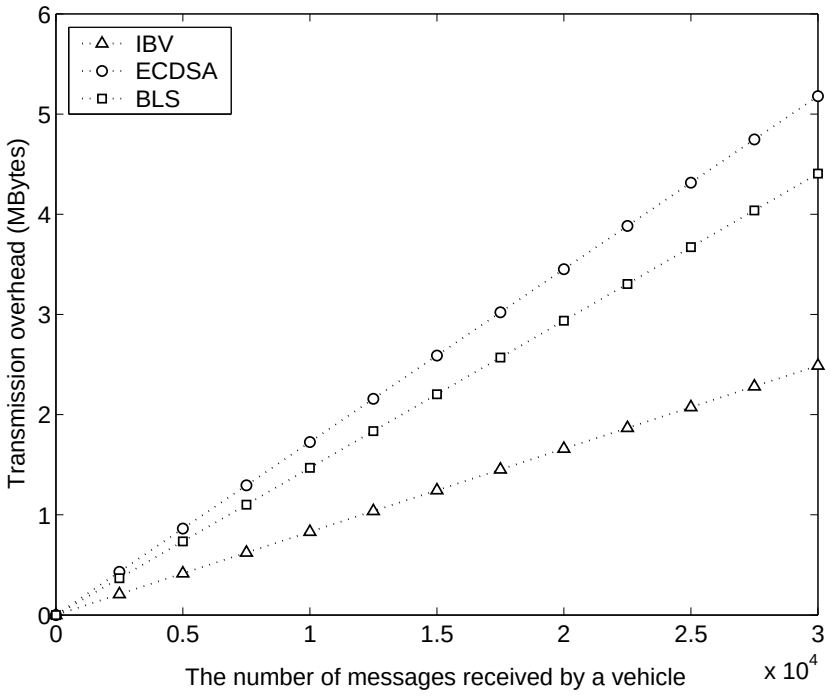


Figure 4.7: Transmission overhead *vs.* the number of messages received by a vehicle

4.3. Since the proposed IBV scheme is based on identity-based cryptography, only a short pseudo identity with 58 bytes is transmitted along with the original message. Thus, the total transmission overhead of IBV is 29+58 bytes as shown in Table 4.3.

Fig. 4.7 shows the relationship between the transmission overhead and the number of messages received by a vehicle. Clearly, as the number of messages increases, the transmission overhead increases linearly. The transmission overheads of ECDSA is the largest among the three schemes, and the transmission overhead of the IBV is much smaller than the other two. We can further observe that the transmission overhead of the IBV scheme is 56.5 percent of that of BLS and 48.1 percent of that of ECDSA. On the other hand, as shown in Fig. 4.7, within the observation window of 1 minute, when the number of messages increases up to 30000, IBV saves 1.92 Mbytes and 2.69 Mbytes of bandwidth compared with BLS and ECDSA, respectively. Here, 30000 corresponds to the number of messages sent by

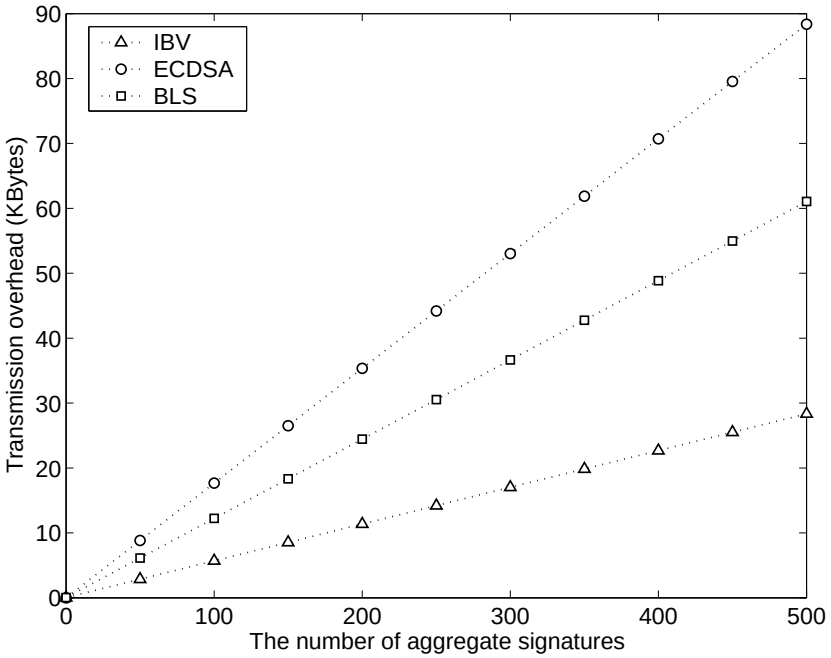


Figure 4.8: Transmission overhead *vs.* the number of aggregate signatures

150 vehicles in 1 minute.

In a delay and forwarding application [25], a non-safety related application in VANETs, vehicles in nature are commuters traveling in/among cities, and thus they can easily be used to carry, transmit, and forward messages. On the other hand, RSUs in nature are message switches which can be used to buffer messages and send messages to vehicles. For example, a message is to be sent from a location A to another location B through VANETs. The message is first carried by a vehicle driving from A towards B, and forwarded to an RSU before the vehicle drives away from the direction of B. Then, the RSU buffers the messages and forwards them to another vehicle that is driving towards the direction B. As such, the messages are repeatedly buffered, forwarded, and finally transmitted to B.

IBV can also be used in the delay and forwarding application for signature aggregation to reduce the communication overhead, which is another merit of IBV and should be emphasized here. In the delay and forwarding application, a vehicle assisting in forwarding

messages could aggregate multiple signatures. With IBV, given n distinct signatures, σ_1, σ_2, ..., σ_n, the aggregate signature is equal to $\sum_{i=1}^{n} \sigma_i$. It is worth noticing that unlike BLS, IBV does not require that n distinct messages have to be sent from the same sender. We compare the transmission overhead due to signature aggregation with the overhead without signature aggregation.

As shown in Table 4.3, let an RSU send n distinct signatures to a vehicle for forwarding. With the ECDSA scheme, the transmission overhead is in proportion to the number of signatures, namely (56+125)n bytes. In contrast, since BLS and IBV can aggregate signatures, only one aggregate signature is sent. In addition to the signatures, the BLS scheme needs to transmit a certificate with the length of 125 bytes for each message, while the IBV only needs to transmit a pseudo identity with the length of 58 bytes for each message. Thus, the total transmission overhead is 29+125n and 29+58n for the BLS and IBV, respectively. Fig. 4.8 shows the comparisons. The transmission overhead of all the schemes is proportional to the number of aggregate signatures. Compared with ECDSA, BLS is subject to lower transmission overhead; nonetheless, the advantage gained in BLS is not obvious because the certificate dominates the length of the overhead. On the other hand, since no certificate for each message is required in IBV, the advantage gained in the proposed scheme is obvious. From Fig. 4.8, we can see the transmission overhead of the IBV scheme is 46.4 percent of that by BLS and only 32.0 percent of that by ECDSA.

4.7 Summary

We have proposed a novel Identity-based Batch Verification (IBV) scheme for V2V communications in VANETs, which has been identified to be capable of meeting the most important and emerging design requirements on security and privacy preservation ever reported in the literatures. First, the proposed IBV scheme can significantly improve the system performance by fully taking advantages of verifying multiple message signatures at once instead of

the verification in a one-by-one manner. Second, the scheme has also addressed the identity privacy and traceability issues in vehicular networks, where the signature of a message is signed according to a pseudo identity pair and private keys that are generated by the tamper-proof device. Third, the IBV scheme enables TA to retrieve the real identity of a vehicle from any message signature, such that conditional privacy preservation can be achieved. Fourth, the scheme adopts group testing technique, which can efficiently to find invalid signatures from a batch of signatures. Finally, extensive analysis and evaluation have been conducted to demonstrate that the IBV scheme can achieve excellent operational efficiency for vehicular communications in terms of signature verification delay and communication overhead, in comparison with existing counterparts.

Chapter 5

Mitigating Signature Jamming Attacks against Message Authentication

5.1 Introduction

In the previous chapters, efficient message authentication schemes have been introduced to address security-related scalability issues in VANETs. With the proposed schemes, vehicles can verify traffic related messages in a fast way. Nevertheless, in our previous chapters, the normal case is only considered, where vehicles verify regular messages sent by legitimate vehicles, but the abnormal situation is not considered, where a sophisticated attacker may launch a denial of service (DoS) attack deliberately.

Attempting to launch a DoS attack, an attacker may send a significant number of messages with invalid digital signatures. In this case, the security units of the surrounding vehicles will be occupied by the extensive computation effort in verifying those invalid messages, which results in delay in verifying the other regular and legitimate messages. Since the vehicles cannot accept messages before these messages pass through the signature verification, such a delay further leads to loss of the legitimate messages. Particularly, with ECDSA for

example, a vehicle spends 3.87 ms[1] to process a single signature verification, a vehicle can verify about 70-80 messages in a 300 ms broadcasting cycle. Even using IBV, a vehicle can verify about 250-300 messages in a 300 ms broadcasting cycle. Attackers can easily crash the network when they massively floods messages with invalid signatures (e.g., the number of invalid signatures is far larger than 300) as a DoS attack. We specifically term such a DoS attack as *Signature Jamming Attack* (SJA). It is worth noticing that attackers launching a SJA could use false identities in order to prevent a TA from tracing their real identities. At this point, this fact increases the difficulty for us to thwart SJA.

The SJA falls in the category of DoS attacks. It is clear that studies on DoS have been conducted in wireless sensor networks (WSNs) and IEEE 802.11 based access networks [71–75]. Ning *et al.* [71] proposed a lightweight authentication scheme using cryptographic puzzles. Wang *et al.* [72] introduced a dynamic window scheme, in which each individual sensor makes a smart decision on whether to verify or forward signatures. The right choice can mitigate DoS attacks. Dong *et al.* [74] integrated the approaches of an authentication filter and cryptographic puzzles to address the DoS attack. Nevertheless, the characteristics of wireless sensor networks and conventional IEEE 802.11 access networks are different from those of vehicular networks. For example, a VANET has no limited power constraint, and it is generally hard for an attacker to compromise security materials in a vehicle. In addition, vehicles are more dynamic and mobile than sensors. Moreover, an important feature in VANETs is that the time interval of sending messages is fixed, which equals 300 ms. Due to these unique features, the above schemes in WSNs and 802.11 access networks cannot directly apply to VANETs. To thwart SJA, the TESLA [30] scheme for vehicular networks could be an alternative countermeasure to prevent vehicles from verifying a large number of invalid signatures. However, the TESLA scheme brings verification delay at a receiver side. Such a delay is unsuitable for safety-related applications in VANETs.

[1]The 224 bits ECDSA cryptographic delays are quoted from MIRACL cryptographic lib with the 3GHz Pentium IV system.

Motivated by the above observations, this chapter investigates an efficient countermeasure to the SJA. A novel puzzle-based scheme is proposed, which equips each vehicle with an authentication filter. With the proposed scheme, vehicles can exclude messages with invalid signatures in the pre-authentication stage with very low overhead before messages go into the regular signature verification process. The proposed signature filer is characterized by being free from any real-time verification delay at both sender and receiver sides, and can achieve the best scalability in presence of massive massages of false signatures. To achieve the best filtering effect, a hash recommendation mechanism is employed, which enables benign vehicles to associate with each other and cooperatively defend the SJA.

The rest of the chapter is organized as follows. Chapter 5.2 briefly introduces the preliminaries including our system model, attack model, hash chain, RSA encryption, and the puzzling mechanisms. Chapter 5.3 presents a puzzle-based message authentication scheme. Chapter 5.4 introduces the proposed hash recommendation mechanism to solidify the puzzle-based scheme. Chapter 5.5 analyzes the performance of the proposed schemes through extensive experiments and simulations. Finally, the summary is given in Chapter 5.6.

5.2 Systems and Security Preliminaries

5.2.1 System Model

We consider a VANET where each vehicle periodically broadcasts traffic related messages to its one-hop neighbors. The message broadcast period is 300 ms in accordance with DSRC [1]. The local time on each vehicle is synchronized as introduced in [71, 76]. Vehicles communicate with each other based on IEEE 802.11p protocol, and they have the same communication range. Public Key Infrastructure (PKI) is adopted, where each vehicle has a public/private key pair and the corresponding PKC. Without loss of generality, vehicles could have multiple public/private key pairs [12] or could use identity based signature scheme for any specific reason and design premise, which, nevertheless, is transparent to the

design of our scheme in this chapter.

5.2.2 Attack Model

In general, there are two types of attacks on a cryptosystem: passive attack and active attack. For a passive attack, an attacker cannot interact with any of the parties involved, while just attempting to break the system by observing and analyzing data. For an active attack, an attacker attempts to break security by altering or adding information communicated between parties. We considers the active attack in this chapter, where the attacker is sitting among vehicles and attempts to send a large number of bogus messages with invalid signatures. The purpose of the attacker is to delay normal message verification of the other vehicles by flooding messages with invalid signatures. The attacker could be either stationary or moving, and could be a normal vehicle with its own private key and public key. In order to escape from being traced by the trust authority, the attacker that launches the SJA does not use its own public key to expose its real identity.

SJA is different from a pure Dummy Message Jamming Attack (DMJA). Attackers using DMJA attempt to occupy the whole wireless bandwidth and interfere normal message transmission. The number of sent messages in DMJA is far larger than the number of sent messages in SJA. Therefore, SJA is a weaker attack compared with DMJA.

5.2.3 Design Objective

Our design objective in this chapter is to provide countermeasure of the SJA in the message authentication process for vehicular communications. Specifically, based on the aforementioned attack model, we have the following design objectives.

- *Mitigation of SJA*: It is well-known that DoS is notorious and extremely hard to prevent. Due to the nature of DoS, the damage caused by the SJA cannot be 100% avoided. Nonetheless, our objective is to mitigate the damage capability of SJA as much as

possible.

- *Minimized authentication delay*: Safety-related traffic messages sometimes bear critical information and cannot be delay and/or dropped.
- *Authenticity*: It is to ensure that the received messages are indeed sent from original message senders.
- *Integrity*: It is to guarantee that an attacker in the middle cannot modify or pollute messages and the corresponding authentication credentials, e.g., signatures.

5.2.4 Hash Chain

Hash chain [77] in an essential cryptography approach for network security in many applications. It is widely used for secure password authentication [79], system micropayment [79], stream data authentication [80], and secure data forwarding [30]. The computation of a hash chain is based on a one-way hash function. Let $H(.)$ denote a one-way hash function, where it is easy to compute $H(m)$ given a message m with an arbitrary length, but hard to derive m given $H(m)$. The generation of a hash chain is shown Fig. 5.1.

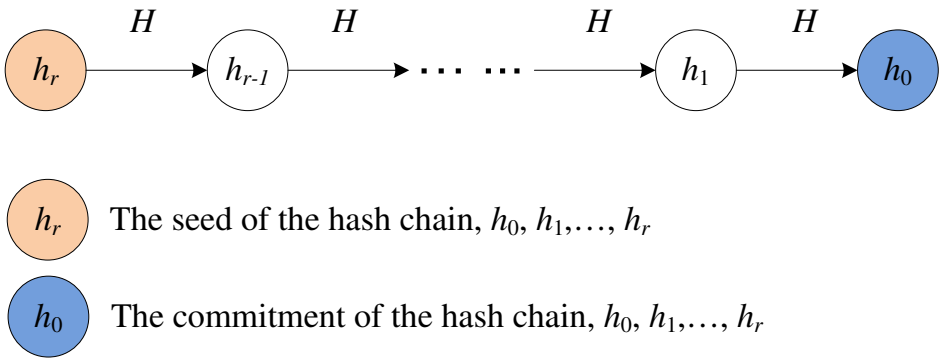


Figure 5.1: A one way hash Chain

First of all, we select is a random number s, and let $h_r = s$. Then, we compute $h_i =$

$H(h_{i+1})$, $i = 0, 1, 2, \ldots, r-1$. h_0 is called *tip* or *commitment* of the hash chain. One can apply the above hash chain by revealing the chain elements in the opposite order. Hash chain has the following properties: (1) given x, it is computationally hard to find $y \neq x$ such that $H(x) = H(y)$; (2) it is computationally hard to find a pair of messages, x and y, such that $H(x) = H(y)$; (3) given h_i, it is easy to compute h_{i-1}, but given h_{i-1}, it is computationally hard to find h_i, $i = 1, ..., r$. Since the one way hash chain has extremely low computation overhead, recently it has been widely used for the design of many security protocols in wireless networks.

5.2.5 Hash-based Puzzle

A puzzle is a question that needs to take some time to solve. Ning et al. [71] introduced a hash-based puzzle scheme to mitigate DoS in wireless sensor network. A hash-based puzzle is based on computation of a hash function. Since the hardness of reverse computation of a hash function is determined by the length of the hash, the length of the hash determines the security of a hash function. Under the computation ability of state-of-the-art computers, a hash function with a length of 160 bits, such as SHA-1, could be generally taken as secure enough to defend a brute force attack. On the other hand, a hash with a shorter length can be more easily computed in the reverse direction. With the above knowledge, a hash-based puzzle scheme can be described as follows:

$$H(Q \mid S) = \underbrace{00...0}_{l \text{ bits}} \text{xx...x}.$$

where "00..0" presents l bits of "0", and "xx...x" presents any bit pattern. l is set in the range of 20-26 [71], which is much shorter than 160 bits as that taken by SHA-1 [58]. Q is called puzzle. Thus, given the puzzle Q, the solution to the puzzle is S such that $H(Q \mid S)$ satisfies the above pre-defined bit pattern. The computation time to solve a puzzle depends on the value of l.

5.3 Puzzle-based Message Authentication Scheme

5.3.1 Overview

Filter verification is a process of pre-authentication performed before signature verification, which is light-weight and must be faster than a regular signature verification process. The proposed pre-authentication mechanism can be incorporated with any PKI based signature scheme, while the study of PKI-based signature schemes is not in the scope of this chapter.

With the existing signature schemes for VANETs, a message has a common abstract format: $\langle m \mid \sigma \mid Cert \rangle$, where m is the message, σ is the signature on the message, and $Cert$ is the PKC. In the event that the ID-based signature is employed, $Cert$ can be omitted. As presented at the beginning of this chapter, the existing signature schemes cannot countermeasure the vicious effect due to SJA, which motivates the endeavor of the study in this chapter.

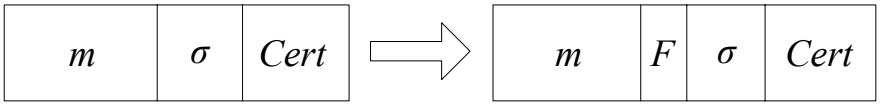


Figure 5.2: A new message format

To thwart SJA, the proposed signature scheme has one more parameter added to each message, thus the message format is $\langle m \mid F \mid \sigma \mid Cert \rangle$, as shown in Fig. 5.2. The parameter F is called *signature jamming attack filter*. With the proposed scheme, a vehicle that receives the message $\langle m \mid F \mid \sigma \mid Cert \rangle$ first verifies the filter F. If F is valid, the vehicle further follows the regular signature verification process to check σ. Otherwise, if F is invalid, the vehicle drops the message directly and hence no signature σ verification is needed. When the vehicle does not have enough information to verify F, it uses the public key in $Cert$ to verify σ. The filter F should be light-weighted such that the verification of F should be very fast. Therefore, such F can be used to quickly filter invalid messages at the first authentication step.

5.3.2 Core of proposed pre-authentication scheme

Although the idea of using a hash-based puzzle mechanism has been explored in [71] for wireless sensor networks, it is not suitable for VANETs due to the following two factors. First, VANETs have a highly dynamic network topology. Second, message verification for VANETs should not bring any delay at either sender side or receiver side. This chapter proposes a novel puzzle-based scheme that will be perfectly working in a highly dynamic environment of VANETs.

The sender off-line generates a hash chain $h_0, h_1, \ldots, h_r$, wherein $h_i = H(h_{i+1})$, $i = 0, 1, 2, \ldots r-1$, $h_r = s$ is a random number, and r is the length of the hash chain. In our scheme, each element of a hash chain is orderly attached with a message to be sent. The first sent message, denoted by m_0, and its credentials are presented as follows,

$$\textit{ID} \mid m_0 \mid 0 \mid h_0 \mid \{H(\textit{ID} \mid m_0 \mid 0 \mid h_0)\}_{sk} \mid Cert_{pk}.$$

where $\langle 0 \mid h_0 \rangle$ is the filter F, and $\{H(\textit{ID} \mid m_0 \mid 0 \mid h_0)\}_{sk}$ is the signature σ. Here, $\{m\}_{sk}$ represents signing m with the private key sk. *ID*, sk, and $Cert_{pk}$ denotes the sender's identity, private key, and PKC, respectively. As mentioned before, the proposed pre-authentication mechanism can incorporate any state-of-the-art PKI signature scheme, such as RSA, ECDSA, and IBV. Following the message is the index of the hash chain, where 0 denotes the first element of the used hash chain. We call the first message including h_0 as *commitment message* in the following context.

On the recipient side, the vehicle that receives the message first computes $H(\textit{ID} \mid m_0 \mid 0 \mid h_0)$, and then verifies the signature $\{H(\textit{ID} \mid m_0 \mid 0 \mid h_0)\}_{sk}$ with the sender's public key pk. If the signature is valid, the vehicle buffers *ID*, 0 and h_0, which will be used to authenticate the upcoming messages with the identity *ID*.

Compared with the first message, the subsequent messages at the sender have different contents as shown below.

$$\langle \textit{ID} \mid m_i \mid i \mid h_i \mid S \mid \sigma \mid Cert_{pk} \rangle,\ i = 1, 2, \ldots, r,$$

where $\langle i \mid h_i \mid S \rangle$ is the filter F, and σ equals $\{H(\textit{ID} \mid m_i \mid i \mid h_i \mid S)\}_{sk}$. For any message, $\textit{ID} \mid m_i \mid i \mid h_i \mid S$, $i \neq 0$, it must satisfy the following format.

$$H(\textit{ID} \mid m_i \mid i \mid h_i \mid S) = \underbrace{00...0}_{l \text{ bits}} \text{xx...x}.$$

Given m_i, the sender needs to compute S, which is the solution to the puzzle with the above format. h_i is the i-th element of the previously used hash chain, and is also called the *key* to the puzzle. In VANETs, the standard [1] requires each message to be sent every 300 ms, and thereby $i \mid h_i$ is sent every 300 ms accompanying with m_i as shown in Fig. 5.3. In our scheme, m_i includes the sending timestamp t_i.

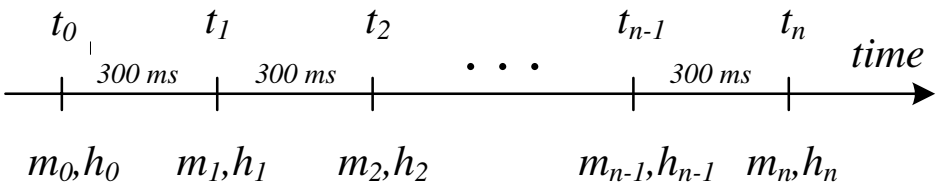


Figure 5.3: Each key is sent every 300 ms

Upon receiving $\textit{ID} \mid m_i \mid i \mid h_i \mid S$, the receiver first verifies the puzzle key h_i. Since the previous key h_{i-1} ($i > 0$) associated with *ID* has been buffered, the receiver can verify $h_{i-1} \stackrel{?}{=} H(h_i)$. If the receiver does not have h_{i-1} but has h_j ($0 \leq j < i-1$)[2], it can still verify h_i by checking $h_j \stackrel{?}{=} H^{(i-j)}(h_i)$, where $H^{(i)}(m) = H(H^{(i-1)}(m))$ and $H^{(1)}(m) = H(m)$. If the key h_i is invalid, the receiver just drops the message due to the fact that only the sender with h_j can know h_i. If the key h_i is valid, the receiver further verifies the solution S, which is valid one if $H(\textit{ID} \mid m_i \mid i \mid h_i \mid S)$ can satisfy the predefined bit format: l bits 0 followed by $160 - l$ bits "x" (x=1 or 0), if SHA1 is used. Here we called l as the *secure length* of the proposed filter.

[2]If the receiver still does not have h_j, it has to verify σ. A valid σ means a valid h_i. However, verifying σ is subject to SJA. We leave the solution introduction in Chapter 5.4.

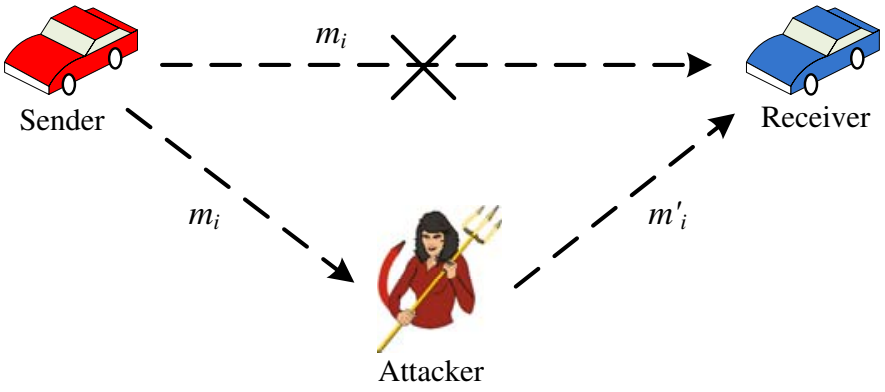


Figure 5.4: An example of the man in the middle attack

In addition to the above verification steps, m_i is dropped if the sending time t_i does not equal $t_j + 300 \times (i - j)$, where t_j is the timestamp in the previously received message m_j. Note that j is equal to $i - 1$ if the receiver did not lose the previous message from the sender. If t_i equals $t_j + 300 \times (i - j)$, the receiver further verifies whether the *time difference* δ between the sending time t_i and the instant time t that m_i is received is less than a threshold, denoted by Δ.

By manipulating the threshold Δ and the secure length l, we can mitigate the threat of Man-In-The-Middle (MITM) attack. In our scheme we make the threshold Δ equal 100 ms. We define the length of l such that the time on calculating the solution S to a puzzle is much larger than Δ. Suppose an attacker attempting to tamper with m_i in the middle as shown in Fig. 5.4. The attacker must compute a new solution S'_i to the puzzle after he receives $ID \mid m_i \mid i \mid h_i \mid S$. As such, the attacker has to spend more than Δ to get S'_i. However, while the receiver receives this forged message m'_i and S'_i, they will find out that the time difference $\delta = t - t_i$ is larger than Δ. Note that the attacker cannot modify t_i because the DSRC standard defines the sending time interval as 300 ms as shown in Fig. 5.3 and the local time on each vehicle is synchronized [71, 76]. As long as the receiver verifies m_i that contains the timestamp t_i, the receiver thus can predict the timestamps in the following received messages. Therefore, our scheme is secure against the MITM attack.

In a word, the proposed puzzle-based signature filter scheme can achieve a fast pre-authentication process, which needs only a few hash operations at the sender and receiver. Moreover, unlike the TESLA related scheme which is based on delayed authentication, with our scheme a receiver can verify each message immediately when it is received.

5.3.3 Offline Speedup Mechanism at Sender

Although the proposed puzzle-based filter scheme can effectively achieve desired fast pre-authentication, it brings delay at the sender side due to the computation of a puzzle solution S, which may take more than Δ. Since the calculation of S can be performed only when message m is ready, thus additional delay is incurred in sending each message. To reduce the delay, one approach is to shorten the secure length l so that the latency on computing S becomes shorter. Nonetheless, this is at the expense of weakened security protection where an adversary can compute S more quickly.

To deal with the aforementioned problem, we introduce a novel approach as an offline speedup mechanism to incorporate with the proposed puzzle-based filter scheme. Our design is motivated by the observation that the computation of the puzzle solution S should be kept independent of the message m such that the delay of computing S can be performed offline, wherein the delay at the sender can be completely removed. Our approach is characterized by using an RSA encryption mechanism with a small public key as presented in the following paragraphs.

Algorithm 2 presents the process of the authentication credential generation. Similar to the original pre-authentication scheme, a hash chain is first pre-generated, and each message orderly uses one hash chain element. In Step 2, the RSA encryption parameters are pre-defined. The message sender has its own secret parameters, denoted as p and q. The parameter n is a public parameter with a small length. Note that the RSA encryption does not require a strong security level, which will be further discussed in Chapter 5.5. In Step 3,

Algorithm 2: Filter Generation

Input: a message m_i, the element index of a hash chain i

Result: $\langle ID \mid m \mid i \mid h_i \mid A \mid B \mid C \mid D \rangle$

1 Generate a hash chain $h_0, h_1, \ldots, h_r$, such as $h_i = H(h_{i+1})$, $i = 0, 1, \ldots r-1$, and $h_r = s$, where s is a random number.

2 Choose two distinct large prime numbers p and q, and compute $n = p \cdot q$ and $\varphi(n) = (p-1)(q-1)$. Let e be equal to 3. Compute d such that $e \cdot d = 1 \bmod \varphi(n)$. d is a secure key and only known by the sender. Let $ID = n$.

3 Find a puzzle solution $S \in \mathbb{Z}_n^*$, where $\mathbb{Z}_n^* = \{\text{a is an integer}, a \in [0, n] \mid gcd(a, n) = 1\}$, such that

$$H(h_i \mid S) = \underbrace{00...0}_{l \text{ bits}} \text{xx...x}.$$

4 Compute S^d and save the tuple $\langle i, h_i, S, S^d \rangle$ in local database. Step 1-4 are done off-line.

5 Given a message m, find $a, b \in \mathbb{Z}_n^*$, i.e., a is coprime to n, such that $a \cdot H_n(m) + b = S^d \bmod n$, where $H_n(.)$ is a hash function mapping $\{0,1\}^*$ to an integer in $[0, n]$. a, b can be found in the following two steps. (i) Randomly select a number $a \in \mathbb{Z}_n^*$, i.e., a is coprime to n, and $a \neq 1$; (ii) Compute $b = S^d - a \cdot H_n(m) \bmod n$.

6 Compute $A = a^3 \bmod n$, $B = b^3 \bmod n$, $C = 3a^2b \bmod n$, $D = 3ab^2 \bmod n$ such that $A + B + C + D = a^3 + b^3 + 3a^2b + 3ab^2 = (a+b)^3 \bmod n$.

7 Output $\langle ID \mid m \mid i \mid h_i \mid A \mid B \mid C \mid D \rangle$.

the message sender pre-computes the puzzle solution S such that

$$H(h_i \mid S) = \underbrace{00...0}_{l \text{ bits}} \text{xx...x}.$$

Having the solution S, each vehicle also needs to pre-compute S^d mod n, which is used in Step 5 directly. Since this process is completely independent of the message, it can be performed offline. When a message m is generated, the sender vehicle finds $a, b \in \mathbb{Z}_n^*$, a is coprime to n such that $a \cdot H_n(m) + b = S^d$ mod n, where $H_n(.)$ is a hash function mapping $\{0,1\}^*$ to an integer in $[0, n]$. a, b can be found in the following two steps. (i) Select a number $a \in \mathbb{Z}_n^*$, a is coprime to n, and $a \neq 1$; (ii) Compute $b = S^d - a \cdot H_n(m)$ mod n. Then, vehicles compute a^3 mod n, b^3 mod n, $3a^2b$ mod n, $3ab^2$ mod n. For clarity of presentation, let $A = a^3$ mod n, $B = b^3$ mod n, $C = 3a^2b$ mod n, and $D = 3ab^2$ mod n. Notice that $a^3 + b^3 + 3a^2b + 3ab^2 = (a + b)^3$ mod n. Finally, vehicles broadcast the tuple $\langle \textit{ID} \mid m \mid i \mid h_i \mid A \mid B \mid C \mid D \mid \sigma \mid Cert_{pk} \rangle$ to their neighbor, where $\langle i \mid h_i \mid A \mid B \mid C \mid D \rangle$ is the filter F, and σ equals $\{H(\textit{ID} \mid m_i \mid i \mid h_i \mid A \mid B \mid C \mid D)\}_{sk}$.

With the above, the process of a filter generation can be done in a very efficient way since no delay at the sender side is caused due to the offline computation of each puzzle solution. The computation of A, B, C, D in Step 6 of Algorithm 2 only needs a few multiplication operations, and the computation time can be negligible especially when n is small.

Algorithm 3 presents the process of filter verification. Upon receiving the filter $F = \langle \textit{ID} \mid m \mid i \mid h_i \mid A \mid B \mid C \mid D \rangle$, the receiver verifies h_i and the timestamp within m. This process is exactly the same as the original filter design. If this step succeeds, the receiver compute $S = AH_n(m)^3 + B + CH_n(m)^2 + DH_n(m)$. This equation holds because

$$\begin{aligned} S &= S^{\varphi(n)} = S^{e \cdot d} = S^{3 \cdot d} \text{ mod } n \\ &= (S^d)^3 \text{mod } n = (aH_n(m) + b)^3 \text{ mod } n \\ &= a^3 H_n(m)^3 + b^3 + 3a^2 b H_n(m)^2 + 3ab^2 H_n(m) \text{ mod } n \\ &= AH_n(m)^3 + B + CH_n(m)^2 + DH_n(m) \text{ mod } n. \end{aligned}$$

Once the receiver has the puzzle solution S and the key h_i, it verifies whether $H(h_i \mid S)$ has the predefined format as shown in Step 3 of Algorithm 3. If this step succeeds, the

Algorithm 3: Filter Verification

Input: $\langle ID \mid m \mid i \mid h_i \mid A \mid B \mid C \mid D \rangle$

Result: 1, accepted; 0, denied

1 Verify t_i in m_i. If $t_i \neq t_j + 300 \cdot (i - j)$ ms, where t_j $(0 \leq j < i)$ is previously received in m_j, then return 0;

2 Verify $h_j \stackrel{?}{=} H^{(i-j)}(h_i)$, where h_j $(0 \leq j < i)$ is previously received. If the equation does not hold, then return 0;

3 Let $n = ID$. Compute $S = A \cdot H_n(m)^3 + B + C \cdot H_n(m)^2 + D \cdot H_n(m)$ mod n.

4 Verify whether the following equation holds

$$H(h_i \mid S) \stackrel{?}{=} \underbrace{00...0}_{l \text{ bits}} \text{xx...x}.$$

If the above equation does not hold, return 0;

5 Verify whether $9 \cdot A \cdot B \stackrel{?}{=} C \cdot D$ mod n. If the equation holds, then return 1, and 0 otherwise.

receiver further verifies whether $9 \cdot A \cdot B \stackrel{?}{=} C \cdot D$ mod n. This requirement ensures the security of our scheme by preventing an attacker from forging a bogus message m'. Appendix A gives the proof. If the verification of any step above fails, the receiver just drops the message m.

It is clear that the process of pre-authentication credential verification is as fast as the credential generation. To verify a message, the receiver only needs to do two hash operations and seven multiplication operations. We leave our scheme's performance analysis in Chapter 5.5.

5.3.4 Security Analysis

An attacker attempting to generate a bogus m has the following approaches: forging a puzzle key h_i, computing another puzzle solution S', or compromising a secure key d. Our puzzle-based filter scheme along with the offline speedup mechanism is secure against all these

attempts.

- *Forging a puzzle key* h_i: Our scheme uses a one-way hash chain as the puzzle key. An attacker cannot forge h_i based on previous hash elements h_j $(0 < j < i)$ because it is computationally hard to derive h_i given h_j. An attacker can compute h_i only if it knows h_k $(k > i)$, which is nonetheless released 300 ms later. Here, the security of using one-way hash chain is the same as that of TESLA.

- *Computing a new puzzle solution* S': An attacker could attempt to compute a new puzzle solution S' such that $H(h_i|S')$ has the predefined format, which is defined in Step 3 of Algorithm 2. Further, the attacker can forge m', A', B', C', D' such that $S' = A' \cdot H_n(m')^3 + B' + C' \cdot H_n(m')^2 + D' \cdot H_n(m') \bmod n$. However, the attacker can start to compute S' only after he receives the puzzle key h_i. In our scheme, we have a length of l for the key which is long enough such that the computation time for a puzzle solution is far larger than Δ. In this way, after the attacker forges m' and S', the expected arriving time of m has passed and the expected message authentication time is also expired.

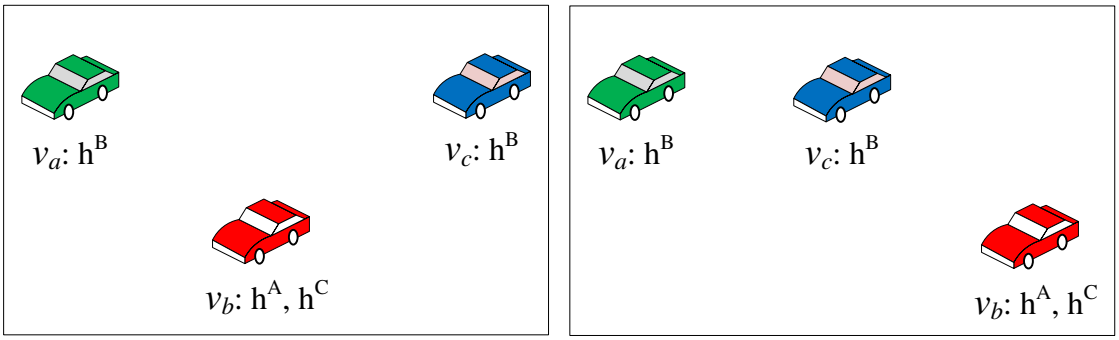
- *Compromising a secure key* d: If an attacker compromises the secret key d, he will be able to forge a message m' easily. Suppose an attacker knows d, he can compute $a+b$ in Algorithm 2 because $a+b = (a+b)^{3d} = (A+B+C+D)^d \bmod n$. He can also compute $aH_n(m)+b$ because $aH_n(m)+b = S^d$ and further compute a, b. Then, the attacker can forge new $m', a', b', A', B', C', D'$ such that $a'H_n(m') + b' = aH_n(m) + b = S^d \bmod n$ and $S = A' \cdot H_n(m')^3 + B' + C' \cdot H_n(m')^2 + D' \cdot H_n(m') \bmod n$. As such, the attacker forges the message m' without modifying S and S^d. Fortunately, in our scheme only a message sender knows the secret parameter d. To compromise d, the attacker has to factorize a large prime n, which equals to $p \cdot q$. Therefore at this point, the security of our scheme is the same as that of large integer factorization. Actually, choosing a relatively small n such that an attacker cannot factorize it within Δ is sufficient to

make our scheme secure. Note that it is useless for an attacker to spend more than Δ to factorize n and compute d because the expected time for the receiver to verify m' is expired. Chapter 5.5 will further discuss the security in our scheme by using a small n. In addition, Appendix B proves that attackers cannot forge m' when only m, a, b, S^d are given.

5.4 Hash Recommendation

Under the proposed scheme, a vehicle can verify the filter of any received message only if it has the corresponding puzzle key, which is either h_0 in $\langle ID \mid m_0 \mid 0 \mid h_0 \mid \sigma_0 \mid Cert_{pk} \rangle$, or h_i in $\langle ID \mid m_i \mid i \mid h_i \mid A \mid B \mid C \mid D \mid \sigma_i \mid Cert_{pk} \rangle$. However, a vehicle does not always have a puzzle key that is associated to a certain sender. For example, in Fig. 5.5(a) scenario I, v_a and v_c are not in each other's communication range; on the other hand in Fig. 5.5(b) scenario II, v_a and v_c are in each other's communication range while v_c overtakes v_a. At this moment, v_a has not verified vehicle v_c's puzzle key yet, and thus v_a cannot verify v_c's filter. As such, an intuitive solution is that v_a verifies the signature in v_c's message so as to obtain v_c's puzzle key, by which v_a can verify the filters in v_c's subsequent messages. However, as long as v_a verifies a message first via its signature, this simply opens a door for the SJA to occur.

To enable a vehicle v to verify a received message first via a filter instead of a signature, a recommendation mechanism is introduced to make v aware of the authentication credential of another vehicle before the vehicle enters the communication range of v. With the credential, v can go through the filter instead of signature verification to authenticate messages from the vehicle. For example in Fig. 5.5(a) scenario I, before v_c comes into the communication range of v_a, the proposed recommendation mechanism enable v_a to receive credential of v_c such that v_c's messages can be verified via a filter. Therefore, at the moment of the scenario II in Fig. 5.5(b), v_a can verify the filter of any v_c's message before verifying its signature to mitigate the threat of SJA.



(a) senario I (b) senario II

Figure 5.5: An example of traffic scenarios

5.4.1 Credential Exchange in Hash Recommendation

Let us review the message format and understand what serves as credentials in the message authentication. There are two types of messages launched by a vehicle. One is *commitment message* that contains the commitment h_0 of a hash chain with a format $\langle ID \mid m_0 \mid 0 \mid h_0 \mid \sigma_0 \mid Cert_{pk} \rangle$; while the other is *common messages* with a format $\langle ID \mid m_i \mid i \mid h_i \mid A \mid B \mid C \mid D \mid \sigma_i \mid Cert_{pk} \rangle$, where h_i with $i \geq 0$ is a puzzle key. Verifying a puzzle key h_i needs either h_{i-1}, h_{i-2}, $\cdots$, or h_0. As long as a vehicle v_b verifies h_j sent by vehicle v_a where $0 \leq j < i$, v_b can thus verify the subsequent messages of v_a containing h_i, $i > j$. Here, h_j can be taken as the authentication credential. The newest released hash element of v_a's hash chain: $h_0, h_1, \ldots, h_r$, is denoted as h^A, and $h^A = h_i$ if v_a has released $i+1$ ($i < n$) hash elements in the chain, which are $h_0, h_1, \ldots, h_i$. To differentiate vehicles' IDs and facilitate finding the index of h_i in a hash chain, h^A is also called a *recommendation hash* of the vehicle and $h^A = \langle ID_A \mid i \mid h_i \rangle$. Note that a recommendation hash $\langle ID_A \mid i \mid h_i \rangle$ serves as a practical presentation of authentication credential and is contained in any common message $\langle ID \mid m_i \mid i \mid h_i \mid A \mid B \mid C \mid D \mid \sigma_i \mid Cert_{pk} \rangle$.

In our recommendation mechanism, each vehicle sends an outgoing message along with k (or less than k) recommendation hashes according to Algorithm 4. Let m_i contain k rec-

Algorithm 4: Recommendation hash selection

Input: β IDs whose corresponding messages have been verified

Result: Broadcast k recommendation hashes

1 Suppose that a vehicle has α neighbors at a time shot, and has already verified β ($\beta < \alpha$) vehicles' messages. Then, the vehicle has β recommendation hashes, each of which corresponds to a neighbor. The set of such β recommendation hashes is called a *recommendation set*. Note that the size of a recommendation set is dynamically changeable.

2 If a vehicle receives a new recommendation hash from another vehicle, the vehicle inserts the newly obtained recommendation hash into its local recommendation set. Here, a vehicle can newly obtain a recommendation hash if the vehicle successfully verifies a commitment message and abstracts its recommendation hashes from the message.

3 If a recommendation hash is not in use for a while, the recommendation hash is deleted form the set. In other words, the hashes in the recommendation set are updated according to their freshness.

4 Let $\mathcal{N}$ denote the size of the recommendation set of vehicle v, and k be a threshold on the number of recommendation hashes. When v sends a periodic message, the message is attached with k selected recommendation hashes from the local recommendation set. If $\mathcal{N} < k$, $k = \mathcal{N}$.

ommendation hashes, h^A, $h^B \ldots$, up to the number of k, and each recommendation hash represents a vehicle whose previous filter has been verified by the message sender. Once receiving a new recommendation hash, a vehicle saves it to their recommendation set defined in Algorithm 4 and uses it to verify the upcoming messages that have the same ID as that in the received recommendation hash. As such, each recommendation hash works as a recommendation "letter" for a particular vehicle, and the exchange of the "letters" constructs the desired trust-relationship network among the benign vehicles.

An example is given by using Fig. 5.5(a) scenario I. v_a and v_c are in the communication range of v_b. Suppose v_b has verified v_a's and v_c's messages, and v_a and v_c have verified v_b's messages. Hence, v_b has h^A and h^C, and both v_a and v_c have h^B. Since v_a is not in the communication range of v_c, hence, v_a does not have h^C, and v_c does not have h^A. After v_c overtakes v_b, as shown in Fig. 5.5(b) scenario II, v_c is in the communication range of v_a. Since v_a and v_c do not have each other's recommendation hash, v_a and v_c cannot verify each other's filter. However, with our proposed recommendation mechanism in Fig. 5.5(a) scenario I, v_b has sent the recommendation hashes h^A and h^C along with its message such that v_a and v_c can obtain h^C and h^A, respectively. In this way, v_a and v_c can verify their filters each other. Such a recommendation mechanism takes advantage of the Inter-Vehicle Communication (IVC) in the VANETs, and the more densely connected a network is, the better our scheme performs.

Note that recommendation hashes sent by an attacker cannot be accepted because a recommendation hash is a part of the message content, where a vehicle only accepts a message after the message's filter and signature are successfully verified. In order to make a legitimate vehicle to believe a fake recommendation hash, the attacker has to sign a message and then reveals its public key and identity. This is out of the scope of our attack model in the sense that an attacker does not intend to reveal his identity to the public. In addition, our mechanism requires the priority to verify a common message higher than that for a commitment message. Here, a vehicle could receive a message which has its recommendation hash either

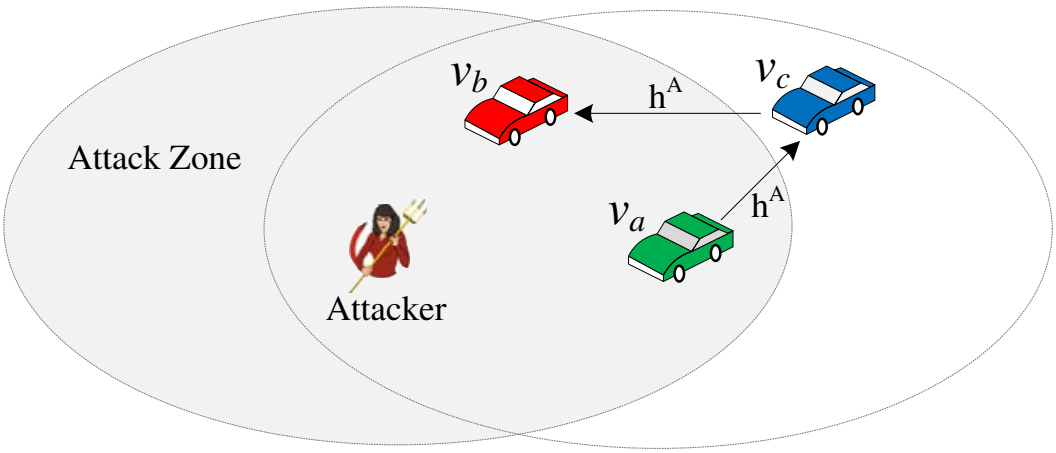


Figure 5.6: An example scenario of defending SJA on a new joining vehicle

already verified or not yet verified. Our scheme designates a higher priority of verification on a message with an already verified recommendation hash. As such, an attacker who attempts to launch SJA cannot delay vehicles to verify the common messages whose recommendation hashes have been disseminated. On the other hand, an attacker could just launch an attack at a certain area which does not affect the whole network. Therefore, as long as a legitimate vehicle verifies a commitment message or a signature in a common message, the legitimate vehicle could recommend the sender's hash to other vehicles. Then, other vehicles can verify the sender filter in the subsequent common messages.

Fig. 5.6 exemplifies a scenario of SJA on vehicle v_a that newly joins the VANET. An attacker sends invalid signatures in its communications range (or referred to as its *attack zone* shown in Fig. 5.6). Since v_b has already had v_c's recommendation hash, the attacker cannot prevent v_b from verifying v_c's messages. Further since v_b in the attack zone might not have a chance to verify v_a's signatures, it cannot verify v_a's recommendation hash h^A. In the event that v_c is outside of the attack zone of the attacker, v_c can verify v_a's signature and h^A by sending the recommendation hash h^A to v_b. Once v_b has h^A, v_b can forward h^A to other vehicles so that h^A is known in the entire attack zone. As such, v_b is able to verify the newly joined vehicle v_a's filters. In addition, the time (hop) to life of a hash recommendation is

restricted to a small number (e.g. five hops to live) so as to the recommendation mechanism not only avoids high communication overhead but also prevents a recommendation hash being stored at a vehicle for a long time. The life time of a hash chain in our scheme could be several minutes or tens of minutes. Before a vehicle changes to use a new hash chain, it can recommend itself to others.

5.5 Performance Analysis

5.5.1 Authentication Delay

The authentication delay caused by filter generation and verification is evaluated in this subchapter.

Filter generation is a puzzle generation process. With the proposed offline speedup mechanism presented in Chapter 5.3.3, it needs to find a puzzle solution $S \in \mathbb{Z}_n^*$, such that

$$H(h_i \mid S) = \underbrace{00...0}_{l \text{ bits}} \text{xx...x}.$$

In the subchapter an experiment is conducted on the puzzle generation time with different lengths of l, Windows CryptoAPI [81] is employed to compute MD5 and SHA1 hashes on a computer with a 1.5 GHz Intel(R) M processor and 1.5 GB DDR RAM. The experiment examines the computation time of solving a puzzle with a different number of l's changed from 1 to 22, and 100 puzzle solutions are found to get the average computation time for each case of a specific number of l's. The result is shown in Fig. 5.7, which clearly shows that the mean time to find a puzzle exponentially increases as l increases. Since the puzzle solution generation in our scheme does not depend on the message itself, the puzzle solution can be found off-line, thereby subject to minimal sending delay compared with that in [71].

Filter verification shown in Algorithm 3 takes a number of hashes for calculating S. Since the hash computation can be performed in an extremely short time, we only need to discuss the computation delay for S: $S = A \cdot H_n(m)^3 + B + C \cdot H_n(m)^2 + D \cdot H_n(m) \bmod n$, which

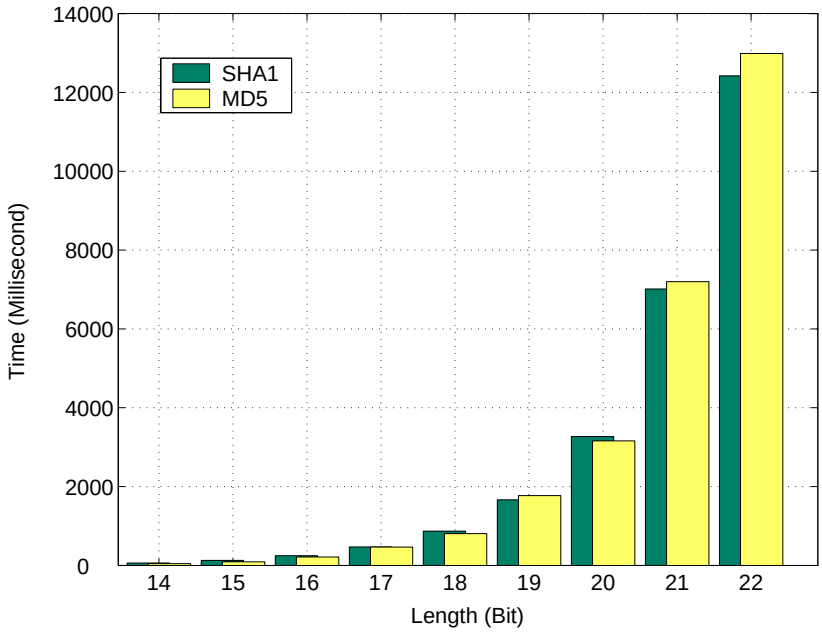


Figure 5.7: The mean time to generate a puzzle solution *vs*. the length of l

takes 5 multiplications, as well as the verification of $9 \cdot A \cdot B \stackrel{?}{=} C \cdot D \bmod n$, which takes 2 multiplications. Thus, the time on the 7 multiplications can also be negligible even if n is 1024-bit long. Note that a strong security level is not required since a filter is used for a local and instant pre-authentication purpose.

Table 5.1: Comparisons of the verification delay in different schemes

	Ours	Puzzle [71]	TESLA [76]
Authentication :	$\surd$	$\surd$	$\surd$
Without sender-side delay:	$\surd$	$\times$	$\surd$
Without receiver-side delay :	$\surd$	$\surd$	$\times$

Another issue is that n can be reused for a number of transmissions but needs to be updated time from time. In case it takes m seconds to successfully factor n, using n in

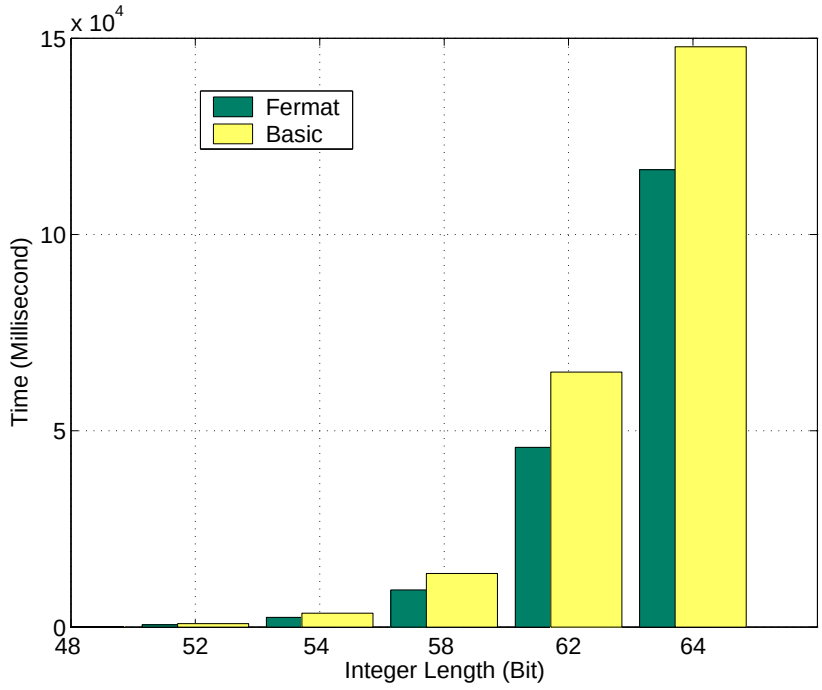


Figure 5.8: The factorization time *vs*. the length of an integer

the proposed pre-authentication mechanism can ensure security for m seconds right after n is being used for the first time. Thus, vehicles should update to get a new n less than every m seconds. To clearly observe the factorization time with distinct lengths of n, a simple experiment is conducted again by using the computer with a 1.5 GHz Intel(R) M processor and 1.5 GB DDR RAM, where an integer is factorized using the Basic and Fermat methods [82], respectively. Fig. 5.8 shows that the factorization time exponentially increases as the integer length increases, and a 64-bit n takes more than 100 seconds for factorization. We also verify the time of performing 7 multiplications with 64-bit n, which is 2.5546×10^{-5} ms and is negligible to the calculation of S. In our work, we recommend using 128-bit n to ensure a longer update period and a more reliable pre-authentication mechanism, while without losing the computation simplicity. Table 5.1 gives a comparison of verification delay in different schemes. Compared with TESLA and the puzzle scheme in [71], our proposed

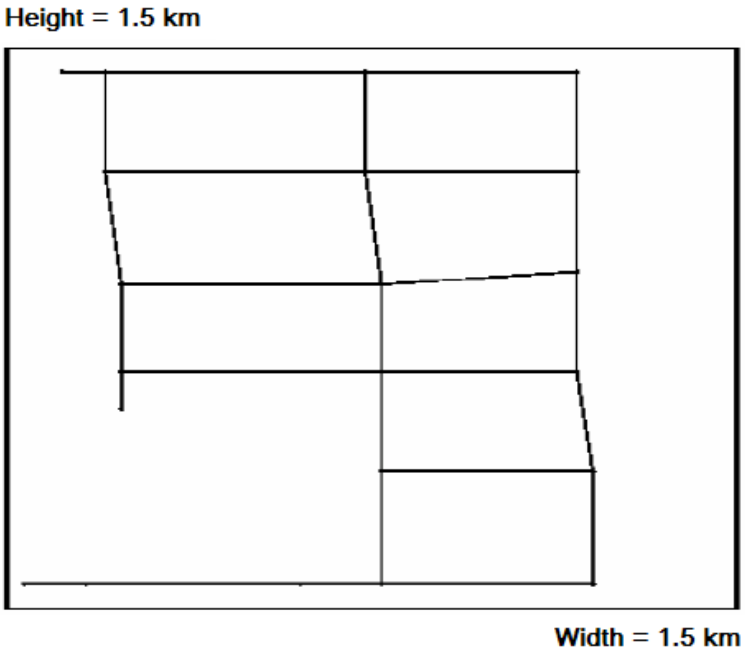


Figure 5.9: A city map with span of 1.5 kilometers

scheme produces neither authentication delay at receiver nor delay at the sender, which can better serve for real-time and dynamic applications such as VANETs.

5.5.2 Verification Percentage

With the proposed hash recommendation scheme to thwart SJA, vehicles need to first verify a filter in a message. To verify a filter, vehicles need to have recommendation hashes of its neighbors. Let α denote the number of neighbors that a vehicle v has, and β denote the number of the neighbors that vehicle v already has their recommendation hashes. Hence, the vehicle v is able to verify β out of α neighbors' messages. Here β/α is termed Verification Percentage (VP), which is desired to be large for better performance.

A simulation is conducted to examine VP under different traffic scenarios and different parameter selection as shown in Fig. 5.10-Fig. 5.13. A city map [76] is used with 9 roads and

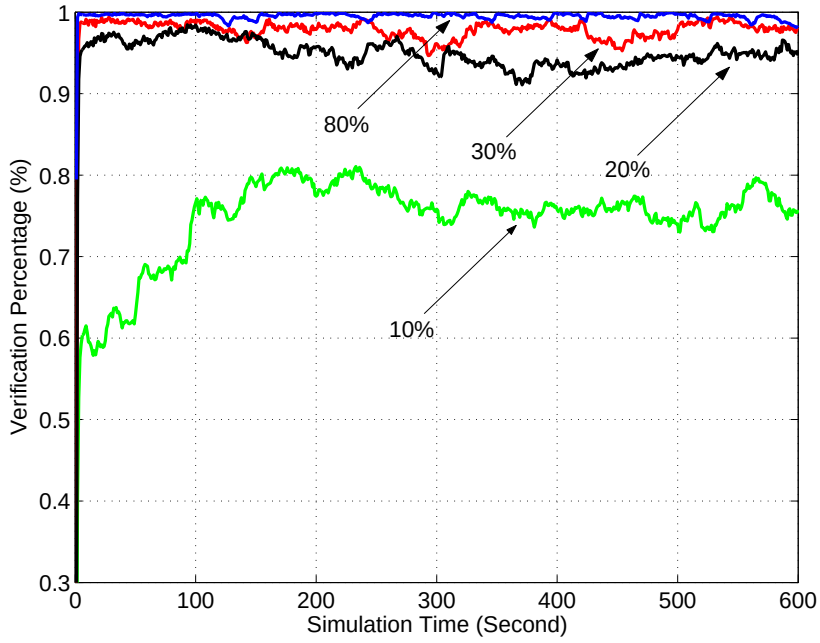


Figure 5.10: Verification percentage is observed in terms of different probability (p)

a span of 1.5 kilometers as shown in Fig. 5.9. On each road vehicles are initially uniformly allocated and move in both directions, either towards west (north) or east (south). At an intersection, a vehicle randomly selects one of possible directions and continue to move. When a vehicle runs out of the city, a new vehicle is generated. The density of vehicles (i.e., distance in meter per vehicle) ranges from 10 to 40, the driving speed ranges from 40 km/h to 100 km/h, and the communication range of each vehicle is 200 meters. The period of sending a recommendation hash message is 0.5 seconds. The number (denoted by k in Algorithm 4) of recommendation hashes within a message ranges from 1 to 8. The simulation duration is 10 minutes.

Let p denote the probability that a vehicle initially has a neighbor's recommendation hash. It is clear that a larger p leads to more scalable verification. On the other hand, p also represents the percentage of the number of valid signatures that a newly joined vehicle receives. Given a vehicle's verification ability, a vehicle can verify N signatures in 300 ms.

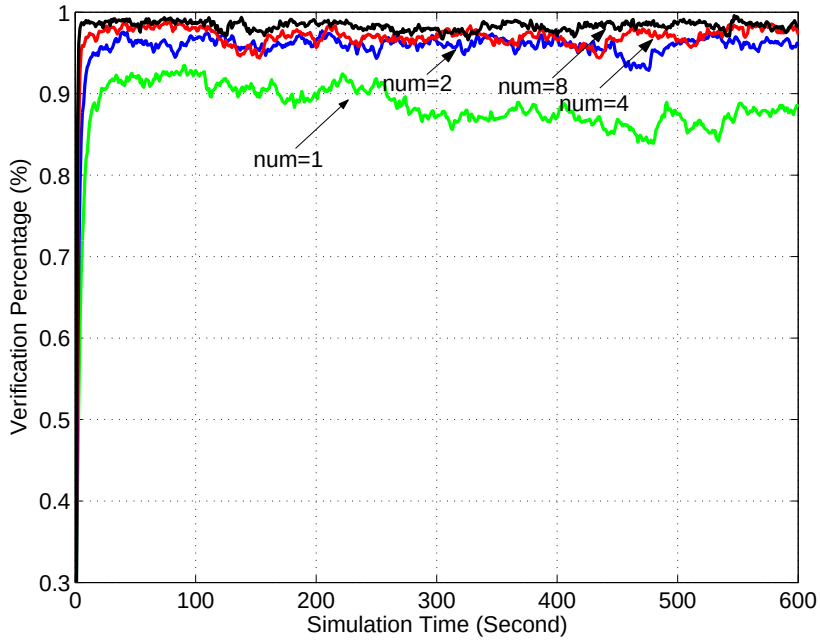


Figure 5.11: Verification percentage is observed in terms of different sending recommendation hash number

However, only about $p \cdot N$ signatures are valid, and $(1-p) \cdot N$ signatures are invalid. Fig. 5.10 shows how VP responds when p is 10%, 20%, 30%, and 80%, respectively. From Fig. 5.10, we can see that when p is as low as 10%, our hash recommendation scheme can still achieve 78% of VP, and almost 100% VP can be achieved when p is equal to 80%.

Given p equals 30%, we evaluate how VP is affected when different numbers of recommendation hashes are sent per 0.5 seconds. Let num denote the number of recommendation hashes sent per 0.5 seconds. From Fig. 5.11, we can see that VP is near 80% in average when num equals 1, and it is increased when num is increased. When num equals 2, VP is equal to 97% in average. However, when num goes even larger, our scheme cannot be benefitted much. This implies that our scheme can achieve high VP without taking much communication overhead.

In Fig. 5.12, we observe whether the driving speed effects VP when $num = 4$. In our

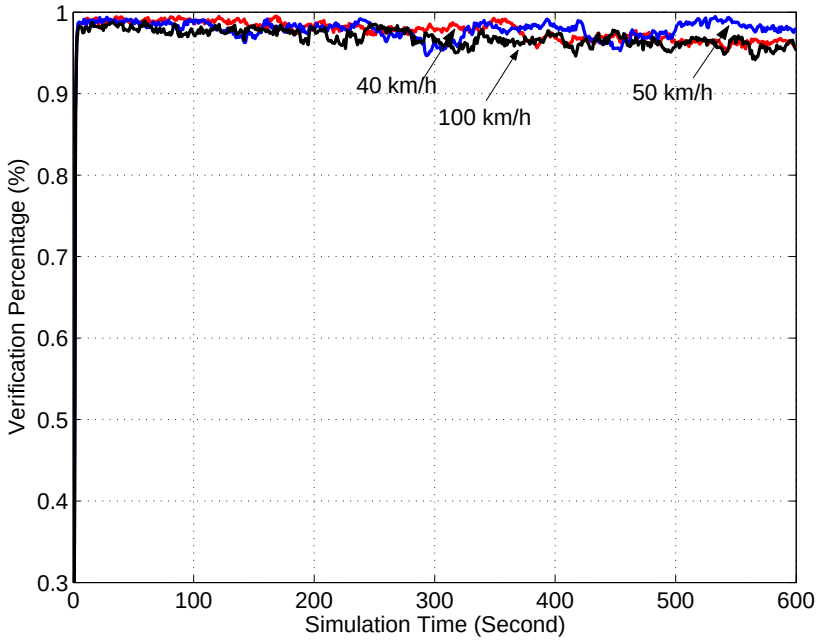


Figure 5.12: Verification percentage is observed in terms of different driving speed

simulation, each vehicle has a baseline driving speed on top of a uniformly random offset ranging from -5 km/h to 5 km/h. In Fig. 5.12, the three curves represent that the baseline driving speeds of vehicles are respectively equal to 40 km/h, 50 km/h, and 100 km/h. From Fig. 5.12, we can see that the driving speed has little effect on VP.

In Fig. 5.13, we change the number of vehicles in our simulation to observe how the traffic density affects VP. The four curves in Fig. 5.13 respectively represent the densities: the inter-vehicle distances are respectively 10m, 20m, 30m, and 40m. Clearly, VP decreases as the traffic density decreases. However, even when the inter-vehicle distance increases to 40m, our scheme can still achieve about 80% of VP.

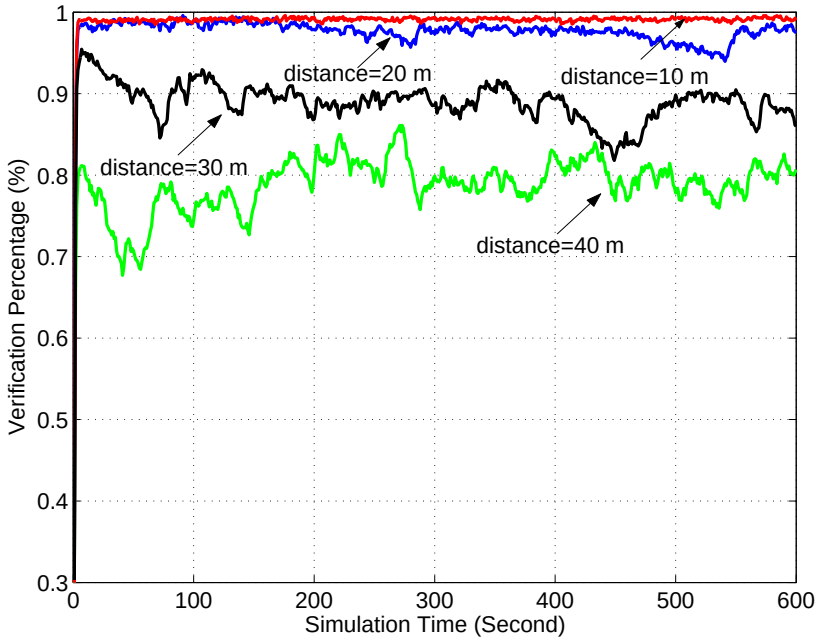


Figure 5.13: Verification percentage is observed in terms of different recommendation hash sending interval

5.6 Summary

In this chapter, we have investigated a possible attack in VANETs, called Signature Jamming Attack (SJA), that could seriously affect the functions of VANETs. To thwart SJA, we have proposed an enhanced hash-based puzzle scheme which can be perfectly fitted into the highly dynamic environment of VANETs. To fully explore the ad hoc characteristic of VANETs, we have introduced a hash recommendation mechanism, with which legitimate vehicles can share their information so as to effectively support the proposed hash-based puzzle scheme. To evaluate the performance of the proposed schemes, we have conducted extensive experiments and showed that our proposed scheme does not produce verification delay at either a sender side or a receiver side. Moreover, through extensive simulations we have demonstrated that our scheme can achieve high verification percentage in distinct traffic scenarios

even when the percentage of neighboring vehicles with acceptable hash recommendations is low.

Chapter 6

Conclusions and Future Work

In this chapter, the contributions of this thesis are concluded, and followed by the future work.

6.1 Contributions

The major contributions of this thesis can be summarized as follows:

- Firstly, an RSU-aided message authentication scheme, called RAISE, is introduced. RAISE taking advantage of the assistance of RSUs makes a vehicle fast verify its neighboring vehicles messages, and thus reduce message authentication delay. Compared with traditional PKI-based message authentication protocols for security and privacy, the computation cost and communication overhead of RAISE are significantly reduced. In addition, RAISE can also achieve conditional privacy preservation. RAISE is capable to trace the real identity of a malicious vehicle when the vehicle abuses its behavior.

- Secondly, considering the situation of absence of RSUs in certain areas, we further proposed an efficient ID-based Batch Verification (IBV) scheme for message authentication in VANETs. IBV work as a supplementary approach to RAISE, and does not

needs the assistance of RSUs. IBV is capable to make vehicles verify a batch of signatures once in stead of using the approach in a one-by-one verifying manner. Hence the average authentication delay on a batch of signatures can be reduced. In addition, IBV can still achieve the same security and conditional privacy requirements as RAISE. To find invalid signatures in a batch of signatures, group testing technique is adopted, which can find invalid signatures efficiently.

- Thirdly, a denial of service attack against message authentication in VANETs is introduced. Such an attack is termed Signature Jamming Attack (SJA), which could delay benign vehicles to verify legitimate messages. To thwart SJA, we proposed an authentication filter mechanism, with which vehicles are able to exclude invalid signatures in the pre-authentication stage before they process the normal signature verification. Our proposed filter scheme has very low verification overhead. It does not produce any real time verification delay at an sender side or at an receiver side.

6.2 Future Work

Our research has made a notable progress on achieving secure message authentication in VANETs. Yet, addressing security and privacy issues is still a very wide-open field, and there are several research directions to be explored to complement our efforts.

6.2.1 Conditional Privacy Preservation in Different Directions

In this thesis, we have discussed and worked on the conditional privacy preservation in a vertical direction, where vehicles are anonymous to each other but their real identities are transparent to a trusted third party, such as TA.

However, in reality, conditional privacy in VANETs needs to be protected in different directions. In some cases, a vehicle would like to expose its identity and location in a certain location or an area. For example, a group of friends drive several cars, and travel in a city

that is strange to them. To real-time trace each other's locations, they may take advantage of VANETs to share vehicle's identities and locations. However, after finishing this trip, they no longer wish to expose their identities and locations for others. In this example, the condition of privacy is regarding to location and time. Particularly, vehicles would like to expose their identities and locations in a certain time period. To the best of our knowledge, this conditional privacy related issues have not been properly and completely solved yet. Thus, conditional privacy protection in different directions is a future work for us to explore.

6.2.2 Incentive in Non-safety Related Applications

In this thesis, we have proposed efficient message authentication schemes for safety related applications in VANETs. However, for non-safety related applications, there are many security issues that have not been well addressed.

For non-safety related applications, for example in a vehicle-based delay tolerant network, vehicles could be used to forward and deliver digital information. A vehicle is treated as a free information carrier. However, in reality, vehicles could refuse to provide the service due to non-profit. As such, we have to design an efficient incentive mechanism and effectively stimulate vehicles to forward messages. However, designing an efficient incentive mechanism is challenging because we have to properly consider and answer the following questions. First, what kind of value (e.g., money, gas coupon, or virtual reputation) could be used as the profit in an incentive mechanism? The value must interest all drivers/vehicles. Otherwise, some vehicles that are not interested in the value could still deny forwarding messages for others. Second, how much value should be given to a package forwarder? Third, how do vehicles redeem the value? and where do vehicles redeem the value? The location for redeem should be easy for vehicle to find. Fourth, how do we thwart replay attacks and collusion attacks? Since some attackers can collude together, and forward the same package circularly and repeatedly in order to gain more profit. Fifth, how do we guarantee the fairness in an incentive system? We need to guarantee that a vehicle must gain its profit after it for-

wards data for others. From the above mentioned issues, we can clearly see that designing an efficient incentive mechanism for VANETs is challenging. Yet, so far existing reported work has not completely solved these issues. Therefore, the designing work on incentive mechanism for non-safety related applications in VANETs still needs exploration in our future research.

In conclusion, because of these unresolved research challenges, we will continue to work on these challenges in the future. We will also consider other latest research progresses such as secure and efficient certificate revocation and study the security issues related to them.

Appendix A

Proof of Filter Unforgeability

In this appendix, we prove that it is computationally hard for attackers to forge m', A', B', C', D' when given m, A, B, C, D and S, where S, A, B, C, D are subject to $S = A \cdot H_n(m)^3 + B + C \cdot H_n(m)^2 + D \cdot H_n(m)$ mod n and $9 \cdot A \cdot B = C \cdot D$ mod n. Before we proof this, first we review the quadratic residuosity problem. Let n be a composite number n that is the product of distinct large prime numbers p and q. Suppose there exists t such that $t^2 = a$ mod n, and a and n are given. If p and q are given, it is easy to compute t. Otherwise, if p and q are unknown, it is hard to compute t.

To forge A', B', C', D', attackers must make them satisfy $A' \cdot B' = C' \cdot D'$ mod n. Attackers select two random numbers in $\mathbb{Z}_n^*$ as A', B', and compute $X = A'B'$ mod n. If attackers attempts to forge m', they have to compute C' that is subject the following two equations.

$$\begin{cases} D' = X/C' \bmod n & (1) \\ S = A' \cdot H_n(m')^3 + B' + C' \cdot H_n(m')^2 + D' \cdot H_n(m') \bmod n & (2) \end{cases}$$

Equation (1) and (2) can merge together to Equation (3) as shown below.

$$H_n(m')^2 \cdot C'^2 + (A' \cdot H_n(m')^3 + B' - S)C' + XH_n(m') = 0 \bmod n \quad (3)$$

Equation (3) is a quadratic equation. $H_n(m')^2$ is the coefficient of C'^2. Computing C' is

a quadratic residuosity problem. As we presented before, it is hard to compute C' without knowing the factorization of n.

Appendix B

Proof of Message Unforgeability

In this appendix, we prove that given m, a, b and S^d, which are subject to $aH_n(m) + b = S^d \bmod n$, it is computationally hard for attackers without knowing the factorization of n to forge m' such that $aH_n(m') + b = S^d \bmod n$, where $a, b, S^d \in \mathbb{Z}_n^*$, i.e., a, b, S^d are coprime to n, $H_n(.)$ is a hash function mapping $\{0,1\}^*$ to an integer in $[0, n]$.

$$\begin{cases} aH_n(m) + b = S^d \bmod n & (1) \\ aH_n(m') + b = S^d \bmod n & (2) \end{cases}$$

Suppose attackers can find m' such that $aH_n(m') + b = S^d \bmod n$, and then we have two equations, (1) and (2), presented above. We compute Equation (1) minus Equation (2), and have Equation (3).

$$a(H_n(m) - H_n(m')) = 0 \bmod n \quad (3)$$

From Equation (3), we can see that there exists an integer k, such that $a(H_n(m) - H_n(m')) = kn$. Due to a is coprime to n, there exists an integer $k' = k/a$. Then we have $H_n(m) - H_n(m') = k'n$. $H_n(m')$ can thereby be presented below.

$$H_n(m') = H_n(m) - k'n \quad (4)$$

In Equation (4), k' can be any integer, i.e., $k' = ..., -2, -1, 0, 1, 2, ...$ However, the values of $H_n(m)$ and $H_n(m')$ must be two integers in $[0, n]$. If $k = 0$, $H_n(m') = H_n(m)$.

In this case, $H_n(m')$ is not a forgery. If $k \neq 0$, $H_n(m')$ cannot be in the range of $[0, n]$. Therefore, $H_n(m)$ cannot be forged.

As such, the only way for attackers to forge m' is to find m' such that $H_n(m') = H_n(m)$. The security level to forge m' is equivalent to the security level to break a hash function. Suppose the length of n is 128 bits. Given m and m', the probability that $H_n(m') = H_n(m)$ is $1/2^{64}$.

Bibliography

[1] Dedicated Short Range Communications (DSRC), [Online]. Available: http://grouper.ieee.org/groups/scc32/dsrc/index.html.

[2] National Highway Traffic Safety Administration, "Traffic Safety Facts Annual Report," [Online]. Available: http://www.nhtsa.gov/portal/nhtsa_static_file_downloader.jsp?file=/staticfiles/DOT/NHTSA/NCSA/Content/TSF/TSF2006- EE.pdf, Jul. 2007.

[3] Denver Regional Council of Governments, "2006 Annual Report on Traffic Congestion in the Denver Region," [Online]. Available: http://www.drcog.org/documents/Final%20Annual%20Report%202006.pdf, May. 2007.

[4] J. A. Misener, "Vehicle-Infrastructure Integration (VII) and Satety: Rubber and Radio Meets the Road in California," *Intellimotion*, Vol. 11, No. 2, pp. 1-3, 2005.

[5] K.-C Chen, "Medium Access Control of Wireless LANs for Mobile Computing," *IEEE Network*, Vol. 8, No. 5, pp. 50-63, 1994.

[6] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: A New VANET-based Smart Parking Scheme for Large Parking Lots," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM'09)*, Brazil, 2009.

[7] K. Sampigethava, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "CARAVAN: Providing Location Privacy for VANET," in *Proceedings of the Embedded Security in Cars (ESCAR)*, 2006.

[8] IEEE Standard 1609.2, "IEEE Trial-Use Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages," Jul, 2006.

[9] J. P. Hubaux, S. Capkun, and J. Luo, "The Security and Privacy of smart vehicles," *IEEE Security and Privacy Magazine*, Vol. 2, No. 3, pp. 49-55, 2004.

[10] M. Raya and J. P. Hubaux, "Security Aspects of Inter-vehicle Communications," in *Proceedings of Swiss Transport Research Conference*, 2005.

[11] M. Raya and J. -P. Hubaux, "The Security of Vehicular Ad Hoc Networks," in *Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks (SASN'05)*, Nov. 2005.

[12] M. Raya and J.-P. Hubaux, "Securing Vehicular Ad Hoc Networks," *Journal of Computer Security*, Vol. 15, No. 1, pp. 39-68, 2007.

[13] X. Sun, X. Lin, and P.-H. Ho, "Secure Vehicular Communications Based on Group Signature and ID-based Signature Scheme," In *Proceedings of IEEE International Conference on Communications (ICC'07)*, pp. 1539-1545, 2007.

[14] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A Secure and Privacy-preserving Protocol for Vehicular Communications" *IEEE Transaction on Vehicular Technology*, Vol. 56, No. 6, pp. 3442-3456, 2007.

[15] G. Calandriello, P. Papadimitratos, A. Lioy, and J.-P. Hubaux, "Efficient and Robust Pseudonymous Authentication in VANET," in *Proceedings of the International workshop on Vehicular ad hoc Networks (VANET'07)*, pp. 19-28, 2007.

[16] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient Conditional Privacy Preservation Protocol for Secure Vehicular Communications," in *Processings of the IEEE International Conference on Computer Communications (INFOCOM'08)*, Phoenix, Arizona, 2008.

[17] J. Freudiger, M. Raya, M. Félegyházi, P. Papadimitratos, and J.-P. Hubaux, "Mix-Zones for Location Privacy in Vehicular Networks," in *Proceedings of ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, 2007.

[18] M. Raya, A. Aziz, and J. P. Hubaux, "Efficient Secure Aggregation in VANETs," in *Proceedings of International workshop on Vehicular ad hoc Networks (VANET'06)*, pp. 67-75, 2006.

[19] M. Raya, P. Papadimitratos, I. Aad, D. Jungels and J.-P. Hubaux, "Eviction of Misbehaving and Faulty Nodes in Vehicular Networks," *IEEE Journal on Selected Areas in Communications, Special Issue on Vehicular Networks*, Vol. 25, No. 8, pp. 1557-1568, 2007.

[20] G. Calandriello, P. Papadimitratos, A. Lioy, and J.-P. Hubaux, "Efficient and robust pseudonymous authentication in VANET," in *Proceedings of the International workshop on Vehicular ad hoc networks (VANET'07)*, 2007.

[21] Y. Xi, K. Sha, W. Shi, L. Schwiebert, and T. Zhang, "Enforcing Privacy Using Symmetric Random Key-set in Vehicular Networks," in *Proceedings of the International Symposium on Autonomous Decentralized Systems (ISADS)*, pp. 344-351, 2007.

[22] J.-H. Song, V. W. S. Wong and V. C. M. Leung, "Wireless Location Privacy Protection in Vehicular Ad-Hoc Networks," *Mobile Networks and Applications*, 2009.

[23] A. Wasef, and X. Shen, "MAAC: Message Authentication Acceleration Protocol for Vehicular Ad Hoc Networks," in *Proceedings of IEEE Global Communications Conference (GLOBECOM '09)*, Nov, 2009.

[24] A. Wasef, and X. Shen, "ASIC: Aggregate Signatures and Certificates Verification Scheme for Vehicular Networks," in *Proceedings of IEEE Global Communications Conference (GLOBECOM '09)*, Nov, 2009.

[25] R. Lu, X. Lin, and X. Shen, "SPRING: A Social-based Privacy-preserving Packet Forwarding Protocol for Vehicular Delay Tolerant Networks," in *Processings of the IEEE International Conference on Computer Communications (INFOCOM'10)*, Mar, 2010.

[26] S. Dietzel, E. Schoch, F. Kargl, B. Konings, and M. Weber, "Resilient Secure Aggregation for Vehicular Networks," *IEEE Network - Special Issue on "Advances in Vehicular Communications Networks*, 2010.

[27] Y. Sun, X. Lin, R. Lu, X. Shen, and J. Su, "A Secure and Efficient Revocation Scheme for Anonymous Vehicular Communications,"in *Proceedings of IEEE International Conference on Communications (ICC'10)*, May, 2010.

[28] Y. Sun, X. Lin, R. Lu, X. Shen, and J. Su, "Roadside Units Deployment for Efficient Short-time Certificate Updating in VANETs," in *Proceedings of IEEE International Conference on Communications (ICC'10)*, May, 2010.

[29] A. Wasef, and X. Shen, "REP: Location Privacy for VANETs Using Random Encryption Periods", *ACM Mobile Networks and Applications (MONET)*, to appear.

[30] A. Perrig, R. Canetti, J. D. Tygar, D. Song, "The TESLA Broadcast Authentication Protocol," *CryptoBytes*, Vol. 5, No. 2, pp. 2-13, 2002.

[31] A. Perrig, R. Canetti, J. D. Tygar, D. Song, "Efficient Authentication and Signing of Multicast Streams Over Lossy Channels," *IEEE Symposium on Security and Privacy (S&P'00)*, 2000.

[32] Shamus Software. MIRACL library, [Online]. Available: http://www.shamus.ie/index.php?page=Elliptic-Curve-point-multiplication.

[33] H. Wedde, S. Lehnhoff, B. Bonn, "Highly Dynamic and Scalable VANET Routing for Avoiding Traffic Congestions," in *Proceedings of the International Workshop on Vehicular ad hoc Networks (VANET'07)*, pp. 81-82, 2007.

[34] W. Wang, F. Xie, M. Chatterjee, "An Integrated Study on Mobility Models and Scalable Routing Protocols in VANETs," in *Processings of the International Workshop on Mobile Networking for Vehicular Environments*, pp. 97-102, 2007.

[35] B. Scheuermann, C. Lochert, J. Rybicki, M. Mauve, "A Fundamental Scalability Criterion for Data Aggregation in VANETs," in *Proceedings of the International Conference on Mobile Computing and Networking (Mobicom'09)*, pp. 285-296, 2009.

[36] J. Freudiger, M. Raya, and M. Feleghhazi, "Mix Zones for Location Privacy in Vehicular Networks," in *Processings of the International Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS'07)*, Aug. 2007.

[37] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, Vol. 22, No. 5, pp. 644-654, 1976.

[38] Douglas R. Stingson, "Cryptography: Theory and Practice, third edition," *CRC Press*, 2005.

[39] L. Sweeney, "K-ANONYMITY: A Modle for Protecting Privacy," *International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems*, Vol. 10, No. 5, pp. 557-570, 2002.

[40] The Network Simulator - ns-2. [Online] Available: http://nsnam.isi.edu/nsnam/index.php.

[41] D. R. L. Brown, "Generic Groups, Collision Resistance, and ECDSA, " *Designs, Codes and Cryptography*, Vol. 35, pp. 119-152, 2005.

[42] X. Lin, *Secure and privacy-preserving vehicular communications*, Ph.D Thesis, University of Waterloo, 2008.

[43] K. Ren, W. Lou, R. H. Deng, and K. Kim, "A Novel Privacy Preserving Authentication and Access Control Scheme in Pervasive Computing Environments," *IEEE Transactions on Vehicular Technology*, Vol. 55, No. 4, pp.1373-1384, 2006.

[44] D. Boneh and M. Franklin, "Identity-based Encryption from the Weil pairing,", in *Proceedings of Crypto*, LNCS, Vol. 2139, pp. 213-229, 2001.

[45] S. Galbraith, K. Harrison, D. Soldera, "Implementing the Tate Pairing," *Algorithmic Number Theory*, LNCS, Vol. 2369, pp. 69-86, 2002.

[46] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "A Survey of Two Signature Aggregation Techniques," *CryptoBytes*, Vol. 6, No. 2, 2003.

[47] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," in *Proceedings of Asiacrypt*, Vol. 2248, pp. 514-532, 2001.

[48] A. Fiat, "Batch RSA," in *Proceedings of Crypto*, pp. 175-185, 1989.

[49] D. Naccache, D. M'Raïhi, S. Vaudenay, and D. Raphaeli, "Can D.S.A Be Improved? Complexity Trade-offs with the Digital Signature Standard," in *Proceedings of EUROCRYPT*, LNCS, Vol. 950, pp. 77-85, 1994.

[50] J. C. Cha and J. H. Cheon, "An Identity-based Signature From Gap Diffie-Hellman Groups," in *Proceedings of Public Key Cryptography*, pp. 18-30, 2003.

[51] F. Zhang, R. Safavi-Naini, and W. Susilo, "Efficient Verifiably Encrypted Signature and Partially Blind Signature from Bilinear Pairings," in *Proceedings of Indocrypt*, LNCS, Vol. 2904, pp. 191-204, 2003.

[52] F. Zhang and K. Kim, "Efficient ID-based Blind Signature and Proxy Signature from Bilinear Pairings," in *Proceedings of Australasian Conference on Information Security and Privacy (ACISP'03)*, LNCS, Vol. 2727, pp. 312-323, 2003.

[53] H. Yoon, J. H. Cheon, and Y. Kim, "Batch Verification with ID-based Signatures," in *Proceedings of Information Security and Cryptology*, pp. 233-248, 2004.

[54] J. Camenisch, S. Hohenberger, and M. ∅. Pedersen, "Batch Verification of Short Signatures," in *Proceedings of EUROCRYPT*, LNCS, Vol. 4514, pp. 246-263, 2007.

[55] J. Camenisch and A. Lysyanskaya, "Signature Schemes and Anonymous Credentials from Bilinear Maps," in *Proceedings of Crypto*, LNCS, Vol. 3152, pp. 56-72, 2004.

[56] M. Raya, P. Papadimitratos, and J.-P. Hubaux, "Securing Vehicular Communications," *IEEE Wireless Communications Magazine, Special Issue on Inter-Vehicular Communications*, Vol. 13, No. 5, pp. 8-15, 2006.

[57] B. Kim, K. Choi, J. Lee, and D. Lee, "Anonymous and Traceable Communication Using Tamper-Proof Device for Vehicular Ad Hoc Networks," in *Proceedings of the International Conference on Convergence Information Technology (ICCIT'07)*, 2007.

[58] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," IETF RFC 3174, 2001.

[59] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, Vol. 31, No. 4, pp. 469-472, 1985.

[60] V. S. Miller, "Use of elliptic curves in crytography," in *Proceedings of Advance in Cryptology*, pp. 417-426, Aug. 1985.

[61] A. Miyaji, M. Nakabayashi, and S. Takano, "New Explicit Conditions of Elliptic Curve Traces for FR-reduction", *IEICE Transactions on Fundamentals*, Vol. E84-A, No. 5, pp. 1234-123, 2001.

[62] D. Du and F.K. Hwang, "Combinatorial Group Testing and Its Applications 2nd Edition," *World Scientific Press*, Singapore, 2000.

[63] G. M. Zaverucha and D. R. Stinson, "Group Testing and Batch Verification," *International Conference on Information Theoretic Security (ICITS'09)*, 2009.

[64] N. Ye, "Robustness of the Markov-Chain Model for Cyber-Attack Detection," *IEEE Transactions on Reliability*, Vol. 53, No. 1, pp. 116-123, 2004.

[65] N. Ye, "A Markov Chain Model of Temporal Behavior for Anomaly Detection," in *Proceedings of the IEEE Systems, Man, and. Cybernetics Information Assurance Workshops*, 2000.

[66] A.A. Markov, "Extension of the limit theorems of probability theory to a sum of variables connected in a chain," reprinted in Appendix B of: R. Howard. Dynamic Probabilistic Systems, volume 1: Markov Chains. John Wiley and Sons, 1971.

[67] M. Scott, "Efficient implementation of crytographic pairings," [Online]. Available: http://ecrypt-ss07.rhul.ac.uk/ Slides/Thursday/mscott-samos07.pdf

[68] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology*, LNCS, Vol. 3152, pp. 41-55, 2004.

[69] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," In *Procedings of Eurocrypt*, LNCS, Vol. 2656, pp. 416-432, 2003.

[70] T. Guneysu, C. Paar, "Ultra High Performance ECC over NIST Primes on Commercial FPGAs," in *Proceedings of the 10th International Workshop on Cryptographic Hardware and Embedded Systems (CHES'08)*, pp. 62-78, 2008.

[71] P. Ning, A. Liu, W. Du, "Mitigating DoS Attacks against Broadcast Authentication in Wireless Sensor Networks," *ACM Transactions on Sensor Networks (TOSN)*, Vol. 4, No. 1, 2008.

[72] R. Wang, W. Du, and P. Ning, "Containing Denial-of-service Attacks in Broadcast Authentication in Sensor Networks," in *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (Mobihoc'07)*, pp 71-79, 2007.

[73] K. Ren, W. Lou and Y. Zhang, "LEDS: Providing Location-aware End-to-end Data Security in Wireless Sensor Networks," *IEEE Transactions on (TMC)*, Vol. 7, No. 5, pp. 585-598, 2008.

[74] Q. Dong, D. Liu, P. Ning, "Pre-Authentication Filters: Providing DoS Resistance for Signature-Based Broadcast Authentication in Wireless Sensor Networks," in *Proceedings of ACM Conference on Wireless Network Security (WiSec'08)*, pp. 2-12, 2008.

[75] S. Hyun, P. Ning, A. Liu, W. Du, "Seluge: Secure and DoS-Resistant Code Dissemination in Wireless Sensor Networks," in *Proceedings of Information Processing in Sensor Networks (IPSN'08)*, pp. 445-456, April 2008.

[76] X. Lin, X. Sun, X. Wang, C. Zhang, P.-H. Ho and X. Shen," TSVC: Timed Efficient and Secure Vehicular Communications with Privacy Preserving," *IEEE Transactions on Wireless Communications*, Vol. 7, No. 12, pp. 4987-4998, 2008.

[77] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, Vol. 24, No. 11, pp. 770-772, 1981.

[78] A. Chefranov, "One-Time Password Authentication with Infinite Hash Chains," *Book of Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*, pp.283-286, 2008.

[79] R. Rivest and A. Shamir, "PayWord and MicroMint: Two Simple Micropayment Schemes," *CryptoBytes*, Vol 2, pp. 7-11, No. 1 1996.

[80] P. Golle and N. Modadugu, "Authenticating Streamed Data in the Presence of Random Packet Loss," in *Proceedings of Network and Distributed System Security Symposium (NDSS'01)*, pp. 13-22, Feb. 2001.

[81] Windows CryptoAPI, [Online] Available: http://msdn.microsoft.com/en-us/library/aa380255 (VS.85).aspx.

[82] Integer factorization source code, [Online] Available: http://www.frenchfries.net/paul/factoring/source.html.