Managing and Complementing Public Key Infrastructure for Securing Vehicular Ad Hoc Networks

by

Albert Wasef

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, 2011

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

Recently, vehicular ad-hoc network (VANET) has emerged as an excellent candidate to change the life style of the traveling passengers along the roads and highways in terms of improving the safety levels and providing a wide range of comfort applications. Due to the foreseen impact of VANETs on our lives, extensive attentions in industry and academia are directed towards bringing VANETs into real life and standardizing its network operation.

Unfortunately, the open medium nature of wireless communications and the high-speed mobility of a large number of vehicles in VANETs pose many challenges that should be solved before deploying VANETs. It is evident that any malicious behavior of a user, such as injecting false information, modifying and replaying the disseminated messages, could be fatal to other legal users. In addition, users show prime interest in protecting their privacy. The privacy of users must be guaranteed in the sense that the privacy-related information of a vehicle should be protected to prevent an observer from revealing the real identities of the users, tracking their locations, and inferring sensitive data. From the aforementioned discussion, it is clear that security and privacy preservation are among the critical challenges for the deployment of VANETs. Public Key Infrastructure (PKI) is a well-recognized solution to secure VANETs. However, the traditional management of PKI cannot meet the security requirements of VANETs. In addition, some security services such as location privacy and fast authentication cannot be provided by the traditional PKI. Consequently, to satisfy the security and privacy requirements, it is prerequisite to elaborately design an efficient management of PKI and complementary mechanisms for PKI to achieve security and privacy preservation for practical VANETs. In this thesis, we focus on developing an efficient certificate management in PKI and designing PKI complementary mechanisms to provide security and privacy for VANETs. The accomplishments of this thesis can be briefly summarized as follows.

Firstly, we propose an efficient Distributed Certificate Service (DCS) scheme for vehic-

ular networks. The proposed scheme offers a flexible interoperability for certificate service in heterogeneous administrative authorities, and an efficient way for any On-Board Units (OBUs) to update its certificate from the available infrastructure Road-Side Units (RSUs) in a timely manner. In addition, the DCS scheme introduces an aggregate batch verification technique for authenticating certificate-based signatures, which significantly decreases the verification overhead.

Secondly, we propose an Efficient Decentralized Revocation (EDR) protocol based on a novel pairing-based threshold scheme and a probabilistic key distribution technique. Because of the decentralized nature of the EDR protocol, it enables a group of legitimate vehicles to perform fast revocation of a nearby misbehaving vehicle. Consequently, the EDR protocol improves the safety levels in VANETs as it diminishes the revocation vulnerability window existing in the conventional Certificate Revocation Lists (CRLs).

Finally, we propose complementing PKI with group communication to achieve location privacy and expedite message authentication. In specific, the proposed complemented PKI features the following. First, it employs a probabilistic key distribution to establish a shared secret group key between non-revoked OBUs. Second, it uses the shared secret group key to perform expedite message authentication (EMAP) which replaces the time-consuming CRL checking process by an efficient revocation checking process. Third, it uses the shared secret group key to provide novel location privacy preservation through random encryption periods (REP) which ensures that the requirements to track a vehicle are always violated. Moreover, in case of revocation an OBU can calculate the new group key and update its compromised keys even if the OBU missed previous rekeying process.

For each of the aforementioned accomplishments, we conduct security analysis and performance evaluation to demonstrate the reliable security and efficiency of the proposed schemes.

Acknowledgements

Words fall short to express my gratitude and appreciation to my advisor Professor Xuemin (Sherman) Shen. Your continuous guidance and constructive comments were always leading my way through this research. I have gained many skills from Professor Shen including writing papers, problem-solving, giving presentations, and too many things to mention here. Professor Shen is always my symbol for hard work and excellence. Your belief in me and continuous support have been a tremendous incentive for completing this research. It is my honor that I worked under the supervision of Professor Shen.

I would like also to thank and express my appreciation to the examining committee members: Professor Sagar Naik, Professor Zhou Wang, Professor Liping Fu, and Professor Weisong Shi. I appreciate their time and effort devoted for reading my Ph.D. thesis and providing me with their insightful comments and invaluable suggestions, thereby further improving the quality of my research work.

I would like to thank all the members of the Broadband Communications Research (BBCR) Group for their support and collaboration. My discussion with Yixin Jiang, Rongxing Lu, Nizar Alsharif, Xiaodong Lin, Ho Ting Cheng, Chenxi Zhang, and Mohamed Elsalih was a tremendous resource of knowledge for me. It is a privilege for me to be a member of the BBCR group and enjoy the friendly working environment in the group.

Also, I would like to thank the ECE administrative and technical staff for their help, kindness, patience, and cooperation.

Finally, special thanks goes to my beloved wife, Marian, and my kids, Andrew and Veronica for helping and supporting me to go this far. Marian, I owe you my deepest gratitude for your endless patience and support that accompanied me along this long journey. Your words were always supporting and helping me to go through many difficult times. You are always my strength.

This thesis is dedicated to my parents for their endless support, inspiration, and encouragement.

Contents

List of Tables

List of Figures

List of Abbreviations

CA	Certification Authority
CDH	Computational Diffie-Hellman Problem
CRL	Certificate Revocation List
DCS	Distributed Certificate Service
DSRC	Dedicated Short Range Communication
ECDLP	Elliptic Curve Discrete Logarithm Problem
EDR	Efficient Decentralized Revocation
EMAP	Expedite Message Authentication
GPS	Global Positioning System
HSM	Hardware Security Module
KDC	Key Distribution Center
MANET	Mobile Ad Hoc Network
MA	Master Authority
OBU	On Board Unit
PKI	Public Key Infrastructure
PK	Public Key
REP	Random Encryption Period

RSU	Roadside Unit
SK	Secret Key
TA	Trusted Authority
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VANET	Vehicular Ad Hoc Network
WAVE	Wireless Access in Vehicular Environments

Chapter 1

Introduction

1.1 Overview of Vehicular Ad Hoc Networks

Every year many accidents happen causing injuries and fatalities. For example, the road-injury statistics in Canada in 1996 indicate that around 230,000 person incurred serious injuries, and road fatalities were around 3,000 [1]. The traffic statistics are worse in the U.S. The health care expenses associated with these accidents form a burden on the economy of any country. These statistics raise the query to achieve better road safety. As a result for the advances in wireless communications technologies, vehicular ad-hoc network (VANET) emerged as an excellent candidate to change the life style of the traveling passengers along the roads and highways in terms of improving the safety levels and providing a wide range of comfort applications.

Due to the foreseen impact of VANETs on our lives, extensive attentions in industry and academia are directed towards bringing VANETs into real life and standardizing its network operation. As a result, IEEE developed the IEEE 1609 Wireless Access in Vehicu-

lar Environments (WAVE) standard for VANETs [2]. Moreover, the American Society for Testing and Materials (ASTM) and IEEE have developed Dedicated Short Range Communication (DSRC) [3] as the basic vehicular communications technology, where DSRC has bandwidth of 75 MHz at the frequency range of 5.9 GHz. VANETs consist of network entities, mainly including vehicles, which is usually called On Board Units (OBUs), and infrastructure Roadside Units (RSUs). Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communications are two basic vehicular communication modes, which respectively allow vehicles to communicate with each other or with the roadside infrastructure.

According to the WAVE standard for VANETs [2], each vehicle will periodically broadcast a message over a single hop every 300 $msec$. The nominal transmission range of each vehicle is 300 m. When a vehicle receives a message, it takes an action based on the content of the received message. For example, if a traffic jam warning is received, the vehicle can select another route to reach its destination. To enable vehicles to exchange messages with each other or with the RSUs, they should be loaded with the following:

- Wireless communication module for transmitting and receiving messages;
- A set of sensors for collecting information about the neighboring environment, e.g., slippery road, icy road, a vehicle ahead, etc;
- A Global Positioning System (GPS) receiver for determining the vehicle's location;
- On-Board processing Unit (OBU) that processes all the gathered information and sends messages to its neighboring vehicles; and
- A Hardware Security Module (HSM), which is a tamper-resistant module used to store the security materials, e.g., secret keys, certificates, etc., of the OBU. The HSM

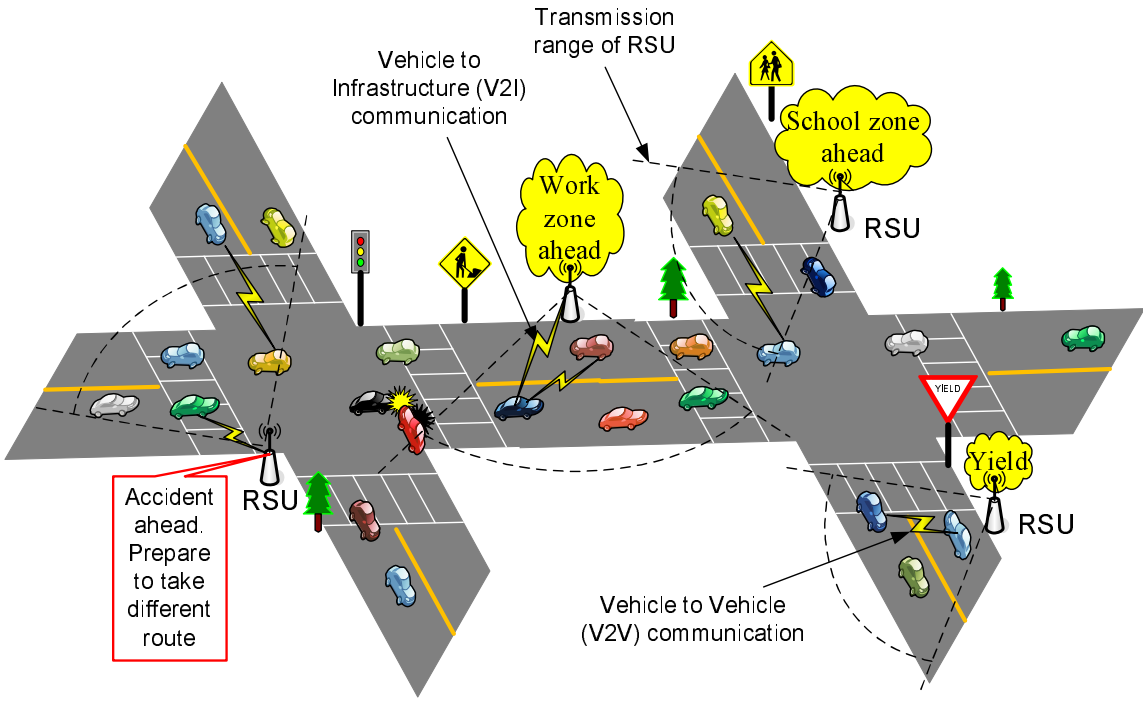


Figure 1.1: Examples of safety applications in VANET

in each OBU is responsible for performing all the cryptographic operations such as signing messages, verifying certificates, keys updating, etc.

VANETs have a wide variety of safety applications. This includes but not limited to: cooperative driving, collision warning, work zone warning, road feature notification, lane change assistance, highway merging assistance, etc. Some examples of these safety applications are shown in Fig. 1.1.

1.2 Network Characteristics

The characteristics of VANETs are different from any other networks. Those characteristics can be summarized as follows [4] [5].

1. Rapid change in topology: Since vehicles are moving with high speeds, the topology of VANET is prone to frequent and rapid changes.

2. Network subject to fragmentation: When the headway between two vehicles traveling on the road is greater than the communication range of the vehicles, it is said that the network is fragmented. Due to the high speed of the vehicles, it is highly likely that fragmentation occurs in many parts of the network, especially when the density of the vehicles is low.

3. No significant power and memory constraints: Since the batteries of the vehicles are continuously charging while the vehicle is moving, nodes (vehicles) in VANET do not suffer from the conventional power constraints of the hand-held devices in the Mobile Ad Hoc Networks (MANETs). In addition, vehicles have ample memory storage compared to the limited memory in the hand-held devices in MANETs.

4. Large scale: VANET constitutes the largest instance of MANETs that the world have ever seen, where the order of the number of the nodes is in the range of 10^7.

5. Predictable mobility patterns: A double-edged sword of VANETs is that the paths a vehicle can take are well-defined by the roads' boundaries and directions. Consequently, the mobility of vehicles can be expected to a large extent. Such mobility prediction can be helpful in data dissemination, however, it also can lead to breaching the privacy of the users as it may facilitate vehicle tracking.

6. Variable network density: The number of vehicles in one area of the road is temporally changing during the day, e.g., roads in the rush hours are more congested than other times of the day. Also, at the same time of the day, some areas on the roads get more congested than other areas subject to the interest of the drivers.

7. Well-defined node locations: A prerequisite for deploying VANETs is to equip each vehicle with a Global Positioning System (GPS) receiver. Thus, each vehicle has the ability to get a relatively accurate information about its location. Nodes' locations are of particular advantage to data dissemination and routing protocols as well as safety applications.

1.3 Motivation

On one hand, VANETs will make a revolution in our traditional transportation systems. On the other hand, the open medium nature of wireless communications and the high-speed mobility of a large number of vehicles in VANETs pose many challenges that should be solved before deploying VANETs. It is evident that any malicious behavior of a user, such as injecting false information, modifying and replaying the disseminated messages, could be fatal to other legal users. In addition, users show prime interest in protecting their privacy. The privacy of users must be guaranteed in the sense that the privacy-related information of a vehicle should be protected to prevent an observer from revealing the real identities of the users, tracking their locations, and inferring sensitive data [6],[7]. One key factor for increasing the social acceptance of VANETs is to ensure the privacy preservation in VANETs. From the aforementioned discussion, it is clear that security and privacy preservation are among the critical challenges for the deployment of VANETs.

To satisfy the security and privacy requirements, it is prerequisite to elaborately design a suite of mechanisms to achieve security and privacy preservation for practical VANETs. In academia, a lot of efforts have been done to provide secure and privacy-preserving vehicular communications.

A well-recognized solution is to deploy Public Key Infrastructure (PKI) [8], where each OBU has a set of authentic certificates. To protect the privacy of users, each OBU should use a certificate for a short duration and after that it has to replace this certificate, i.e., OBUs continuously consume their certificate sets. Eventually, each OBU will need to update its certificates. In classical PKI, any certificate update must be performed through a central Certification Authority (CA), which sends the updated certificate to the requesting OBU through the available RSUs on the roads. The centralized certificate update process in the classical PKI may be impractical in the large scale VANETs due to the following reasons: (1) Each CA encounters a large number of certificate update requests which can render the CA a bottle-neck; (2) The certificate update delay is long relative to the short V2I communication duration between the immobile RSUs and the highly mobile OBUs during which the new certificate should be delivered to the requesting OBU. The long certificate update delay is due to the fact that a request submitted by an OBU to an RSU must be forwarded to the CA, and CA has to send the new certificate to that RSU which in turn forwards the new certificate to the requesting OBU. Accordingly, the classical PKI should be pruned or optimized to satisfy the certificate service requirement in vehicular communication scenarios. To provide a practical certification service for VANETs, it is required for each OBU to efficiently update its certificate in a timely manner. The certification service should also be decentralized to enable VANET to efficiently process the expected large number of certificate update requests. Moreover, to protect the user privacy, the updated certificates should be anonymous and free from the key escrow issue. In

addition, an important issue that should be taken into consideration in the employed PKI is the efficient authentication as each vehicle will receive a large number of messages from its neighboring vehicles.

Another important issue is the roaming between different domains [9],[10]. The OBUs should have the capabilities to roam between domains administered by different CAs. The Wireless Access in Vehicular Environments (WAVE) standard [2] did not consider the roaming between different domains, and the interopreability between different CAs is still an open problem that has not been previously tackled in the VANET literature.

Efficient certificate management is essential for reliable and robust operation of any PKI. A critical part of any certificate management scheme is the revocation of misbehaving nodes. Certificate revocation can be centralized or decentralized. For centralized revocation, a central entity, such as the CA, is the only entity in the network that can take the revocation decision for a certain node. For decentralized revocation, the node revocation is done by the neighboring nodes of the misbehaving node. In the centralized revocation, the distribution of CRLs is prone to long delays [11] [12]. Moreover, centralizing the revocation decision to the CA renders the CA a bottleneck and a single point of failure. If the revocation process experiences a long delay, the misbehaving vehicle can harm other vehicles until the revocation process is completed. Consequently, an efficient decentralized revocation technique is necessary for the reliable operation of VANETs.

According to the Dedicated Short Range Communications (DSRC) specifications [3], each vehicle periodically broadcasts a message every 300 msec. Those messages are not intended to a specific vehicle, but multicasted to neighboring vehicles on the road. Even though PKI is employed to secure VANET, the multicasted messages contain critical information, such as location, speed, and direction of the transmitting vehicle. These critical information are sent in clear and are not secured by PKI. An adversary can manipulate this

information to track a vehicle even if anonymous certificates are employed. For example, an eavesdropper can use the current location of a targeted vehicle and its current speed to calculate the expected time for receiving another message from the same vehicle at another location on the road. Even if the targeted vehicle changed its anonymous certificate, the eavesdropper may still able to track it. Consequently, there is a need to supplement PKI with a scheme that is capable of preserving the location privacy of the users in VANETs.

1.4 Research Contributions

The research contributions of this thesis can be summarized as follows.

- We propose an efficient Distributed Certificate Service (DCS) scheme. The proposed scheme offers a flexible interoperability for certificate service in heterogeneous administrative authorities, and an efficient way for any On-Board Units (OBUs) to update its certificate from the available infrastructure Road-Side Units (RSUs) in a timely manner. In addition, the DCS scheme introduces an aggregate batch verification technique for authenticating certificate-based signatures, which significantly decreases the verification overhead. Security analysis and performance evaluation demonstrate that the DCS scheme can reduce the complexity of certificate management, and achieve excellent security and efficiency for vehicular communications. This work is presented in Chapter 3.

- We propose an Efficient Decentralized Revocation (EDR) protocol based on a novel pairing-based threshold scheme and a probabilistic key distribution technique. Because of the decentralized nature of the EDR protocol, it enables a group of legitimate vehicles to perform fast revocation of a nearby misbehaving vehicle. Consequently,

the EDR protocol improves the safety levels in VANETs as it diminishes the revocation vulnerability window existing in the conventional Certificate Revocation Lists (CRLs). By conducting detailed performance evaluation, the EDR protocol is demonstrated to be reliable, efficient, and scalable. This work is introduced in Chapter 4. The DCS and EDR schemes can provide the targeted efficient PKI management to provide most of the security requirements for VANETs.

- We propose complementing PKI with group communication to achieve location privacy and expedite message authentication. In specific, the proposed complemented PKI features the following. First, it employs a probabilistic key distribution to establish a shared secret group key between non-revoked OBUs. Second, it uses the shared secret group key to perform expedite message authentication (EMAP) which replaces the time-consuming CRL checking process by an efficient revocation checking process. Third, it uses the shared secret group key to provide novel location privacy preservation through random encryption periods (REP) which ensures that the requirements to track a vehicle are always violated. Moreover, in case of revocation an OBU can calculate the new group key and update its compromised keys even if the OBU missed previous rekeying process. This work is presented in Chapter 5. The DCS , EDR, and PKI complementing mechanisms can provide a comprehensive security and privacy for VANETs.

1.5 Outline of The Thesis

The remainder of the thesis is organized as follows. We present the state of the art in security and privacy preservation in VANETs in Chapter 2. Chapter 3 introduces the efficient

Distributed Certificate Service (DCS) scheme. In Chapter 4, we present the proposed Efficient Decentralized Revocation (EDR) protocol. Chapter 5 introduces the complemented PKI to achieve expedite message authentication and location privacy. Finally, Chapter 6 concludes the thesis and presents the future work.

Chapter 2

State of The Art in Security and Privacy in VANETs

In this Chapter, we present the state of the art in security and privacy preservation in VANETs. This Chapter is structured as follows. In section 2.1, we illustrate the security threats to VANETs. The security mechanisms to ensure secure vehicular communications are detailed in section 2.2. We overview the anonymous authentication, location privacy, and revocation mechanisms in sections 2.4, 2.5, 2.6, respectively. Finally, we summarize the Chapter in section 2.7.

2.1 Security Threats

VANET applications are either safety-related or comfort applications which are respectively related to people's lives and financial transactions. If vehicular communications are not secured, a number of attacks can be launched, which may affect the reliability of the system

and lead to lives or financial losses. Consequently, it is mandatory to study the different types of attacks in VANETs in order to well-design the adopted security mechanisms such that they can thwart all the possible attacks and provide reliable and robust services. Threats can be classified according to the type of the targeted application [13], the targeted security service [14], or the type of the threat itself [15].

Laurendeau *et al.* [14] classified the threats in VANETs according to the targeted security service as follows:

1. Threats to authentication: Authentication is concerned with protecting the network from malicious users whether internal (legitimate) or external (illegitimate). The threats to authentication can be summarized as follows.

    - Masquerading: For masquerading attack, a malicious node impersonates a legitimate node by fabricating an identity or stealing a working OBU from a legitimate user. Then, the attacker launches more attacks such as injecting false messages or blackhole attacks.
    - Sybil attack: This attack is one of the serious attacks to VANETs, where a malicious vehicle uses multiple identities to claim several vehicles at the same time within an area. As a result, it has a severe impact on location-based services, and it can even leads to a threat to lives of other drivers.
    - Blackhole: For this attack, an insider node deliberately drops all the received messages which are destined to other nodes, thus, forming a blackhole in the network. A group of nodes can collude together to partition a segment of the network by forming a set of blackholes around this segment, thus, preventing the delivery of any message to any node in this segment.

- Greyhole: Greyhole attack is a variant of blackhole attack, where an attacker selectively forwards some messages but not all the messages. This kind of attack is difficult to detect because it is indistinguishable from normal packet dropping event due to poor network condition, for example, when the fragmentation of VANET is frequent.
- Replay attack: An attacker records a message at some time and replays it at a later time to get specific benefits such as causing a user to be double charged for the same service.
- GPS Spoofing: An attacker can use GPS satellite simulator to generate fake GPS signals causing vehicles to get erroneous location information, which can lead to serious accidents. Another possibility is that if the vehicles use the spoofed GPS clock to generate the time-stamps attached to the transmitted messages, replay attacks can be easy to a large degree.
- Broadcast tampering: An attacker can broadcast fake information about the traffic conditions to cause accidents or let other vehicles clear the way for the sake of the attacker.
- Transaction tampering: For this attack, an attacker tries to modify or fabricate a transaction query or reply causing an incorrect charge to the user.

2. Threats to confidentiality: Confidentiality is concerned with preventing unauthorized nodes from eavesdropping or inferring the data exchanged between legitimate users.

   - Eavesdropping: An eavesdropper can listen to the messages transmitted over the air. Then, the eavesdropper can build a complete profile about how often and when the services are used by a particular user.

- Location tracking: In safety applications, each user periodically broadcasts its location. An attacker can eavesdrop and manipulate these data to track the user and possibly infer sensitive information about the user such as the user's real identity which violates the user's privacy.

3. Threats to availability: Availability is concerned with making the services offered by VANET applications always available to legitimate users.

   - DoS: An attacker can deliberately flood or jam the communication channel with falsified data to disrupt the communication between legitimate nodes and possibly prevent the service altogether.
   - Malware: An attacker can insert a malware, e.g., virus, worm, trojan, etc., in the firmware update messages broadcast by the network. As a result, the malware could be installed in the operating system of the vehicles or the RSUs, which can disrupt their performances.
   - Spamming: In this attack, an attacker can continuously broadcast spamming data in an area to make the service (or the application) offered by VANET in this area experience long delays, hence, degrading the performance of the offered service.

Raya *et al.* [13] classified the threats to vehicular networks according to the type of the targeted application into three main categories:

1. Attacks on safety-related applications: The result of such kind of attacks ranges from traffic congestion to accidents and losses of lives.

2. Attacks on payment-based applications: Vehicular networks support applications such as toll collection and location-based services, which include some financial trans-

actions. This means that this type of applications will be targeted by many attackers. Attacks on payment-based applications usually results in financial frauds.

3. Attacks on privacy: Vehicular networks can make the tracking of vehicles easy because vehicles can be tracked from their transmitted messages.

Raya *et al.* [15] gave a classification of the security threats in vehicular networks according to the type of the attacks as follows:

1. Jamming: An attacker jam the communication channel to prevent vehicles from communicating together.

2. Forgery: An attacker can forge a message to cause accidents or to get some rational benefits. The freshness and correctness of the exchanged messages between vehicles are very important to ensure that the received messages are not forged.

3. In-transit traffic tampering: The attackers in such kind of attacks deliberately drop, corrupt, or modify messages.

4. Impersonation: For this attack, the attacker aims to convince other vehicles that he is a legitimate vehicle. For example, the attacker can claim that he is an emergency vehicle to make other vehicles yield the road in front of him.

5. Privacy violation: If the attacker can collect enough messages from a vehicle, it may be easier to gain some knowledge about the personal information of the driver. This is considered a violation of the privacy of users.

6. On-board tampering: The attacker can tinker with the sources of information in the vehicle. For example, he can by-pass a sensor or put some ice around the temperature sensor to force the vehicle to send bogus warning of icy road.

2.2 Security Requirements

In order to protect VANETs against the threats mentioned in the previous section, the security mechanisms employed in VANETs should satisfy the following security requirements [4] [16] [17] [18]:

1. Authentication: Entity authentication is required to prevent illegitimate users from injecting bogus messages into the network. Each vehicle in the network should possess an authentic identity. When a vehicle receives a message, it first checks the authenticity of the sender's identity before performing further processing to the received message. Besides entity authentication, data authentication is also a concern to ensure that the contents of the received data is neither altered nor replayed.

2. Non-repudiation: Non-repudiation is necessary to prevent legitimate users from denying the transmission or the content of their messages. Users anticipate the network to provide a high level of liability, where a vehicle involved in a crash should be efficiently identified. Liability can be achieved by investigating the messages saved in each vehicle involved in the crash. However, if non-repudiation cannot be guaranteed, this process will be infeasible.

3. Privacy: Providing privacy is mainly related to preventing the disclosure of the real identities of the users and their locations information. Privacy can be provided by introducing identity anonymity such that any observer could neither identify the real identity nor correlate the real identity with the current location of any user. An observer is an attacker launching tracking attacks by installing receivers on the roads to eavesdrop the messages broadcast by the OBUs. By trying to correlate some of the broadcast certificates to an OBU, the observer may be able to track that OBU.

4. Access control: Access control is necessary to ensure reliable and secure operation of the system. Any misbehaving entity should be revoked from the network to protect the safety of other legitimate entities in the network. Moreover, any actions taken by that misbehaving entity should be canceled.

5. Data consistency: Data consistency is important requirement for safety applications to protect the users against malicious insider users. In general, vehicular networks have honest majority of users. However, if one of the vehicle's sensors becomes malfunction, it will start transmitting messages with wrong information. The same situation applies to legitimate users who have malicious behavior. Such kind of users can affect the network operation by generating rogue messages. The danger of these faulty messages is that they have been issued by legitimate users, which means that they are authentic messages yet containing falsified data. Misinterpreting these falsified messages as valid ones can lead to harmful actions taken by the vehicles. To mitigate the effect of such kind of messages, authentic messages received by a vehicle should be cross-checked with other received messages from other vehicles to ensure data consistency.

6. Availability: Users may be frustrated if the VANET services become temporarily unavailable due to attacks such as Denial of Service (DoS). Availability means that messages not only reach all the targeted recipients but also reach at the correct time. However, the acceptable latency differs according to the type application. In other words, VANET applications should be prioritized according to the bounds of the acceptable delay in each application. Vehicles are typically moving with high speeds, which means that they cannot be involved in long term communication sessions. This is another factor that should be taken into consideration.

2.3 Security Mechanisms

In this section, we discuss the guidelines of the basic security mechanisms to meet the security requirements mentioned in the previous section. The security requirements of the data authentication and non-repudiation can be achieved by using digital signature. To implement digital signatures, asymmetric cryptography scheme is required, in which each entity has a public/private key pair. In asymmetric cryptography schemes, an entity uses its unique private key for generating a unique digital signature for every outgoing message. When a signed message is received, the recipient uses the public key of the sender to verify the digital signature of the sender on the message. Successful digital signature verification implies that the content of the message is not altered, and the sender is the only one who can generate this message, i.e., achieving data authentication and non-repudiation. To achieve entity authentication, the public key of each entity must be authentic and all the entities in the network should be able to validate its authenticity. From the aforementioned discussion, it is clear that securing VANETs requires Public Key Infrastructure (PKI), where each entity in the network possesses an authentic certificate generated by a trusted Certification Authority (CA). Eq. (2.1) shows the fundamental elements in an authentic certificate $cert_i$ generated by a CA for an entity i, where ID_i is the identity of i, PK_i is the public key of i, V_i is the validity period of the certificate $cert_i$, and $sig_{CA}(ID_i||PK_i||V_i)$ is the signature of the CA, using the CA private key SK_{CA}, on the concatenation $(ID_i||PK_i||V_i)$.

$$cert_i = (ID_i, PK_i, V_i, sig_{CA}(ID_i||PK_i||V_i)) \tag{2.1}$$

Any entity can verify the certificate $cert_i$ by verifying the signature $sig_{CA}(ID_i||$ $PK_i||V_i)$ using the public key PK_{CA} of the CA, which is known to all the entities in the

network. The purpose of the certificates generated by the CA is to bind the identity of the certificate holder to its public key in an authentic way. Any entity in VANETs should sign any outgoing message M. The message format (msg_{format}) of an outgoing message is indicated in Eq. (2.2), where $sig_i(M)$ is the signature of i on the message M using the private key SK_i corresponding to the public key PK_i included in $cert_i$.

$$msg_{format} = M||sig_i(M)||cert_i \tag{2.2}$$

Any vehicle receiving a message in the format given in Eq. (2.2), first, verifies $cert_i$ using the CA public key PK_{CA} to achieve entity authentication. Then, it verifies the signature $sig_i(M)$ using the public key PK_i included in the certificate $cert_i$ of entity i to achieve data authentication and non-repudiation since entity i is the only entity that can generate the signature $sig_i(M)$ using its private key SK_i.

The mechanism that can be used to provide access control in PKI is the revocation of misbehaving nodes. Revocation can be achieved by employing Certificate Revocation Lists (CRLs). A CRL is a list containing the identities of the certificates of the misbehaving nodes. Before verifying any received message, each node checks whether or not the sender is included in the up-to-date CRL.

Data consistency can be simply achieved by cross-checking the received messages with the previously received messages and excluding the anomaly ones. According to availability, there is no clear mechanism to defend the network against availability attacks, especially DoS attacks. However, a possible solution is to switch to other communication technologies, e.g., cellular communications, FM, etc., when a DoS attack is detected.

So far we have discussed the security mechanisms used to meet the VANETs' security requirements except for the privacy requirement. In traditional PKI, binding a fixed iden-

tity of the certificate holder to a fixed public key can lead to a severe violation of the users' privacy as tracking a user can be trivially performed by linking the unique certificate identifier included in the broadcast messages by that user. It should be noted that the WAVE standard [2] details how PKI can be used to provide authentication. However, WAVE does not mention how privacy preservation can be achieved. Consequently, how can the privacy of the users be preserved in PKI? One viable solution for preserving the privacy of the users is to use anonymous authentication, where the authentication of an entity can be verified without disclosing the real identity of that entity. However, if VANETs offer absolute privacy preservation for the users, the liability of VANETs will be violated. Consequently, VANETs should provide conditional privacy preservation such that only a trusted entity is allowed to reveal the real identity of the users after getting a juridical permission. In the following sections, we discuss the state of the art for achieving anonymous authentication, location privacy preservation, and revocation. It should be noted that through the rest of the chapter the words vehicle, OBU, and node will be used interchangeably.

2.4 Anonymous Authentication

The widely acceptable mechanisms to achieve anonymous authentication are pseudonyms, group signatures, and hybrid (i.e., using pseudonym and group signatures). In this section, we discuss the details of these mechanisms indicating the advantages and disadvantages of each mechanism as follows.

2.4.1 Pseudonyms

Instead of binding the real identity of the certificate holder to its public key in the certificate issued by the CA, a pseudonym, i.e., pseudo identity, can replace the real identity of the certificate holder in the issued certificate as shown in Eq. (2.3), where PID_i is a pseudonym for entity i.

$$cert_i = (PID_i, PK_i, V_i, sig_{CA}(PID_i||PK_i||V_i)) \tag{2.3}$$

The message format in the pseudonym anonymous authentication is indicated in Eq. (2.2). It should be noted that the words pseudonym and pseudo identity will be used interchangeably throughout of rest of the thesis.

Only the CA can relate the authentic pseudonym PID_i to the real identity ID_i of the certificate holder, thus, providing conditional privacy preservation. Each entity periodically changes the used pseudonym certificate to confuse attackers. The shorter the period during which a certificate is used the higher the privacy protection. Consequently, vehicles will consume a large number of certificates in a short period, and they will need to get fresh certificates. Providing new certificates for the vehicles in VANETs is a very challenging issue. In addition, the revocation of certificates of the misbehaving vehicles in PKI is performed using Certificate Revocation Lists (CRLs). Since the scale of VANETs is very large and each vehicle periodically changes its certificate, the total number of the used certificates in the network will be huge. As a result, the problem of certificate revocation is another challenge that should be carefully considered when investigating the anonymous authentication problem in VANETs. The proposed mechanisms in the literature to update the pseudonym certificates can be classified to either central or distributed mechanisms as follows.

1. **Central certificate update:** In this mechanism, the CA is the only entity that can

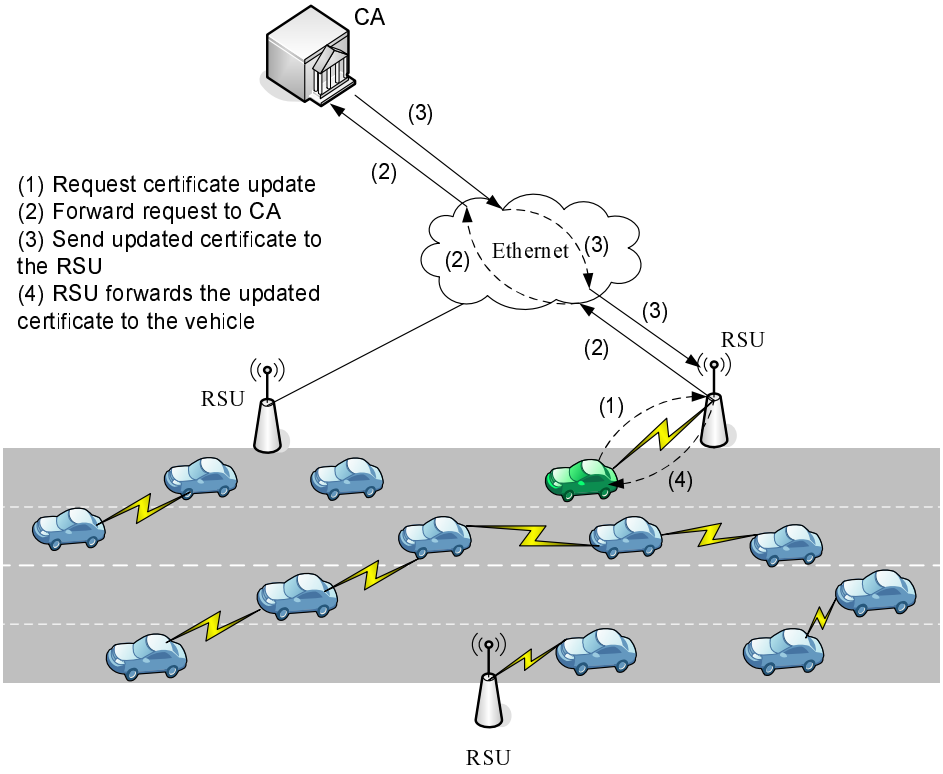


Figure 2.1: On the fly central certificate update

issue the certificates for the vehicles. The CA can generate the certificates using a static or on the fly mechanisms as follows.

- *Static:* In the static mechanism, proposed by Raya *et al.*, the CA loads a huge number of anonymous certificates in each vehicle during the initial registration of the vehicle [4]. The number of the loaded certificates should be large enough to suffice the usage for a long period, e.g., one year. Each vehicle can update its anonymous certificate set during the annual inspection of the vehicle. The advantage of this mechanism is its simplicity in generating the pseudonym certificates for the vehicles since it does not depend on the infrastructure RSUs.

On the other hand, the disadvantage of this mechanism is the requirement of huge database at the CA to maintain the relation between the real identity and pseudonyms for all the vehicles. In addition, the revocation becomes inefficient since revoking one vehicle implies revoking all the loaded certificates in that vehicle. Hence, the CRL size becomes very large, which constitutes inefficiency in terms of the CRL distribution.

- *On the fly central certificate update:* In this mechanism, a vehicle that needs to get a new certificate sends a request to the CA via the nearest RSU as shown in Fig. 2.1. Then, the CA issues a new certificate for that vehicle and returns it via the same RSU. It should be noted that the RSUs and the CA in one domain are interconnected through an Ethernet [2] as indicated in Fig. 2.1. In this mechanism, each vehicle only possesses a few certificates compared to the static mechanism. The advantage of this method is avoiding the large increase in the CRL size since revoking one vehicle implies revoking a small number of certificates. It also should be noted that the certificates with expired validity are self-revoked and revocation is only needed for certificates with a valid validity period. The disadvantages of this method are: (1) Each CA encounters a large number of certificate update requests which can render the CA a bottleneck; (2) The certificate update delay is long relative to the short V2I communication duration between the immobile RSUs and the highly mobile OBUs during which the new certificates should be delivered to the requesting OBU. The long certificate update delay is due to the fact that a request submitted by an OBU to an RSU must be forwarded to the CA, and CA has to send the new certificate to that RSU which in turn forwards the new certificate to the requesting OBU. It should be noted that this mechanism is the adopted certificate update

mechanism in the WAVE standard [2].

2. **Distributed certificate update:** In this mechanism, rather than depending solely on the CA to update the certificates of the vehicles, the RSUs can be authorized to update the certificates of the vehicles in a distributed manner. The work done in chapter 3 falls into this category. The details of this work will not be presented here to avoid redundancy. The advantages of this mechanism are avoiding the bottleneck performance in the CA due to its distributed nature, and it is fast in issuing the certificates for the vehicles since the communication with the CA is avoided. On the other hand, the disadvantage of this mechanism is that it requires a minimum number of RSUs to be deployed to perform well.

2.4.2 Group Signatures

Group signature (GS) [19] is one of the widely acceptable PKI techniques for achieving anonymous authentication. GS allows any group member to sign a message on behalf of the group using its private key without revealing its identity. Any pair of signatures generated by the same group member cannot be linked together by any entity except the trusted group manager (TGM). The TGM is a central trusted entity. The TGM is almost identical to a CA, however, the TGM does not issue certificates for the nodes but instead it issues GS private keys (SK_i^{GS}'s) for the group members and a single group public key (PKG) for all the group members. The message format msg_{format} of an outgoing message in the GS schemes is shown in Eq. (2.4).

$$
\begin{aligned}
&msg_{format} = M||sig_i^{GS}(M), \text{ where} \\
&sig_i^{GS}(M) = \{enc(ID), ZKP(SK_i^{GS})\}, \text{ and} \\
&ZKP(.) = f(M)
\end{aligned}
\tag{2.4}
$$

The group signature $sig_i^{GS}(M)$ on a message M consists of two parts [20]: First, an encrypted identity ($enc(ID)$) of the signer, where the value of $enc(ID)$ is function in a random number, i.e., it is different for each signature; Second, a zero knowledge proof ($ZKP(SK_i^{GS})$) of the private key SK_i^{GS}, where the signer proves to the recipient that it possesses a valid private key (SK_i^{GS}) without revealing any information about that private key. It should be noted that the zero knowledge proof ($ZKP(.)$) is function in the underlying message M. Any signed message can be verified using the group public key (PKG). The revocation in GS is performed by distributing Revocation Lists (RLs) containing the identities of the revoked users. The GS is an excellent way of achieving anonymous authentication as the only identifier in the signature on a message is the encrypted identity of the signer which is different for each message compared to the pseudonym certificate mechanism where the same certificate is used for a number of messages. However, the GS schemes suffer from a large computation overhead in the signature verification process, which limits the number of certificates that can be verified in a given duration. Lin *et al.* [21] proposed a secure and privacy preserving protocol for vehicular communications (GSIS) by employing the short group signatures [20] and the identity-based signatures [22], where all the OBUs are members of the same group and each OBU has a unique private group key used to sign the outgoing messages. It should be noted that in this case there is no certificates. Then, GSIS eliminates the need for certificates update. A similar approach to GSIS is proposed in [23]. The advantage of these schemes is the simplicity of the

revocation process, where each vehicle has only one private key, compared to the usage of pseudonym certificates, where each vehicle has a set of certificates. In other words, these schemes use the RL instead of the CRL, where the size of the RL is much smaller than that of the CRL. The main disadvantage of these schemes is that they inherit the large computation overhead of GS.

2.4.3 Hybrid Schemes

To overcome the drawback of the large computation overhead of the GS schemes, the hybrid schemes can be used. The main idea of the hybrid schemes is to use the private keys of the GS scheme to issue pseudonym certificates for the vehicles. The proposed schemes in the literature can be classified according to the level at which the GS scheme is used as follows.

1. **At the RSU level:** In [24], Lu *et al.* proposed efficient conditional privacy preservation protocol for secure vehicular communications (ECPP). In ECPP, GS is used at the RSU level, where all the RSUs in the network form the group members, i.e., each RSU has a private group key (SK_i^{GS}). The authentication at the vehicles level is based on pseudonym certificates. When a vehicle i enters the coverage area of an RSU j, RSU j may generate a short-lifetime pseudonym certificate for that vehicle i on the fly. The fundamental elements of the generated short-lifetime certificate are indicated in Eq. (2.5).

$$cert_i = (PID_i, PK_i, V_i, sig_j^{GS}(PID_i||PK_i||V_i)) \tag{2.5}$$

   It should be noted that $sig_j^{GS}(PID_i||PK_i||V_i)$ is the group signature of RSU j on

the pseudonym certificate of vehicle i. Any vehicle receiving the certificate in Eq. (2.5) can verify it by checking the signature $sig_j^{GS}(PID_i||PK_i||V_i)$ using the group public key PKG. Also, vehicle i will use its secret key to sign the outgoing message as indicated in Eq. (2.2), where the signature of vehicle i can be verified using its public key PK_i included in the certificate shown in Eq. (2.5). The advantage of this scheme is the simplifying of the revocation process by employing the RL instead of the CRL. Since each pseudonym certificate generated by an RSU includes one group signature, the disadvantage of this scheme is the large computation overhead in the signature verification process inherited from the GS scheme. Another disadvantage is the requirement of a minimum number of deployed RSUs to achieve good performance.

2. **At the OBU level:** In [25], Calandriello *et al.* proposed to use GS at the OBU level, where all the OBUs in the network form the group members, i.e., each vehicle has a private group key SK_i^{GS}. The private key of each vehicle is not used for signing outgoing messages as that in GSIS, but instead, it is used for signing a short-lifetime pseudonym certificate for itself, i.e., the pseudonym certificates are self-generated. For example, vehicle i can generate a pseudonym certificate for itself as shown in Eq. (2.6), where $sig_i^{GS}(PID_i||PK_i||V_i)$ is the signature of vehicle i on the contents of its certificate $cert_i$ using its group private key SK_i^{GS}.

$$cert_i = (PID_i, PK_i, V_i, sig_i^{GS}(PID_i||PK_i||V_i)) \tag{2.6}$$

Any vehicle receiving the certificate in Eq. (2.6) can verify $sig_i^{GS}(PID_i||$ $PK_i||V_i)$ using the group public key PKG. The generated pseudonym certificates are used for signing the outgoing messages. The advantage and the disadvantage of

this scheme are the same as that of ECPP except for the requirement of deploying a minimum number of RSUs.

2.4.4 Additional Considerations Related to Authentication

There are some issues directly related to anonymous authentication in VANETs that should be considered as follows.

Enhancing authentication efficiency

The authentication efficiency in VANETs can be enhanced by the following measures.

1. **Batch verification**

   According to the Dedicated Short Range Communication (DSRC) [3], each OBU periodically broadcasts a safety beacon message every $100 \sim 300$ msec, where entity authentication and message integrity can be achieved by verifying the certificate and digital signature of the sender in the pseudonym and hybrid mechanisms, and only the digital signature of the sender in case of the GS mechanism. In dense traffic areas, each OBU will receive a large number of the safety beacon messages in a short duration, and thus the ability to verify a large number of certificates and/or signatures in a specific period poses an inevitable challenge to the employed authentication mechanism. Consequently, batch verification, where a number of messages can be simultaneously verified, is a good candidate to enhance the authentication efficiency.

   There are some works in the literature that consider enhancing the authentication efficiency via batch verification such as [26] which presents an identity-based batch verification scheme. In this approach, the authentication of the OBUs is achieved via

identity-based authentication. The same master secret key is stored in the tamper-proof device of each OBU. However, tamper-proof devices may have serious flaws [27], which may jeopardize the network security if only one OBU is compromised.

2. **RSU-aided message authentication scheme (RAISE)**
   In order to alleviate the effect of the heavy computations associated with the asymmetric cryptography, Zhang *et al.* [28] proposed RSU-aided message authentication scheme (RAISE). In RAISE, a vehicle ($\mathcal{A}$) entering the coverage range of an RSU ($\mathcal{B}$) establishes a connection with that RSU, where RSU $\mathcal{B}$ sends to vehicle $\mathcal{A}$ a unique pseudonym and shared symmetric secret key between itself and vehicle $\mathcal{A}$. When vehicle $\mathcal{A}$ broadcasts a message in the coverage range of RSU $\mathcal{B}$, it does not sign the message but instead it calculates a keyed message authentication code (HMAC) using the shared secret key sent by RSU $\mathcal{B}$. When this message is received by the neighboring vehicles, they store this message without verifying it. The only entity that can verify this message is RSU $\mathcal{B}$ using the shared secret key between itself and vehicle $\mathcal{A}$. Then, RSU $\mathcal{B}$ informs all the vehicles in its coverage range that the message sent by vehicle $\mathcal{A}$ is valid. In summary, the authors employed HMAC, which is very fast and incurs light-weight computation overhead compared to the digital signature using asymmetric cryptography, to achieve data and entity authentication.

Roaming

An important issue in VANET that is almost overlooked in the literature is the roaming between areas administered by different authorities. The challenge is how to continuously provide the necessary security services to an OBU moving from one service area administered by a CA to another area administered by a different CA. The conventional solution

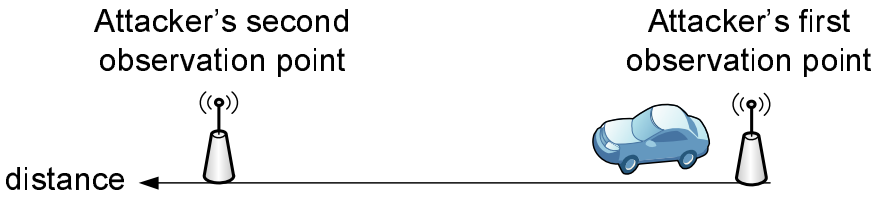


Figure 2.2: Tracking attack scenario

is that an OBU, entering the new service area, has to go to the CA of that service area to be loaded with the necessary security material for using the security services in this area. However, users may not be satisfied with such solution and a mechanism to provide transparent roaming between heterogenous domains should be developed.

2.5 Location Privacy

We have discussed in the previous section the different mechanisms proposed in the literature that can be used to provide anonymous authentication. The anonymous authentication can support identity privacy, but does it support location privacy? Before answering this question, we have to consider what an attacker can do to track a particular vehicle. In the following subsection, we consider a possible tracking scenario.

2.5.1 Tracking Attack

We consider an external passive global observer, which can overhear and correlate any message broadcast in clear in the network. The anonymity set is defined as the set of all possible OBUs which simultaneously change their anonymous certificate between two

observation points controlled by an attacker [29]. Consider an OBU moving between two observation points controlled by the global observer as shown in Fig. 5.2 [30]. The observer can track an OBU if two pseudonym certificates can be correctly correlated. This correlation can be achieved by capturing at least one message at each observation point from the OBU, while it is moving with the same speed and in the same lane for some distance between the observation points controlled by the observer. For example, a message is captured at the first observation point from an OBU moving with speed v, in lane L, and using pseudonym certificate $cert_1$. Given the speed of the OBU and the distance between the two observation points, the observer can expect the time to receive a message from that OBU at the second observation point, say after time t. If a message is captured at the second observation point after time t from an OBU moving with the same speed v, in lane L, and using pseudonym certificate $cert_2$, the observer can conclude that $cert_1$ and $cert_2$ belong to the same OBU. Also, if the OBU under attack is the only OBU, which changes its certificate in the area between the two observation points, i.e., it has anonymity set size being equal one, the observer can track that OBU even if it changes its speed or lane. It can be seen from the previous tracking attack that location privacy can be achieved only if the anonymity set size is greater than one and the OBUs, who changed their certificates, change their speeds and/or their lane locations. In addition, periodically changing the pseudonym certificate of each OBU in PKI is insufficient to provide location privacy for VANETs. It is clear that the previous tracking attack is applicable to the pseudonym certificates and the hybrid anonymous authentication mechanisms as they depend on the certificates. In addition, this tracking attack is also applicable to GS anonymous authentication mechanism as follows: Each group signature contains an encrypted version of the identity of the entity generating that signature [20]. Although, the identity is encrypted, it still can be used as an identifier of the message originator, and it can be used to track a particular vehicle

exactly as described above. It is worth mentioning that the GS mechanism and the hybrid mechanism which applies GS at the OBU level can provide a higher degree of location privacy compared to its counterparts. This is because a vehicle employing these techniques can change the identifying part of its signature for each signed message, thereby, provide more confusion to the attackers. However, these techniques cannot provide perfect location privacy as they still subject to the above attack. In addition, changing the identifiers of the vehicles in each message has a severe impact on the message delivery ratio [31]. According to [31], a vehicle should use the same identity for periods greater than 30 *sec* to achieve satisfactory message delivery ratio. This requirement limits the advantages of employing the GS schemes for anonymous authentication. From the aforementioned discussion, we can conclude that the answer to the question: "Does anonymous authentication provide location privacy?" is no, and anonymous authentication mechanisms should be combined with other mechanisms to achieve robust location privacy. In the next subsection, we discuss the different mechanisms proposed in the literature to achieve location privacy.

2.5.2 Location Privacy Preservation Techniques

There are several proposals in the literature addressing the problem of location privacy in VANETs as follows.

Random silent periods

Sampigethaya et al. [32] proposed to use random silent periods, where each vehicle opt to remain silent for a random period, to protect the location privacy of the vehicles. The concept of random silent period is illustrated in Fig. 2.3. The attacker in Fig. 2.3 captures two messages from vehicles with identifiers x and y, respectively, where the identifier is

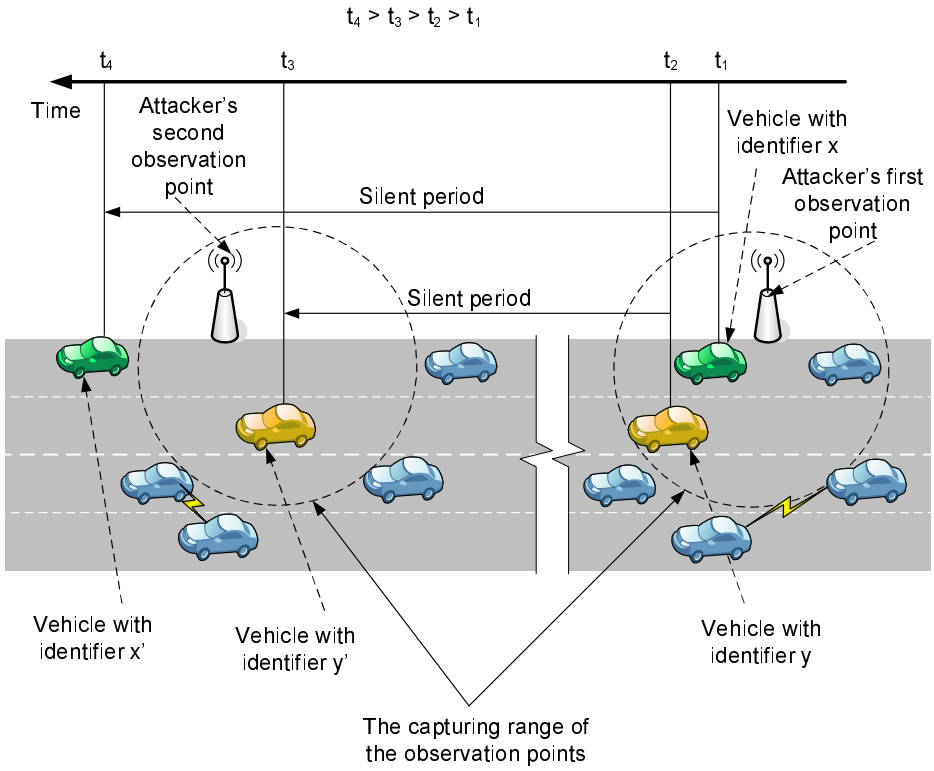


Figure 2.3: Random silent periods

either a pseudonym or the encrypted identity in the GS schemes. After that, vehicles x and y opt to remain silent for random periods as indicated in Fig. 2.3. Then, vehicles x and y change their identifiers to x' and y', respectively. The vehicles x' and y' will not transmit messages until the end of the silent periods. When the vehicle y' broadcasts a message, the broadcast message will be captured in the second observation point of the attacker, however, the attacker will be confused whether the identifier y' belongs to either the owner of the old identifier x or y as the attacker did not receive a message from the vehicle x' since the vehicle x' remains silent. Hence, random silent periods improve the location privacy level. It can be seen that the benefits of the random silent period in the

safety applications are limited as the vehicles have to transmit safety beacon messages every $100 \sim 300$ $msec$. As a result, the maximum silent period is limited to the duration between the broadcast of the safety beacon messages.

To further extend the period during which a vehicle can remain silent, random silent periods can be combined with group communications for VANETs applications excluding safety-related applications. Each group of neighboring vehicles forms a communication group, where the group leader acts as a proxy to all the group members. Combining group communications and random silent periods reduces the number of messages broadcast by the vehicles, hence, increasing the silent period and reducing the probability of being tracked. During the formation of a new group, there is a need to contact with a central registration authority. Hence, the proposed technique requires an online registration authority. Such requirement may not be feasible in a large scale network like VANET. In addition, when a vehicle updates its pseudonym, it has to leave the group and send a request to the group leader to rejoin the group. This may cause a large number of joining requests.

Cryptographic MIX-zones (CMIXes)

Freudiger et al. [33] used Cryptographic MIX-zones (CMIXes) at selected road intersections to provide location privacy. In CMIX, an RSU at an intersection securely provides a symmetric key to any approaching vehicle to establish what is called mix zone. All the data exchanged in the mix zones is encrypted by that symmetric key. In addition, all the vehicles in the mix zone are forced to change their anonymous certificates. As a result of the forced certificate change and the random direction change of each vehicle at road intersections, an attacker on the roadside cannot link a certificate to a particular vehicle, hence, providing location privacy. The concept of CMIXes is illustrated in Fig.

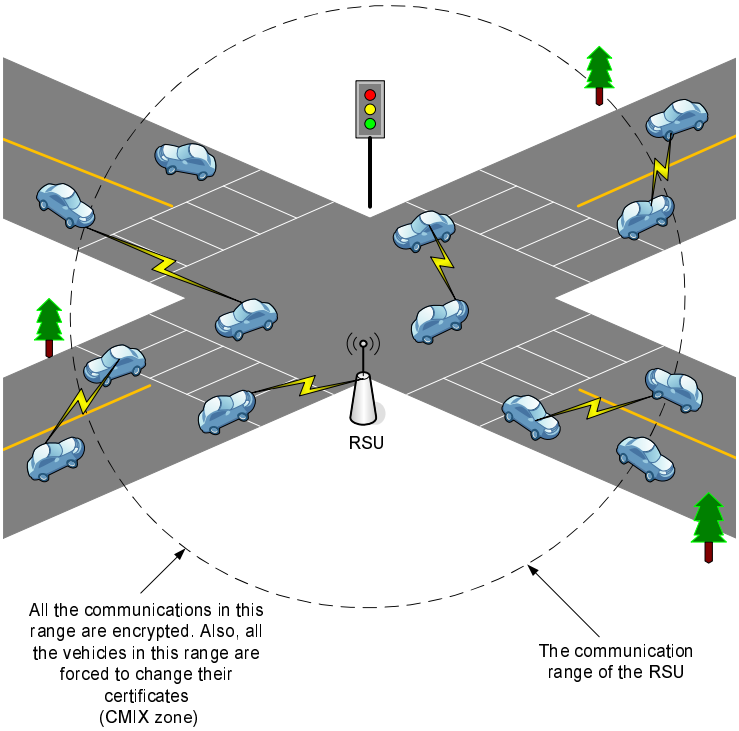


Figure 2.4: The cryptographic MIX-zones

2.4. The accumulation of CMIXes throughout the vehicular network forms what is called mix-network, which maximizes the degree of the location privacy.

2.6 Revocation

Revocation is the basic mechanism to revoke misbehaving vehicles. Revocation can be classified into centralized and decentralized revocation according to the authority taking the revocation decision. In this section, we overview the different centralized and decentralized revocation mechanisms.

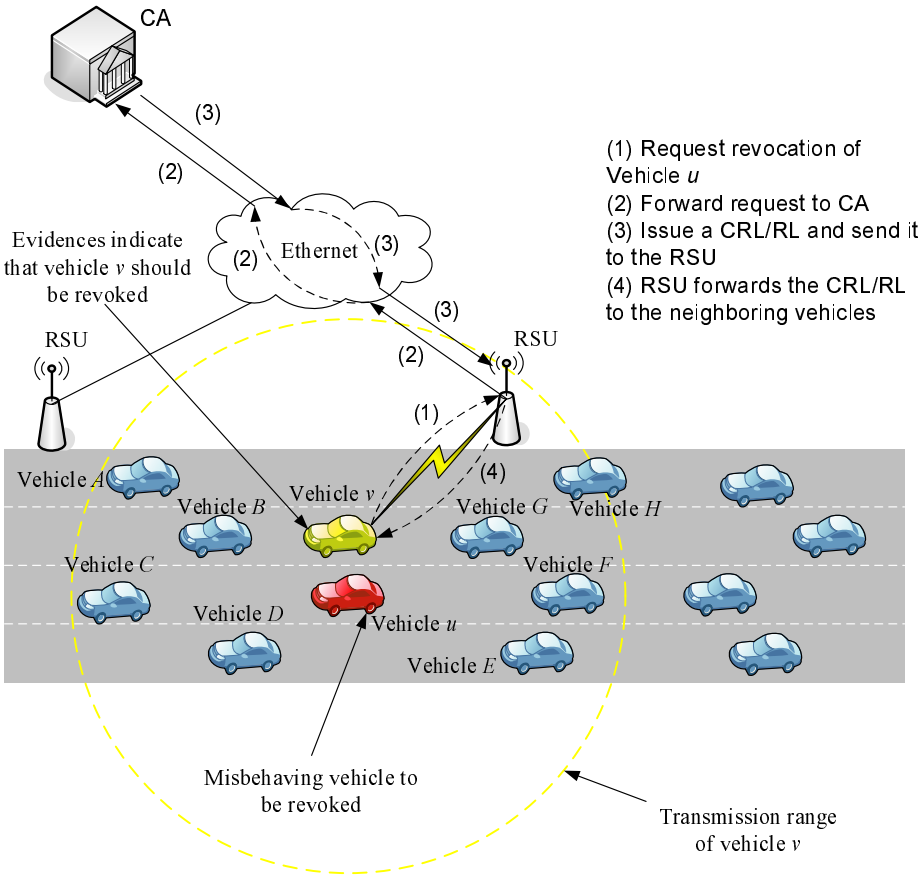


Figure 2.5: Different revocation scenarios

2.6.1 Centralized Revocation

For centralized revocation, a central entity, such as the CA, is the only entity in the network that can take the revocation decision for a certain node. The different mechanisms of centralized revocation in VANETs are discussed below.

CRL/RL

A CRL/RL is a list containing the identities (or pseudonyms) of the revoked vehicles. CRLs are used in the conventional PKI, while RLs are used mainly with GS schemes. In general, the size of a CRL is larger than that of an RL as the CRL will usually contain more than one certificate for each revoked vehicle, while RL contains only one entry for each revoked vehicle. According to the WAVE standard [2], vehicular networks depend on CRLs/RLs to achieve revocation. In such case, to revoke a vehicle, a CRL/RL has to be issued by the CA and broadcast by the infrastructure RSUs. The network scale of VANETs is expected to be very large. Hence, the distribution of CRLs/RLss is prone to long delays [11] [12]. Moreover, centralizing the revocation decision to the CA renders the CA a bottleneck.

A centralized CRL/RL revocation scenario is shown in Fig. 2.5, where vehicle u is misbehaving, and vehicle v collects enough evidences to accuse vehicle u as a misbehaving vehicle. Hence, vehicle v sends a revocation request to the CA via the nearest RSU. After the request reaches the nearest RSU, the request will be forwarded through the RSUs' Ethernet to the CA. When the revocation request reaches the CA, it verifies the request and the evidences, and generates a new signed CRL/RL. Based on the severeness of the misbehavior, the CA may include all certificates of the misbehaving vehicle in the CRL/RL. Then, the CA broadcasts the generated CRL/RL via the infrastructure RSUs.

In VANETs, the most important issue in any revocation method is the delay of delivering the revocation message to the neighboring vehicles of a misbehaving vehicle to prevent that misbehaving vehicle from jeopardizing the safety of its neighbors.

Revocation Using Compressed Certificate Revocation Lists (RC2RL) and Revocation of the Tamper-Proof Device (RTPD)

Raya *et al.* [34] proposed two centralized revocation techniques of a node by the CA: (1) RC2RL (Revocation using Compressed Certificate Revocation Lists), where the traditional CRLs issued by the CA are adopted. However, a CRL is compressed using Bloom filters prior to its broadcasting. The main limitation of RC2RL is that the Bloom filter is lossy, which means that an innocent vehicle may falsely considered as a misbehaving one. (2) RTPD (Revocation of the Tamper-Proof Device), which is used in case that all the certificates of a vehicle are to be revoked. In such case, the CA sends a message to the tamper-proof device used to perform all the security functions, e.g., signing outgoing messages, in the designated vehicle informing it to stop all the security functions.

Misbehavior Detection System (MDS) and Local Eviction of Attackers by Voting Evaluators (LEAVE)

Raya *et al.* [35] proposed Misbehavior Detection System (MDS) and Local Eviction of Attackers by Voting Evaluators (LEAVE). MDS and LEAVE can be used to isolate misbehaving nodes before the revocation data from CA is available to all vehicles. In MDS, the misbehavior that can be identified by monitoring specific parameters of a node, and data anomalies that do not follow any known pattern are distinguished. In LEAVE, a group of neighboring vehicles perform a voting on the misbehavior of a specific vehicle. If the accumulation of the votes exceeds a predefined threshold, a warning message is broadcast to the neighboring vehicles informing them to ignore all the messages transmitted by the misbehaving vehicle. In this way, the neighbors of a misbehaving vehicle can quarantine the misbehaving vehicle until a centralized revocation is issued by the CA. Although

this method is decentralized in quarantining misbehaving vehicles, it makes the revocation decision centralized by the CA, which may render the CA a bottleneck.

2.6.2 Decentralized Revocation

For decentralized revocation, the certificate revocation of a misbehaving node is done by the neighboring nodes of the misbehaving node or the certificate is self-revoked due to its expiry. Some works in the literature addressed the decentralized revocation in VANETs as follows.

Short-Lifetime Certificates

Revocation can also be achieved by relying on certificates with short-lifetimes, where a certificate is automatically revoked after its lifetime expires. An Example of such systems is the ECPP [24] discussed previously. In this scheme, a vehicle gets a number of certificates from an RSU sufficient to protect its privacy until it meets another RSU on its way. If the number of the generated certificates per vehicle is one, then, the certificates will be self-revoked after its lifetime expires, and there is no need for other revocation techniques. However, if the number of the generated certificates is more than one, then, it may be necessary to employ CRL to revoke the generated certificates for the misbehaving vehicles as self-revocation may take long time to revoke all the certificates of the misbehaving vehicle.

2.7 Summary

In this Chapter, we have demonstrated that security is vital for the operation of VANETs, where we have surveyed the different security aspects in VANETs. We have identified the challenges for securing VANETs by discussing security threats and requirements in VANETs. Moreover, we have detailed the necessary security mechanisms to achieve anonymous authentication and location privacy for preserving the users' privacy in VANETs.

Chapter 3

DCS: An Efficient Distributed Certificate Service Scheme for Vehicular Networks

In this Chapter, we propose an efficient Distributed Certificate Service (DCS) scheme for vehicular networks. The proposed scheme offers a flexible interoperability for certificate service in heterogeneous administrative authorities, and an efficient way for any On-Board Units (OBUs) to update its certificate from the available infrastructure Road-Side Units (RSUs) in a timely manner. In addition, the DCS scheme introduces an aggregate batch verification technique for authenticating certificate-based signatures, which significantly decreases the verification overhead. Security analysis and performance evaluation demonstrate that the DCS scheme can reduce the complexity of certificate management, and achieve excellent security and efficiency for vehicular communications.

3.1 Introduction

As discussed in Chapter 2, in order to provide a practical certification service for VANETs, it is required for each OBU to efficiently update its certificate in a timely manner. The certification service should also be decentralized to enable VANET to efficiently process the expected large number of certificate update requests. Moreover, to protect the user privacy, the updated certificates should be anonymous and free from the key escrow issue.

Another important issue is the roaming between different domains [9],[10]. The OBUs should have the capabilities to roam between domains administered by different CAs. The Wireless Access in Vehicular Environments (WAVE) standard [2] did not consider the roaming between different domains, and the interopreability between different CAs is still an open problem that has not been previously tackled in the VANET literature.

According to the Dedicated Short Range Communication (DSRC) [3], which is part of the WAVE standard, each OBU in VANETs periodically broadcasts a message every 300 msec, where entity authentication and message integrity can be achieved by verifying the certificate and digital signature of the sender. In dense traffic areas, each OBU will receive a large number of messages in a short duration, and thus the ability to verify a large number of certificates and signatures in a specific period poses an inevitable challenge to the authentication technique.

To address the aforesaid security and performance issues, we introduce an efficient distributed certificate service (DCS) scheme for vehicular communications, which features the following properties.

1. **Scalability:** The DCS scheme is constructed in a hierarchical way, which enables any OBU to efficiently update its certificate from the available RSUs in a timely manner.

Thus, the DCS scheme offers a distributed certification service. The DCS scheme also offers a flexible inter-operability between different administrative authorities, and it enables OBUs certificates to be free from the key escrow. All such policies efficiently enhance the system scalability, especially when it is deployed in a large-scale and heterogeneous vehicular networks.

2. **Efficiency:** Considering the requirement for each entity to verify a large number of messages in a timely manner, DCS introduces an efficient batch verification technique, which enables any entity to simultaneously verify a mass of signatures and certificates. Thus, the DCS scheme significantly decreases the verification overhead.

Therefore, the DCS scheme can meet the security and efficiency requirements for certificate service in vehicular communications.

Our contribution in this Chapter is proposing an efficient Distributed Certificate Service (DCS) scheme which enables an OBU to update its certificate from any RSU no matter whether the RSU is located in the domain in which the OBU was originally registered or not. Consequently, an OBU is free to roam between domains administered by different authorities. Also, the DCS scheme considers batch verification of certificates and messages signatures. To the best of our knowledge, this is the first approach to address the roaming between different domains in VANETs. Also, the DCS scheme is the first to consider the integration between distributed certificate generation through RSUs and efficient message authentication using batch verification.

The remainder of the Chapter is organized as follows. In section 3.2, the preliminaries are discussed. The system design considerations in the proposed DCS scheme is investigated in section 3.3. The proposed DCS scheme is introduced in section 3.4. Section 3.5 introduces an efficient batch verification technique for authenticating certificate-based

message signatures. Section 3.6 and Section 3.7 respectively present the security analysis and performance evaluation for the proposed DCS scheme, followed by the summary in section 3.8.

3.2 Preliminaries

In this section, we introduce the bilinear pairings. The notations used throughout the Chapter are given in Table 3.1.

3.2.1 Bilinear Pairing

The bilinear pairing [22] is the foundation of the proposed DCS scheme. Let $\mathbb{G}_1$ denote an additive group of prime order q, and $\mathbb{G}_2$ a multiplicative group of the same order. Let P be a generator of $\mathbb{G}_1$, and $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ be a bilinear mapping with the following properties:

1. Bilinear: $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$, for all $P, Q \in \mathbb{G}_1$ and $a, b \in_R \mathbb{Z}_q$.
2. Non-degeneracy: $\hat{e}(P, Q) \neq 1_{\mathbb{G}_2}$.
3. Symmetric: $\hat{e}(P, Q) = \hat{e}(Q, P)$, for all $P, Q \in \mathbb{G}_1$.
4. Admissible: the map $\hat{e}$ is efficiently computable.

The bilinear map e can be implemented using the Weil [36] and Tate [37] pairings on elliptic curves. We consider the implementation of Tate pairing on an MNT curve [38] with embedding degree 6, where $\mathbb{G}_1$ is represented by 161 bits, and the order q is represented

Table 3.1: Notations for DCS

Symbol	Notation
CA_i, CA_w	two arbitrary CAs
RSU_j, RSU_l	two arbitrary $RSUs$
OBU_m, OBU_n	two arbitrary $OBUs$
s	master secret key of MA for secret key generation
α	partial secret signing-key for signing RSU certificates
γ	partial secret signing-key for signing OBU certificates
$P_\circ$	public key used to verify signatures on any message
$S_{\alpha i}$	CA_i secret key to sign RSU certificates
P_α	public key used to verify RSU certificates
$S_{\gamma j_i}$	RSU_j secret key, generated by CA_i, to sign OBU certificates
P_γ	public key used to verify OBU certificates
P_μ	public key used to verify any certificate
PK_i	public key for CA_i
SK_i	secret key for CA_i
PK_{j_i}	RSU_j public key generated by CA_i
SK_{j_i}	RSU_j secret key generated by CA_i
$cert_{RSU_{ji}}$	certificate for RSU_j generated by CA_i
$PK_{m_{ji}}$	OBU_m public key generated by RSU_j using PK_{j_i}
$SK_{m_{ji}}$	OBU_m secret key generated by RSU_j using SK_{j_i}
$vepriod$	OBU certificate validity period
$cert_{OBU_{mji}}$	OBU_m certificate generated by RSU_j using $S_{\gamma j_i}$
t_{stamp}	time stamp
H_1	hash function such that $\{0,1\}^* \in \mathbb{G}_1^*$
H_2	hash function such that $\{0,1\}^* \in \mathbb{Z}_q^*$

by 160 bits. The group order of $\mathbb{G}_1$ is defined as the number of the points on the employed elliptic curve. For an MNT elliptic curve with embedding degree 6 and the order q is represented by 160 bits, the group order of $\mathbb{G}_1$ is 4.5×10^{30} [1], which qualifies the bilinear pairing as a practical choice for securing the large scale VANETs.

The security of the proposed scheme depends on solving the following hard computational problems:

- **Elliptic Curve Discrete Logarithm Problem (ECDLP):** Given a point P of order q on an elliptic curve, and a point Q on the same curve. The ECDLP problem [40] is to determine the integer l, $0 \leq l \leq q-1$, such that $Q = lP$.
- **Computational Diffie-Hellman problem (CDH):** For two unknowns $a,\ b \in \mathbb{Z}_p^*$, the CDH problem [41] is given $aP,\ bP \in \mathbb{G}_1$, compute $abP \in \mathbb{G}_1$.

3.3 System Design Considerations in the Proposed DCS Scheme

In this section, we discuss the security objectives, system architecture, and network model of the proposed DCS scheme.

3.3.1 Security Objectives

In the DCS scheme, we aim to achieve the following security objectives.

[1] This result is obtained using MIRACL library [39].

1. Authentication: Entity authentication is required to prevent illegitimate users from injecting bogus messages into the network. Each vehicle in the network should possess an authentic identity. When a vehicle receives a message, it first checks the authenticity of the sender identity before performing further processing to the received message. Besides entity authentication, data authentication is a concern to ensure that the contents of the received data is neither altered nor replayed.

2. Non-repudiation: Non-repudiation is necessary to prevent legitimate users from denying the transmission or the content of their messages. Users anticipate the network to provide a high level of liability, where a vehicle involved in a crash should be efficiently identified. Liability can be achieved by investigating the messages saved in each vehicle involved in the crash. However, if non-repudiation cannot be guaranteed, this process will be trivial.

3. Privacy: Providing privacy is mainly related to preventing the disclosure of the real identity of the users and their locations information. Privacy can be provided by introducing identity anonymity such that any observer could neither identify the real identity nor correlate the real identity with the current location of any user. An observer is an attacker launching tracking attacks by installing receivers on the roads to eavesdrop the messages broadcast by the OBUs. By trying to correlate some of the broadcast certificates to an OBU, the observer may be able to track that OBU.

4. Transparent roaming: Users will not be satisfied if upon roaming between different network domains, they have to go to a central location to upload new security materials, e.g., keys, certificates, etc., to be able to use the VANET services. Transparent roaming is needed to ensure seamless operation of the OBUs in VANETs.

5. Access control: Access control is necessary to ensure reliable and secure operation of

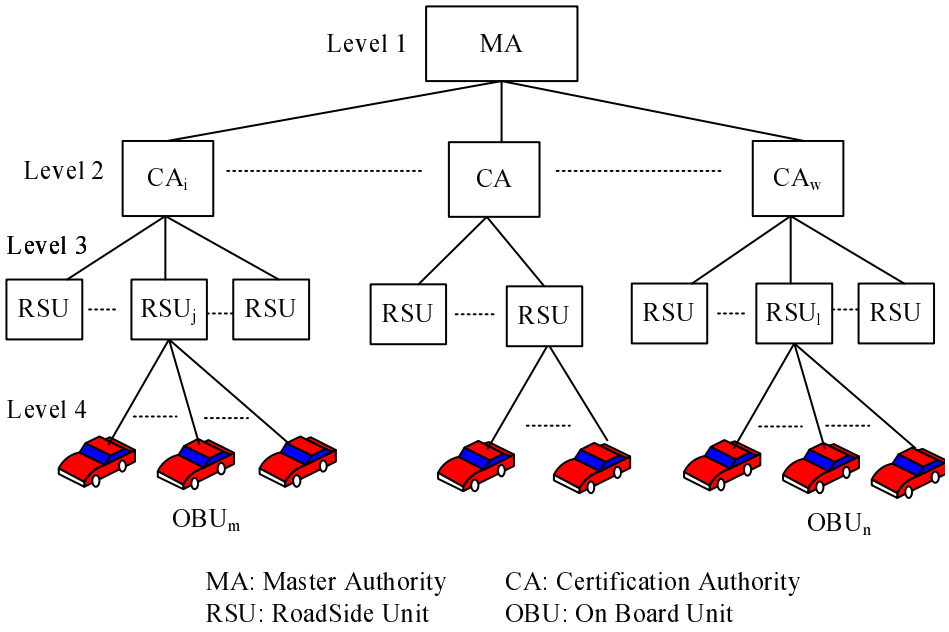


Figure 3.1: The proposed DCS hierarchical architecture

the system. Any misbehaving entity should be revoked from the network to protect the safety of other legitimate entities in the network. Moreover, any actions taken by that misbehaving entity should be canceled.

3.3.2 Architecture

The DCS hierarchical architecture, shown in Fig. 3.1, consists of four levels: the Master Authority (MA), which is the root of the system, is located at level 1; the Certification Authorities (CAs) are located at level 2; the Road Side Units (RSUs) and the On-Board Units (OBUs) are located at level 3 and level 4, respectively. In this architecture, entity authentication for RSUs and OBUs is achieved using certificate-based authentication, while that for CAs is achieved using identity-based cryptography [22].

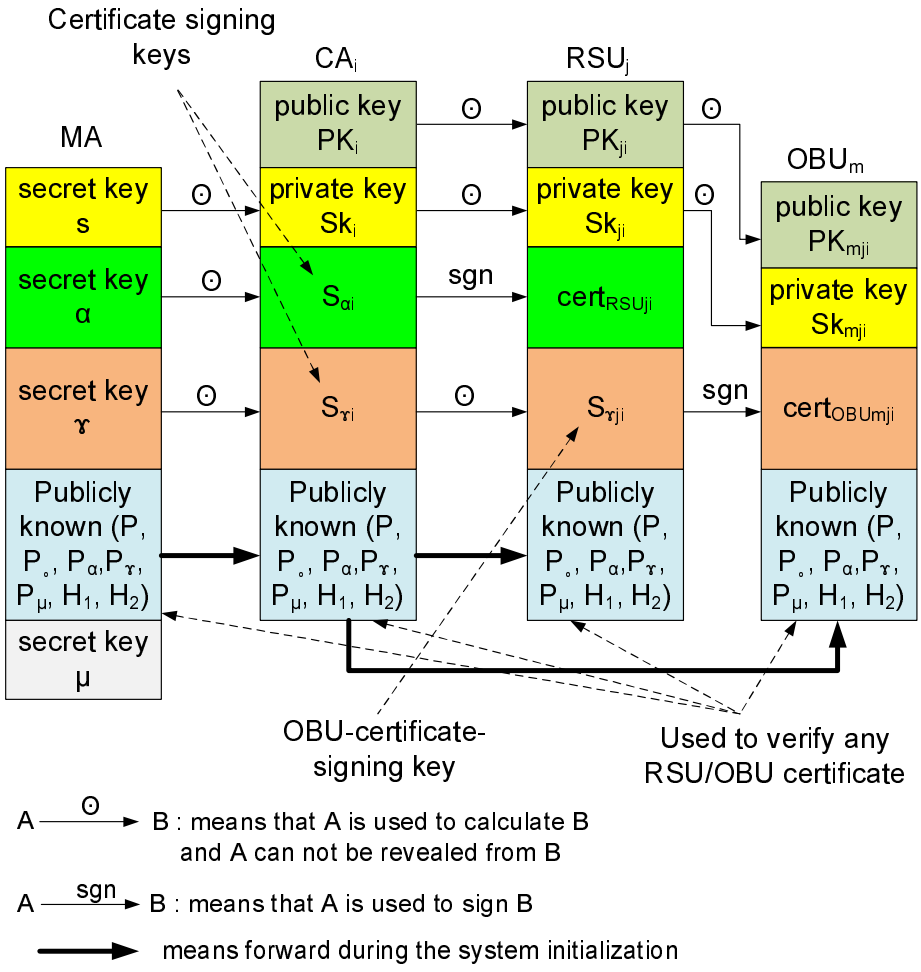


Figure 3.2: The relations of different keys among the network entities in the DCS scheme

Basic Operation of the DCS Scheme

The basic operation of the DCS scheme is as follows.

- The MA is in charge of generating public verification keys for verifying any RSU/OBU certificate. It also generates a public/private key pair for each CA, for signing the outgoing messages and verifying the incoming messages. Moreover, it generates two

secret certificate-signing keys for each CA;

- A CA uses the first certificate-signing keys, issued by the MA, to sign a certificate set for each RSU in its coverage area. Each certificate in the RSU certificate set is shared among a group of RSUs. The CA uses the second certificate-signing key as a partial signing key to generate secret OBU-certificate-signing keys for each RSU;
- An RSU uses the OBU-certificate-signing key to generate short lifetime anonymous certificates for any OBU. The public verification keys can be used by any entity to verify the certificate of any OBU or RSU regardless of the issuer of that certificate. In this way, any OBU can roam transparently between the coverage areas of different CAs. The certificate generation in DCS is derived from the signature schemes proposed in [42], [43].

Fig. 3.2 shows the relations of different keys among the network entities in the DCS scheme.

It should be noted that the hierarchical architecture and the basic operation of DCS conform with that of the WAVE standard as follows. The Wave standard considers a root CA, which is responsible only for authorizing other non-root CAs. The function of the root CA in the Wave standard is comparable to that of the MA in the DCS scheme. The functions of the non-root CAs, RSUs, and OBUs in the WAVE standard and the DCS scheme are similar. Consequently, DCS adopts a realistic hierarchical architecture which conforms to the current standard of VANET.

3.3.3 Network Model

As shown in Fig. 3.3, the network model under consideration consists of the followings.

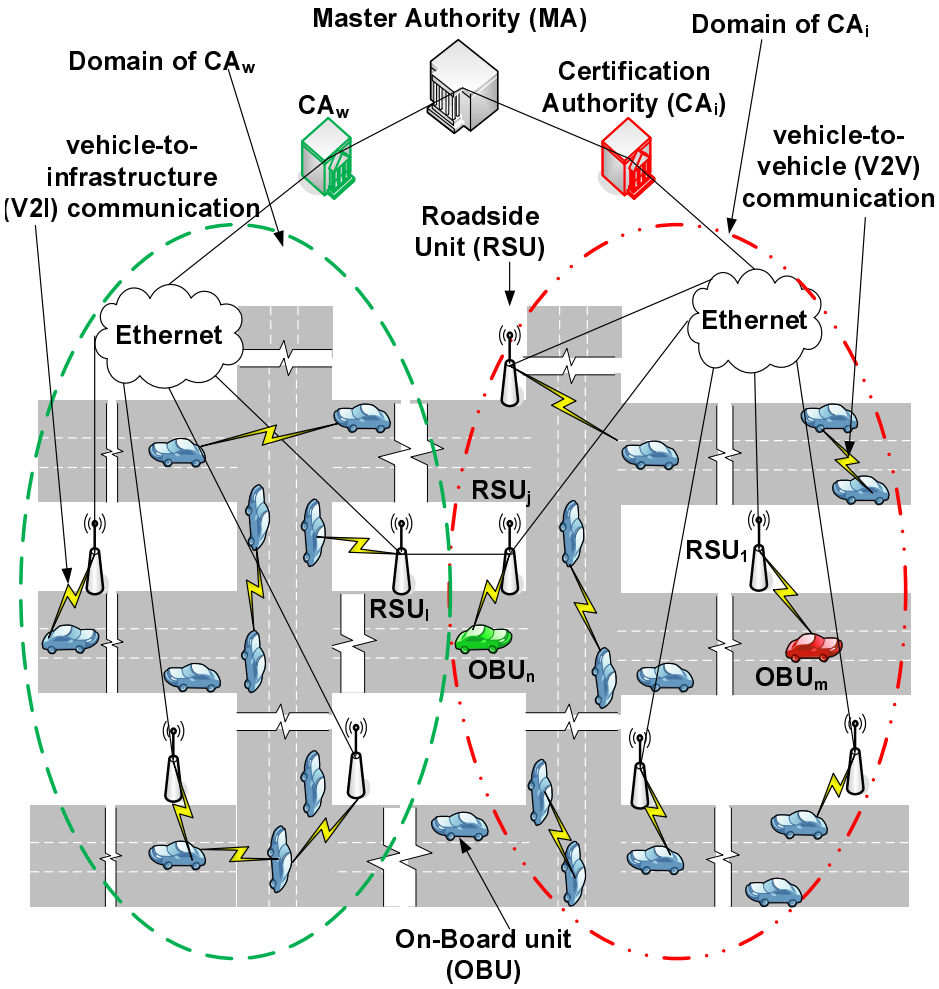


Figure 3.3: The network model

- A Master Authority (MA), which is the highest level in the system and is trustable by all the network entities. The MA has sufficient physical security measures such that it cannot be compromised irrespective of the capabilities of an attacker;

- Certification Authorities (CAs). Each CA is responsible for generating initial certificates for the RSUs and OBUs in its domain. The CAs are connected directly to the

MA. Each CA is physically secure and cannot be compromised;

- Road-Side Units (RSUs), which are fixed units distributed in the network. RSUs in one domain are connected via Ethernet to the CA responsible for that domain. Also, an RSU_j at the border of one domain is connected to the nearest RSU_l in an adjacent domain. These connections are required to check the revocation status of an OBU roaming between two adjacent domains. Moreover, RSUs are responsible for updating the certificates of the OBUs;

- On-Board Units (OBUs), which can communicate either with other OBUs through Vehicle-to-Vehicle (V2V) communications or with the infrastructure RSUs through Vehicle-to-Infrastructure (V2I) communications. Each OBU is equipped with a Global Positioning Service (GPS) receiver which contains the geographical coordinates of the RSUs. It should be noted that a GPS receiver is necessary for the operation of an OBU in VANETs according to the WAVE standard [2];

- According to the WAVE standard, each network entity is equipped with a tamper-resistant Hardware Security Module (HSM) to store its security materials, e.g., secret keys, certificates, etc.

3.4 The Proposed DCS Scheme

In this section, the proposed DCS scheme is presented in detail including the system initialization, certificate issue, certificate update, and certificate revocation.

3.4.1 System Initialization

The initialization stage in the DCS scheme consists of two phases: (1) Phase I which is performed by the MA to generate the security keys necessary for the operation of the DCS scheme, and to upload the necessary security keys in the tamper-resistant HSM of each CA; (2) Phase II which is performed by each CA to upload the required security materials, e.g., keys, certificates, etc., in the tamper-resistant HSM of each OBU and RSU in its domain. It should be noted that both phases of the initialization stage are performed during the registration of CAs with MA in phase I, and RSUs and OBUs with a CA in phase II. In other words, both phases of the initialization stage are performed before triggering any of the VANET services or applications. The details of each phase are as follows.

Algorithm 1 Phase I

Require: ID_{CA_i}

1: Select a random number $s \in \mathbb{Z}_q^*$ as the *master* key, ▷ this is part of each entity secret key
and set $P_\circ = sP$
2: Select random numbers $\alpha, \gamma \in \mathbb{Z}_q^*$, and ▷ these are the *master signing* keys
set $P_\alpha = \alpha P, P_\gamma = \gamma P$
3: Select a random number $\mu \in \mathbb{Z}_q^*$, and
set $P_\mu = \mu P$; ▷ general verification public key
4: Select a hash function $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1^*$
5: Select a hash function $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$
6: **for all** CA_i with identity ID_{CA_i} **do**
7: Set $PK_i = Q_i = H_1(ID_{CA_i}) \in \mathbb{G}_1^*$, ▷ this is CA_i public key
$SK_i = sQ_i$, ▷ this is CA_i secret key
$S_{\alpha i} = \alpha Q_i$, and ▷ this is CA_i certificate-signing key
$S_{\gamma i} = \gamma Q_i$ ▷ this is CA_i certificate-signing key
8: Upload $SK_i, S_{\alpha i}, S_{\gamma i}, P, P_\circ, P_\alpha, P_\gamma, P_\mu, H_1,$ and H_2 in CA_i
9: **end for**

Phase I

The MA executes Algorithm 1 to generate the necessary secret and public keys for the operation of the DCS scheme, and to upload the primary security materials in each CA.

It should be noted that the key s is the *master* secret key, and it will be part of the secret key of each entity. Also, the secret keys α and γ are *master signing* keys, and they will be parts of each signature on the certificates of the RSUs and OBUs, respectively. Moreover, $P_{\circ}, P_{\alpha}, P_{\gamma}$, and P_{μ} are public verification keys, which can be used by any entity in the network to verify any RSU/OBU certificate. In addition, the public key of any CA_i is the hash of its identity ID_{CA_i}.

By the end of Algorithm 1, each CA will have the security materials required to execute phase II.

Phase II

In this phase, each CA_i runs Algorithm 2 and Algorithm 3 to respectively initialize each RSU_j and OBU_m in its domain by uploading them with the necessary security materials for their operation in VANETs as follows.

RSU initialization

Each CA_i executes Algorithm 2, to upload each RSU_j with a certificate $cert_{RSU_{ji}}$, secret OBU-certificate-signing key $S_{\gamma j_i}$ which will be used later by RSU_j to issue

Algorithm 2 Phase II: RSU initialization

Require: $PK_i = Q_i,\ SK_i = sQ_i,\ S_{\alpha i} = \alpha Q_i,$ and $S_{\gamma i} = \gamma Q_i$

1: **for all** RSU_j in the domain of CA_i **do**
2: select random numbers $x_j,\ a_j \in \mathbb{Z}_q^*$, and a pseudo identity PID_j for RSU_j
3: set $SK_{j_i} = x_j SK_i = x_j s Q_i$, ▷ this is RSU_j secret key
$PK_{j_i} = x_j PK_i = x_j Q_i$, ▷ this is RSU_j public key
$S_{\gamma j_i} = x_j S_{\gamma i} = x_j \gamma Q_i$, ▷ this is the secret OBU-certificate-signing key
$U_j = a_j P$, $T_j = H_2(PK_{j_i}||PID_j||U_j||Q_i) \in \mathbb{Z}_q^*$,
$V_j = S_{\alpha i} + a_j T_j P_\mu$, and
$cert_{RSU_{ji}} = (PK_{j_i},\ U_j,\ V_j,\ PID_j,\ Q_i)$ ▷ this is RSU_j certificate
4: Select minimum and maximum value for the validity period (*vperiod*) of any OBU certificate
5: Upload $cert_{RSU_{ji}}$, $S_{\gamma j_i}$, the minimum and maximum value of *vperiod*, $P, P_\circ, P_\alpha, P_\gamma, P_\mu, H_1,$ and H_2 in RSU_j
6: **end for**

certificates for OBUs, the minimum and maximum value of the validity period of OBUs certificates, and publicly known parameters ($P, P_\circ, P_\alpha,\ P_\gamma, P_\mu, H_1,$ and H_2).

Remarks on Algorithm 2

- It should be noted that U_j and V_j are the signature of CA_i on $cert_{RSU_{ji}}$.
- CA_i stores RSU_j real identity, PID_j, $cert_{RSU_{ji}}$, SK_{ji}, and $S_{\gamma ji}$, thus, CA_i can track the operations performed by RSU_j, in case it is compromised, by associating PID_j with its real identity.
- RSU_j or any other entity can verify the certificate $cert_{RSU_{ji}}$ by calculating $T_j = H_2(PK_{j_i}||\ PID_j||U_j||Q_i)$, and accepting if $\hat{e}(P, V_j) = \hat{e}(P_\alpha, Q_i)\hat{e}(T_j U_j, P_\mu)$. This

verification follows since

$$\begin{aligned}\hat{e}(P, V_j) &= \hat{e}(P, S_{\alpha i} + a_j T_j P_\mu) \\ &= \hat{e}(P, \alpha Q_i + a_j T_j P_\mu) \\ &= \hat{e}(P, \alpha Q_i)\hat{e}(P, a_j T_j P_\mu) \\ &= \hat{e}(\alpha P, Q_i)\hat{e}(T_j a_j P, P_\mu) \\ &= \hat{e}(P_\alpha, Q_i)\hat{e}(T_j U_j, P_\mu).\end{aligned} \tag{3.1}$$

- The CA repeatedly runs Algorithm 2 to load each RSU with a set of certificates. Each certificate is shared with a different group of RSUs to enforce the anonymous group signature when generating OBUs certificates.

OBU initialization

Each CA_i executes Algorithm 3, to upload each OBU_m having identity ID_{OBU_m} in its domain with a number (N_{cert}) of short lifetime certificates. The identity ID_{OBU_m} is a unique identity loaded in OBU_m during the manufacturing process.

Remarks on Algorithm 3

- In Algorithm 3, CA_i selects an arbitrary RSU_j in its service area as the certificate issuer, and uses the security materials $\{cert_{RSU_{ji}} = (PK_{j_i},\ U_j,\ V_j,\ PID_j,\ Q_i)$, $SK_{j_i} = x_j s Q_i,\ S_{\gamma j_i} = x_j \gamma Q_i\ \}$ of RSU_j. Note that CA_i is the entity which issued these security materials for RSU_j.

- CA_i stores the real identity (ID_{OBU_m}) and $\{PID_{m,r},\ cert_{m_{ji},r}, SK_{m_{ji},r} | 1 \leq r \leq N_{cert}\}$ of OBU_m, thus, CA_i can efficiently track OBU_m, in case it is compromised, by

Algorithm 3 Phase II: OBU initialization

Require: $\{cert_{RSU_{ji}} = (PK_{j_i},\ U_j,\ V_j,\ PID_j,\ Q_i),\ SK_{j_i} = x_j s Q_i,$ and $S_{\gamma j_i} = x_j \gamma Q_i\ \}$ of RSU_j and ID_{OBU_m} of OBU_m

1: **for all** OBU_m in the domain of CA_i **do**
2: Check the validity of ID_{OBU_m}
3: **if** ID_{OBU_m} is invalid **then**
4: **return** $\perp$
5: **else**
6: **for** r $\leftarrow 1$ **to** N_{cert}, CA_i **do**
7: Select random numbers $y_{m,r},\ b_{m,r} \in \mathbb{Z}_q^*$
8: Set $y_{m,r}SK_{j_i} = y_{m,r}x_j s Q_i$, and ▷ partial secret key
$y_{m,r}PK_{j_i} = y_{m,r}x_j Q_i$ ▷ partial public key
9: **end for**
10: **return** $\{y_{m,r}SK_{j_i},\ y_{m,r}PK_{j_i} | 1 \leq r \leq N_{cert}\}$ to OBU_m
11: **for** r $\leftarrow 1$ **to** N_{cert}, OBU_m **do**
12: Select a random number $z_{m,r} \in \mathbb{Z}_q^*$
13: Set $SK_{m_{ji},r} = z_{m,r}y_{m,r}SK_{j_i} = z_{m,r}y_{m,r}x_j s Q_i$, and ▷ final secret key
$PK_{m_{ji},r} = z_{m,r}y_{m,r}PK_{j_i} = z_{m,r}y_{m,r}x_j Q_i$ ▷ final public key
14: **end for**
15: **return** $\{PK_{m_{ji},r} | 1 \leq r \leq N_{cert}\}$ to CA_i
16: **for** r $\leftarrow 1$ **to** N_{cert}, CA_i **do**
17: Select a validity period $vperiod_{m,r}$, and a pseudo identity $PID_{m,r}$
18: Set $U'_{m,r} = b_{m,r}P$,
$L_{m,r} = H_2(PK_{m_{ji},r} || vperiod_{m,r} || PID_{m,r} || U'_{m,r}) \in \mathbb{Z}_q^*$,
$V'_{m,r} = S_{\gamma j_i} + b_{m,r}L_{m,r}P_\mu$, and
$cert_{OBU_{mji,r}} = (PK_{m_{ji},r}, U'_{m,r}, V'_{m,r}, vperiod_{m,r}, PID_{m,r}, cert_{RSU_{ji}})$
19: **end for**
20: Upload $\{cert_{OBU_{mji,r}} | 1 \leq r \leq N_{cert}\} = \{PK_{m_{ji},r}, U'_{m,r}, V'_{m,r}, vperiod_{m,r},$
$PID_{m,r}, cert_{RSU_{ji}} | 1 \leq r \leq N_{cert}\}$, $P, P_\circ, P_\alpha, P_\gamma, P_\mu, H_1,$ and H_2 in OBU_m
21: CA_i stores ID_{OBU_m} and $\{PID_{m,r},\ cert_{m_{ji},r}, SK_{m_{ji},r} | 1 \leq r \leq N_{cert}\}$
22: **end if**
23: **end for**

associating PID_m to ID_{OBU_m}.

- It should be noted that throughout the rest of the Chapter whenever the subscript r equals 1, it will be omitted for the ease of presentation.

- Any entity in the network can verify a single certificate $cert_{OBU_{mji}}$ by verifying $cert_{RSU_{ji}}$, then, verifying $cert_{OBU_{mji}}$. Alternatively, $cert_{RSU_{ji}}$ and $cert_{OBU_{mji}}$ can be aggregately verified as follows:

  1. Check *vperiod* and proceed only if it is valid;
  2. Calculate $T_j = H_2(PK_{j_i}||PID_j||U_j||Q_i)$ and $L_m = H_2(PK_{m_{ji}}||vperiod||PID_m||U_m^{\backprime})$;
  3. Accept if $\hat{e}(P, V_j + V_m^{\backprime}) = \hat{e}(P_\alpha, Q_i)\hat{e}(P_\gamma, PK_{j_i})\ \hat{e}(T_jU_j + L_mU_m^{\backprime}, P_\mu)$. This verification follows since

$$\begin{aligned}
\hat{e}(P, V_j + V_m^{\backprime}) &= \hat{e}(P, S_{\alpha i} + a_jT_jP_\mu + S_{\gamma j_i} + b_mL_mP_\mu) \\
&= \hat{e}(P, \alpha Q_i)\hat{e}(P, x_j\gamma Q_i)\hat{e}(P, a_jT_jP_\mu + b_mL_mP_\mu) \\
&= \hat{e}(\alpha P, Q_i)\hat{e}(\gamma P, x_jQ_i)\hat{e}(a_jT_jP + L_mb_mP, P_\mu) \\
&= \hat{e}(P_\alpha, Q_i)\hat{e}(P_\gamma, PK_{j_i})\hat{e}(T_jU_j + L_mU_m^{\backprime}, P_\mu).
\end{aligned} \tag{3.2}$$

- Including $cert_{RSU_{ji}}$ in $cert_{OBU_{mji}}$ guarantees that $cert_{OBU_{mji}}$ is generated by a legitimate RSU_j with a valid public key PK_{j_i}. This inclusion also gives the CA the ability to revoke any operation performed by a compromised RSU during the period from the RSU compromising until the detection of the compromised RSU. In other words, consider an attacker compromises an RSU_l having a certificate $cert_{RSU_{li}}$, and the attacker generates some OBUs' certificates from the compromised RSU_l. When the CA detects that RSU_l is compromised, it revokes $cert_{RSU_{li}}$. The revocation of $cert_{RSU_{li}}$ automatically revokes all the OBUs certificates generated by RSU_j, as those

certificates contain the revoked $cert_{RSU_{li}}$.

3.4.2 OBUs Certificates Update

The DCS scheme enables an OBU to update its certificate from an RSU. Thus, the scalability of the DCS scheme stems from the distributed certification service compared to the centralized certification service in the classical PKI where an OBU has to contact a CA to update its certificate. Since the DCS scheme depends on the RSUs to update the certificates of the OBUs, the density of RSUs is crucial to the operation of the DCS scheme. In this section, we discuss the adaptability of the DCS scheme to different densities of RSUs, and how an OBU can update its certificates dynamically even if it is roaming between different domains. In the certificate update process, an RSU generates a number of short lifetime anonymous certificates for an OBU sufficient to secure the communications of the OBU until it meets another RSU. The number of generated certificates by an RSU depends on the RSUs density.

Adapting DCS to Different RSUs Densities

In this section, we discuss how the DCS scheme can adapt to different densities of RSUs. Let T_{RSU} denote the duration an OBU spent between meeting two different RSUs on its way. When the number of RSUs in a given area increases, it is intuitive that T_{RSU} will decrease and vice versa, i.e., T_{RSU} is inversely proportional to the RSUs density. It should be noted that an OBU has to periodically change its certificate during T_{RSU} to avoid being tracked. Since an OBU spends a time of *vperiod*, which is the validity period of the OBU certificate, using the same certificate, the number of certificates N_{cert} required to protect the privacy of that OBU in the duration it spent between meeting two different RSUs can

be calculated as follows

$$N_{cert} = \left\lceil \frac{T_{RSU}}{vperiod} \right\rceil \tag{3.3}$$

An OBU_n moving on the road can calculate its T_{RSU} value based on its direction, speed, and the coordinates of the RSUs initially loaded in its GPS receiver. When OBU_n needs to update its certificates, it sends a request to update its certificate and the value of its T_{RSU} to an RSU_j. Then, using eq. (3.3) and the appropriate value for *vperiod*, RSU_j can calculate the required number of certificates (N_{cert}) that should be generated to the requesting OBU_n to protect its privacy until it meets the next RSU on its way. In this way, the DCS scheme can adapt to different RSUs densities.

OBUs Dynamic Certificates Update

The DCS offers a full interoperability for any OBU to update its certificate in a completely transparent way, even when it roams into a domain different from its home domain. Consider OBU_n, with certificate $cert_{OBU_{nlw}} = (PK_{n_{lw}}, U_n^{\backprime}, V_n^{\backprime}, vperiod,\ PID_n, cert_{RSU_{lw}})$ generated by RSU_l in the domain of CA_w, enters the domain of CA_i, and needs to update its certificate from RSU_j which has a certificate $cert_{RSU_{ji}} = (PK_{j_i},\ U_j,\ V_j, PID_j,\ Q_i)$, as shown in Fig. 3.3 where OBU_n is shown in green. The certificate update algorithm, shown in Fig. 3.4, has two phases: *phase I* for mutual authentication and generating a shared secret key in a non-interactive way, and *phase II* for issuing a bundle of N_{cert} short lifetime anonymous certificates for OBU_n. The *OBU-Certificate-Update* algorithm is as follows.

Phase I

1. When OBU_n receives the periodically broadcast certificate $cert_{RSU_{ji}}$ of RSU_j, it verifies $cert_{RSU_{ji}}$ by calculating $T_j = H_2(PK_{j_i}||PID_j||U_j||Q_i)$ and proceeds only if

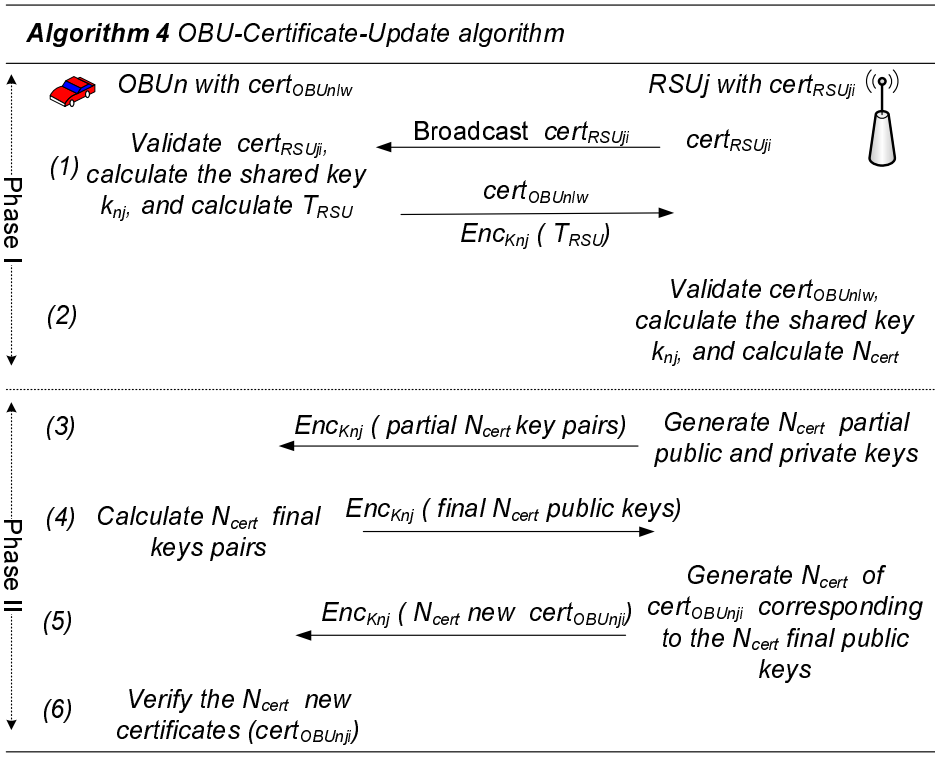


Figure 3.4: OBU Certificate Update

$\hat{e}(P, V_j) = \hat{e}(P_\alpha, Q_i)\hat{e}(T_j U_j, P_\mu)$. If valid, OBU_n calculates the shared secret key (k_{nj}) using its secret key $SK_{n_{lw}}$ and the public key PK_{j_i} of RSU_j included in $cert_{RSU_{ji}}$ as $k_{nj} = \hat{e}(SK_{n_{lw}}, PK_{j_i}) = \hat{e}(z_n y_n x_l s Q_w, x_j Q_i) = \hat{e}(Q_w, Q_i)^{z_n y_n x_l x_j s} = k_{jn}$. Then, OBU_n calculates T_{RSU} based on its speed, destination, and the loaded coordinates of the RSUs. After that, OBU_n encrypts T_{RSU} with k_{nj}, and sends its certificate $cert_{OBU_{nlw}}$ along with the encrypted T_{RSU} to RSU_j;

2. RSU_j verifies $cert_{OBU_{nlw}}$ by calculating $T_l = H_2(PK_{l_w}||PID_l||U_l||Q_w)$ and $L_n = H_2(PK_{n_{lw}} ||vperiod||PID_n||U_n^{\backprime})$, and proceeds only if $\hat{e}(P, V_l + V_n^{\backprime}) = \hat{e}(P_\alpha, Q_l)$ $\hat{e}(P_\gamma, PK_{l_w})\hat{e}(T_l U_l + L_n U_n^{\backprime}, P_\mu)$. If valid, RSU_j calculates the shared secret key as

$k_{jn} = \hat{e}(PK_{n_{lw}}, SK_{j_i}) = \hat{e}(z_n y_n x_l Q_w, x_j s Q_i) = \hat{e}(Q_w, Q_i)^{z_n y_n x_l x_j s} = k_{nj}$ in a non-interactive key agreement way. Then, RSU_j decrypts T_{RSU} using k_{nj}, and calculates N_{cert} using eq. (3.3) based on the bounds of the certificate validity period *vperiod* settled by CA_i.

Phase II

3. As shown in Fig. 3.4, RSU_j selects N_{cert} random numbers $\{y_{n,r}^{\backprime} | 1 \leq r \leq N_{cert}\} \in \mathbb{Z}_q^*$, and calculates N_{cert} partial secret keys as $\{y_{n,r}^{\backprime} x_j s Q_i | 1 \leq r \leq N_{cert}\}$ and the corresponding N_{cert} partial public keys $\{y_{n,r}^{\backprime} x_j Q_i | 1 \leq r \leq N_{cert}\}$. Then, it securely delivers the partial key pairs to OBU_n by encrypting them with the shared secret key k_{nj} established in *Phase I*;

4. OBU_n selects N_{cert} random numbers $\{z_{n,r}^{\backprime} \in \mathbb{Z}_q^* | 1 \leq r \leq N_{cert}\}$, and calculates its final secret keys $\{SK_{n_{ji},r} | 1 \leq r \leq N_{cert}\} = \{z_{n,r}^{\backprime} y_{n,r}^{\backprime} x_j s Q_i | 1 \leq r \leq N_{cert}\}$ and its final public key $\{PK_{n_{ji},r} | 1 \leq r \leq N_{cert}\} = \{z_{n,r}^{\backprime} y_{n,r}^{\backprime} x_j Q_i | 1 \leq r \leq N_{cert}\}$. After that, OBU_n sends its final public keys $\{PK_{n_{ji},r} | 1 \leq r \leq N_{cert}\}$ to RSU_j;

5. For each key in $\{PK_{n_{ji},r} | 1 \leq r \leq N_{cert}\}$, RSU_j chooses a validity period $vperiod_{n,r}$ and a pseudo identity $PID_{n,r}$. After that, RSU_j selects N_{cert} random numbers $\{b_{n,r}^{\backprime} | 1 \leq r \leq N_{cert}\} \in \mathbb{Z}_q^*$, and calculates $\{U_{n,r}^{\backprime} | 1 \leq r \leq N_{cert}\} = \{b_{n,r}^{\backprime} P | 1 \leq r \leq N_{cert}\}$, $\{L_{n,r}^{\backprime} | 1 \leq r \leq N_{cert}\} = \{H_2(PK_{n_{ji},r} || vperiod_{n,r} || PID_{n,r} || U_{n,r}^{\backprime}) | 1 \leq r \leq N_{cert}\} \in \mathbb{Z}_q^*$, and $\{V_{n,r}^{\backprime} | 1 \leq r \leq N_{cert}\} = \{S_{\gamma j_i} + b_{n,r}^{\backprime} L_{n,r}^{\backprime} P_{\mu} | 1 \leq r \leq N_{cert}\}$. Finally, RSU_j issues $\{cert_{OBU_{nji,r}} | 1 \leq r \leq N_{cert}\} = \{(PK_{n_{ji},r}, U_{n,r}^{\backprime}, V_{n,r}^{\backprime}, vperiod_{n,r}, PID_{n,r}, cert_{RSU_{ji}}) | 1 \leq r \leq N_{cert}\}$ and delivers them to OBU_n over a channel secured by the key k_{nj};

6. OBU_n verifies the received certificates $\{cert_{OBU_{nji,r}}|1 \leq r \leq N_{cert}\}$ by calculating $\{L\backprime_{n,r}|1 \leq r \leq N_{cert}\} = \{H_2(PK_{n_{ji}}||vperiod_{n,r}||PID_{n,r}||U\backprime_{n,r})|1 \leq r \leq N_{cert}\}$ and accepts only if

$$\hat{e}(P, \sum_{r=1}^{N_{cert}} V\backprime_{n,r}) = \hat{e}(P_\gamma, \sum_{r=1}^{N_{cert}} PK_{j_i})\hat{e}(\sum_{r=1}^{N_{cert}} L\backprime_{n,r}U\backprime_{n,r}, P_\mu). \tag{3.4}$$

This verification holds since

$$\begin{aligned}
\hat{e}(P, \sum_{r=1}^{N_{cert}} V\backprime_{n,r}) &= \hat{e}(P, V\backprime_{n,1} + V\backprime_{n,2} + \cdots + V\backprime_{n,N_{cert}}) \\
&= \hat{e}(P, S_{\gamma j_i} + b\backprime_{n,1}L\backprime_{n,1}P_\mu + S_{\gamma j_i} + b\backprime_{n,2}L\backprime_{n,2}P_\mu + \cdots \\
&\quad + S_{\gamma j_i} + b\backprime_{n,N_{cert}}L\backprime_{n,N_{cert}}P_\mu) \\
&= \hat{e}(P, \sum_{r=1}^{N_{cert}} S_{\gamma j_i})\hat{e}(P, b\backprime_{n,1}L\backprime_{n,1}P_\mu + b\backprime_{n,2}L\backprime_{n,2}P_\mu + \cdots + b\backprime_{n,N_{cert}}L\backprime_{n,N_{cert}}P_\mu) \\
&= \hat{e}(P, \sum_{r=1}^{N_{cert}} x_j\gamma Q_i)\hat{e}(L\backprime_{n,1}b\backprime_{n,1}P + L\backprime_{n,2}b\backprime_{n,2}P + \cdots + L\backprime_{n,N_{cert}}b\backprime_{n,N_{cert}}P, P_\mu) \\
&= \hat{e}(\gamma P, \sum_{r=1}^{N_{cert}} x_j Q_i)\hat{e}(L\backprime_{n,1}U\backprime_{n,1} + L\backprime_{n,2}U\backprime_{n,2} + \cdots + L\backprime_{n,N_{cert}}U\backprime_{n,N_{cert}}, P_\mu) \\
&= \hat{e}(P_\gamma, \sum_{r=1}^{N_{cert}} PK_{j_i})\hat{e}(\sum_{r=1}^{N_{cert}} L\backprime_{n,r}U\backprime_{n,r}, P_\mu).
\end{aligned} \tag{3.5}$$

By the end of *phase II*, OBU_n gets N_{cert} short lifetime anonymous certificates which are sufficient to protect its privacy until it meets another RSU on its way.

Remarks

- The preceding algorithm enables an OBU_n from one domain (CA_w) to securely up-

date its certificate in another domain (CA_i). Especially, if $i = w$, OBU_n updates its certification in its local domain.

- By increasing the number of the short lifetime certificates an OBU can get from an RSU, the distance an OBU can move without the need to contact another RSU to update its certificates increases. In other words, by changing the number of certificates N_{cert}, the DCS scheme can adapt to different densities of RSUs. Consider a constant $vperiod = 1min$ [4] for all the certificates of an OBU, and the OBU average speed in a domain is 60 Km/h. When an OBU updates its certificates from an RSU for values of N_{cert} equal 5 and 10 certificates, these values are sufficient to protect the privacy of that OBU over distances of 5 km and 10 km, respectively, without the need to contact another RSU.

- When an RSU_j uses one of its certificates ($cert_{RSU_{ji}}$) and signing keys ($S_{\gamma j_i}$) to issue a certificate for an OBU, this is corresponding to using anonymous group signature since $S_{\gamma j_i}$ and $cert_{RSU_{ji}}$ are shared among multiple RSUs. Also, the generated certificate for OBU contains a pseudo identity (PID) which cannot be related to the real identity of the OBU. Since an observer can link an OBU certificate to neither the real identity of the OBU nor the location of the RSU which issued that certificate, the issued certificate $cert_{OBU_{nji}}$ is anonymous.

- The non-interactive key agreement in *Phase I* (Step 1 and Step 2) is very attractive to vehicular networks, since it enables any entity $\mathcal{A}$ to establish a shared secret key with another entity $\mathcal{B}$ by calculating the bilinear pairing of its secret key and the public key of $\mathcal{B}$. The non-interactive key agreement is of significant importance for updating certificates and establishing secure channels in VANETs.

3.4.3 Certificate Revocation

Revocation is required to prevent compromised entities from accessing the network. In the DCS scheme, we adopt the Certificate Revocation List (CRL) method, which is the revocation method employed in the WAVE standard [2]. A CRL is a list containing all the identities and the validity periods of the revoked certificates. It should be noted that the short lifetime certificates of OBUs will be self revoked after their lifetime expires. The certificates of an entity (OBU or RSU) are added to a CRL only if the entity is compromised. When an entity (OBU or RSU) is compromised in one domain, the CA responsible for that domain adds all the certificates of the compromised entity to the current CRL, and broadcasts the new CRL in its domain. Each entity continuously maintains the recently received CRL by removing the certificates with expired validity periods.

According to the distribution of the CRLs in the DCS scheme, each CA distributes the CRL to the RSUs in its domain through its local Ethernet. Then, the RSUs receiving the new CRL broadcasts it to all the OBUs in that domain. Also, the CRL is delivered from the border RSUs in one domain (i) to the border RSUs in the adjacent domain (w) to enable the border RSUs in domain (w) to check the revocation status of the OBUs coming from domain (i). However, the CRL corresponding to domain (i) will be kept in the border RSUs in domain w, and it will not be further broadcast in domain w. For example, a CRL is broadcast by CA_i in its domain (see Fig. 3.3). This CRL is broadcast in domain i until it reaches RSU_j. Then, RSU_j broadcasts this CRL in its coverage area, and it delivers this CRL to RSU_l in domain w. RSU_l stores this CRL to check the revocation status of the OBUs moving from domain i to domain w. In case RSUs do not completely cover the domain of a CA, Laberteaux *et al.* [11] show that V2V communication can be used to efficiently distribute a CRL to all the OBUs. More results about the efficiency of the CRL

distribution using V2V communications can be found in [11].

3.5 Certificate-Based Message Signature and Verification

To satisfy the data authentication and non-repudiation security requirements of VANETs, each entity in the system should be capable of signing and verifying a given message with the corresponding certificate. In this section, we present the basic message signature and verification, followed by the proposed batch verification for message signature and certificate.

3.5.1 OBU/RSU/CA Message Signature and Verification

An OBU_m with $cert_{OBU_{mji}}$ can generate a valid signature $(U_m^{\backprime\backprime}, V_m^{\backprime\backprime})$ for a given message M as follows.

1. Select a random number $c_m \in \mathbb{Z}_q^*$;
2. Calculate $U_m^{\backprime\backprime}$, R_m, and $V_m^{\backprime\backprime}$, where $U_m^{\backprime\backprime} = c_m P$, $R_m = H_2(M||PK_{m_{ji}}||U_m^{\backprime\backprime}||PID_m||t_{stamp}) \in \mathbb{Z}_q^*$, and $V_m^{\backprime\backprime} = SK_{m_{ji}} + c_m R_m P_\mu$;
3. $(U_m^{\backprime\backprime}, V_m^{\backprime\backprime})$ is a valid signature on M.

Any entity in the network can verify the signature $(U_m^{\backprime\backprime}, V_m^{\backprime\backprime})$ on the message M as follows.

1. Verify that the sender of the message is a valid user and check the time stamp t_{stamp};

2. Calculate

$$R_m = H_2(M||PK_{m_{ji}}||U_m^{\prime\prime\prime}||PID_m||t_{stamp}); \tag{3.6}$$

3. Accept if

$$\begin{aligned}\hat{e}(P, V_m^{\prime\prime\prime}) &= \hat{e}(P, SK_{m_{ji}} + c_m R_m P_\mu) \\ &= \hat{e}(P, SK_{m_{ji}})\hat{e}(P, c_m R_m P_\mu) \\ &= \hat{e}(P, z_m y_m x_j s Q_i)\hat{e}(P, c_m R_m P_\mu) \\ &= \hat{e}(sP, z_m y_m x_j Q_i)\hat{e}(R_m c_m P, P_\mu) \\ &= \hat{e}(PK_{m_{ji}}, P_\circ)\hat{e}(R_m U_m^{\prime\prime\prime}, P_\mu).\end{aligned} \tag{3.7}$$

Similarly, any CA or RSU can sign an arbitrary message using the aforementioned procedures.

3.5.2 Batch Verification for Messages Signatures

Consider an OBU $\mathcal{A}$ receives $(U_1^{\prime\prime\prime},\ V_1^{\prime\prime\prime})$, $(U_2^{\prime\prime\prime},\ V_2^{\prime\prime\prime})$, $\cdots$, $(U_K^{\prime\prime\prime},\ V_K^{\prime\prime\prime})$, which are the signatures on the messages $M_1,\ M_2,\ \cdots,\ M_K$, respectively. Then, those signatures can be aggregately verified as follows.

1. Calculate $\overline{V}^{\prime\prime\prime} = \sum_{k=1}^{K} V_k^{\prime\prime\prime}$, and $R_1,\ R_2,\ \cdots,\ R_K$ as in eq. (3.6);

2. Calculate $\overline{U}^{\prime\prime\prime} = \sum_{k=1}^{K} R_k U_k^{\prime\prime\prime}$;

3. Accept if

$$\hat{e}(P, \overline{V}^{\prime\prime\prime}) = \hat{e}(P_\circ, \sum_{k=1}^{K} PK_{OBU,k})\hat{e}(\overline{U}^{\prime\prime\prime}, P_\mu) \tag{3.8}$$

where $PK_{OBU,k}$ is the public key in certificate k.

PROOF: Firstly, we consider an OBU $\mathcal{A}$ receives two messages from OBU_m and OBU_n, where OBU_m generates a signature $(U_m^{\backprime\backprime\backprime}, V_m^{\backprime\backprime\backprime})$ on the message M_1, where $U_m^{\backprime\backprime\backprime} = c_m P$ and $V_m^{\backprime\backprime\backprime} = SK_m + c_m R_m P_\mu = z_m y_m x_j s Q_i + c_m R_m P_\mu$. In addition, OBU_n generates a signature $(U_n^{\backprime\backprime\backprime}, V_n^{\backprime\backprime\backprime})$ on the message M_2, where $U_n^{\backprime\backprime\backprime} = c_n P$ and $V_n^{\backprime\backprime\backprime} = SK_m + c_n R_n P_\mu = z_n y_n x_j s Q_i + c_n R_n P_\mu$. OBU $\mathcal{A}$ calculates $\overline{V}^{\backprime\backprime\backprime} = V_m^{\backprime\backprime\backprime} + V_n^{\backprime\backprime\backprime} = z_m y_m x_j s Q_i + c_m R_m P_\mu + z_n y_n x_j s Q_i + c_n R_n P_\mu$. The received signatures can be aggregately verified by calculating R_m, R_n and checking that

$$\begin{aligned}
\hat{e}(P, \overline{V}^{\backprime\backprime\backprime}) &= \hat{e}(P, z_m y_m x_j s Q_i + c_m R_m P_\mu + z_n y_n x_j s Q_i + c_n R_n P_\mu) \\
&= \hat{e}(P, z_m y_m x_j s Q_i + z_n y_n x_j s Q_i) \hat{e}(P, c_m R_m P_\mu + c_n R_n P_\mu) \\
&= \hat{e}(sP, z_m y_m x_j Q_i + z_n y_n x_j Q_i) \hat{e}(R_m c_m + P R_n c_n P, P_\mu) \\
&= \hat{e}(P_\circ, PK_{m_{ji}} + PK_{n_{ji}}) \hat{e}(R_m U_m^{\backprime\backprime\backprime} + R_n U_n^{\backprime\backprime\backprime}, P_\mu) \\
&= \hat{e}(P_\circ, \sum_{k=1}^{2} PK_{OBU,k}) \hat{e}(\overline{U}^{\backprime\backprime\backprime}, P_\mu)
\end{aligned} \tag{3.9}$$

As for the multiple-message, they can be verified in a similar way.

3.5.3 Batch Verification for Certificates

Consider an OBU_m with certificate $cert_{OBU_{mji}} = (PK_{m_{ji}}, U_m^{\backprime}, V_m^{\backprime}, vperiod_m, PID_m, cert_{RSU_{ji}})$, and OBU_n with certificate $cert_{OBU_{nlw}} = (PK_{n_{lw}}, U_n^{\backprime}, V_n^{\backprime}, vperiod_n, PID_n, cert_{RSU_{lw}})$, where $cert_{RSU_{ji}} = (PK_{j_i},\ U_j,\ V_j,\ PID_j,\ Q_i)$ and $cert_{RSU_{l_w}} = (PK_{l_w},\ U_l,\ V_l,\ PID_l,\ Q_w)$. An independent third party can aggregately verify the OBUs certificates and the RSUs certificates included in them as follows.

1. Check *vperiod* of each certificate and proceed only if valid;
2. Calculate $T_j = H_2(PK_{j_i}||PID_j||U_j||Q_i)$ and $T_l = H_2(PK_{l_i}||PID_l||U_l||Q_i)$;
3. Calculate $L_m = H_2(PK_{m_{ji}}||vperiod_m||PID_m||U_m^{\backprime})$ and $L_n = H_2(PK_{n_{lw}}||vperiod_n||PID_n ||U_n^{\backprime})$;
4. Calculate $\overline{V} = V_j + V_l$, $\overline{V}^{\backprime} = V_m^{\backprime} + V_n^{\backprime}$, $\overline{U} = T_jU_j + T_lU_l$, and $\overline{U}^{\backprime} = L_mU^{\backprime} + L_nU^{\backprime}$;
5. Accept if $\hat{e}(P, \overline{V} + \overline{V}^{\backprime}) = (P_\alpha, Q_i + Q_w)\hat{e}(P_\gamma, PK_{ji} + PK_{lw})\hat{e}(\overline{U} + \overline{U}^{\backprime}, P_\mu)$. This verification holds since

$$\begin{aligned}
\hat{e}(P, \overline{V} + \overline{V}^{\backprime}) &= \hat{e}(P, S_{\alpha i} + a_jT_jP_\mu + S_{\alpha w} + a_lT_lP_\mu + S_{\gamma ji} + b_mL_mP_\mu + S_{\gamma lw} + b_nL_nP_\mu) \\
&= \hat{e}(P, \alpha Q_i + a_jT_jP_\mu + \alpha Q_w + a_lT_lP_\mu + x_j\gamma Q_i + b_mL_mP_\mu + x_l\gamma Q_w + b_nL_nP_\mu) \\
&= \hat{e}(P, \alpha Q_i + \alpha Q_w)(P, x_j\gamma Q_i + x_l\gamma Q_w)\hat{e}(P, a_jT_jP_\mu + a_lT_lP_\mu + b_mL_mP_\mu + b_nL_nP_\mu) \\
&= \hat{e}(\alpha P, Q_i + Q_w)\hat{e}(\gamma P, x_jQ_i + x_lQ_w)\hat{e}(T_jU_j + T_lU_l + L_mU^{\backprime} + L_nU^{\backprime}, P_\mu) \\
&= \hat{e}(P_\alpha, Q_i + Q_w)\hat{e}(P_\gamma, PK_{ji} + PK_{lw})\hat{e}(\overline{U} + \overline{U}^{\backprime}, P_\mu)
\end{aligned} \tag{3.10}$$

For K OBUs, their certificates can be aggregately verified as follows:

$$\hat{e}(P, \overline{V} + \overline{V}^{\backprime}) = \hat{e}(P_\alpha, \sum_{k=1}^{K} Q_k)\hat{e}(P_\gamma, \sum_{k=1}^{K} PK_{RSU,k})\hat{e}(\overline{U} + \overline{U}^{\backprime}, P_\mu) \tag{3.11}$$

where $\overline{V} = \sum_{k=1}^{K} V_k$, $\overline{V}^{\backprime} = \sum_{k=1}^{K} V_k^{\backprime}$, $PK_{RSU,k} = PK_{ji} + PK_{lw} + \cdots$, $\overline{U} = \sum_{k=1}^{K} T_kU_k$, and $\overline{U}^{\backprime} = \sum_{k=1}^{K} L_kU_k^{\backprime}$

3.5.4 Batch Verification for Messages Signatures and Certificates

Consider K OBUs with K certificates generating different K signatures on K different messages. An independent third party can aggregately verify the K signatures and certificates by combining eq. (3.8) and eq. (3.11) as follows.

$$\hat{e}(P, \overline{V} + \overline{V}^{\backprime} + \overline{V}^{\backprime\backprime}) = \hat{e}(P_{\circ}, \sum_{k=1}^{K} PK_{OBU,k})\hat{e}(P_{\alpha}, \sum_{k=1}^{K} Q_k)\hat{e}(P_{\gamma}, \sum_{k=1}^{K} PK_{RSU,k})\hat{e}(\overline{U} + \overline{U}^{\backprime} + \overline{U}^{\backprime\backprime}, P_{\mu}) \tag{3.12}$$

The proof of eq. (3.12) follows directly from eq. (3.9) and eq. (3.10). Eq. (3.12) shows that the DCS scheme overcomes the need to separately verify signatures and certificates of the senders, which is common to most of the existing batch verification schemes. The DCS scheme amplifies the capabilities of any entity in the network to simultaneously verify a relatively large number of signatures and certificates compared to the conventional verification method which verifies signatures and certificates one by one, thus, decreasing the verification overhead.

It should be noted that eq. (3.12) can be used by any OBU or RSU to verify the signatures and the certificates included in the K different messages sent by K OBUs. Consequently, eq. (3.12) represents how authentication can be achieved in V2V and V2I communications.

When there are invalid signatures in the received messages, the data cross checking technique employed in the WAVE standard can alleviate the effect of the invalid signatures. In specific, each OBU_n compares the data included in the received message from an OBU_m with those received from other OBUs. If there is a mismatch, OBU_n rejects the message. It should be noted that the data cross checking technique is useful only when the data

contents of the message are malicious. However, if either the data contents of the message are correct and the signature is invalid or the message and signature are correct and the certificate is invalid, this technique is not useful. In such case, a search approach based on the binary authentication tree [44] can be employed to avoid individually verifying every signature.

3.6 Security Analysis

In this section, we evaluate the proposed DCS scheme according to the security objectives presented in section 3.3.1.

1. Authentication: It can be seen that finding the secret keys $s,\ \alpha,\ \gamma,\ \mu$ from the corresponding public keys $P_\circ,\ P_\alpha,\ P_\gamma,\ P_\mu$ are instances of the ECDLP problem. For example, to find s, we have the following ECDLP problem: given P and $P_\circ = sP$, find s. In DCS, the authentication of RSUs and OBUs is achieved using digital certificates. For example, the signature of any CA_i on the certificate of any RSU_j is $(U_j,\ V_j)$, where $U_j = a_jP$, $T_j = H_2(PK_{j_i}||PID_j||U_j||Q_i) \in \mathbb{Z}_q^*$, and $V_j = S_{\alpha i} + a_jT_jP_\mu$. It can be seen that to forge the certificate of any RSU_j, an attacker should know either $S_{\alpha i} = \alpha Q_i$ or $a_jT_jP_\mu$. Since Q_i is publicly known, finding $S_{\alpha i}$ reduces to finding α which is ECDLP problem as indicated above. Also, since T_j can be easily obtained from the certificate of RSU_j, finding $a_jT_jP_\mu$ reduces to finding a_jP_μ, which can be formulated as a CDH problem, i.e., given $U_j = a_jP$ and $P_\mu = \mu P$, find $a_jP_\mu = a_j\mu P$. The hardness of the CDH problem is closely related to solving the Discrete Logarithm (DL) problem [41]. Similar analogy applies to the OBUs certificates. Since ECDLP and CDH are hard computational problems [40][41], i.e., they cannot be

solved in a sub-exponential time, the certificates of RSUs and OBUs are unforgeable. Since in each communication, an authentication of the sender is performed first, an illegitimate entity cannot communicate with the authentic network users. Also, data authentication is achieved by employing digital signatures, where any message transmitted by any CA, RSU, or OBU has to be signed first. Consequently, any message alteration during the transmission will be detected by the recipient. In clogging attacks, an attacker tries to impersonate a legitimate user, and overwhelms legitimate entities in the network by involving them in a large volume of key exchange or by sending bogus messages [45]. In the DCS scheme, each OBU/RSU authenticates the received messages before being involved in any key exchange or responding to the received message. According to [45], since authentication is done first before taking any action, the clogging attacks is hard to launch in the proposed DCS scheme.

2. Non-repudiation: Non-repudiation is achieved by requiring all the messages exchanged in the network to be digitally signed by its issuer. For example, the signature of any OBU_m on an arbitrary message M is (U'', V''), where $U'' = cP$, and $V'' = SK_{m_{ji}} + cRP_{\mu}$. Similar to the above discussion of the security of RSUs certificates, to forge the signature of OBU_m on M, the attacker has to find either $SK_{m_{ji}}$, which is ECDLP problem, or cRP_{μ}, which is CDH problem. Consequently, the signature of any entity cannot be forged. In addition, since non-repudiation is guaranteed, the liability requirement is also achieved since users cannot deny the transmission or the content of their messages.

3. Privacy: In DCS, privacy is preserved by the following techniques:

   - *Anonymous authentication:* Anonymous authentication is employed in DCS in the sense that each OBU has a certificate containing only a pseudo identity,

which cannot lead in any way to the real identity of the OBU. Furthermore, by deploying anonymous authentication, the DCS scheme can efficiently prevent an adversary from tracking the real identity of the users.

- *Frequent certificate update:* OBUs certificates have a short-lifetime. As a result, each OBU has to periodically change its certificate, which decreases the probability of being tracked by an external observer.
- *Anonymous certificate issuer:* Since each RSU certificate is shared among multiple RSUs, the RSU certificate included in each OBU certificate cannot lead to the location where the OBU issued its certificate.
- *Avoiding key escrow:* When an OBU_m updates its certificate from an RSU_j, RSU_j sends a partial secret key $y_m x_j s Q_i$ to OBU_m. After that, OBU_m calculates its final secret key as $SK_{m_{ji}} = z_m y_m x_j s Q_i$. It can be seen that finding $SK_{m_{ji}}$ from the partial secret key is ECDLP problem. Since the secret key of any OBU cannot be forged, the DCS is free from the key screw which is common to any PKI. As a result, the messages signed by the secret key of any OBU can only be verified by the public key of that OBU, and this signature cannot be generated by any other entity in the network, hence, achieving high privacy level.

Although the DCS offers a collation of privacy preserving mechanisms, an observer can still launch a tracking attack on an OBU. However, this tracking attack requires an observer to launch a large number of receivers along the path of the targeted OBU, and the targeted OBU has to move with the same velocity and in the same lane between any pair of adjacent receivers launched by the observer [4]. To protect the OBUs against this tracking of attack, the DCS can be efficiently integrated with Random Encryption Periods (REPs) proposed in [46]. In REPs, using group

communications, an OBU surrounds itself with an encrypted communication zone to violate the conditions of being tracked by an observer.

4. Transparent roaming: Since any OBU can update its certificate from any RSU in the network, the DCS scheme overcomes the need to re-register the OBU entering a new domain with the new CA. Consequently, the transparent roaming is guaranteed in the DCS scheme.

5. Access control: Any illegal network access by a compromised RSU can be efficiently thwarted, since a CA can broadcast a revocation message including the certificates of that RSU. Once receiving that revocation message, all the OBUs can de-associate with that compromised RSU. Also, all the OBUs certificates issued by that RSU are revoked, as the revoked RSU certificates are contained in those certificates. In addition, a CA can revoke any misbehaving OBU by broadcasting a CRL containing the certificate of the misbehaving OBU. Consequently, all the network RSUs and OBUs terminate the communications with that OBU.

3.7 Performance Evaluation

In this section, we evaluate the performance of the DCS scheme from different aspects.

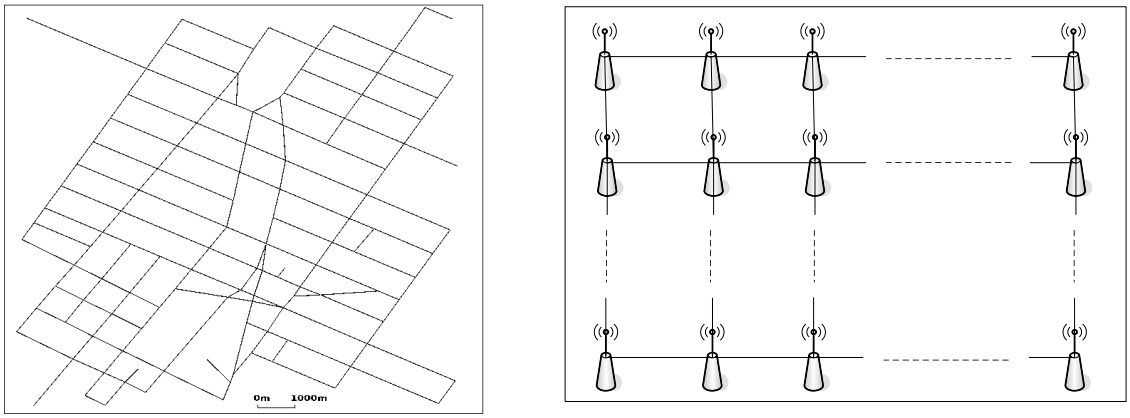
3.7.1 OBU Certificate Update Delay

In this subsection, we compare between the OBU certificate update delay in the DCS scheme, the ECPP protocol, and the classical PKI where an OBU has to contact a CA to update its certificates.

Table 3.2: DCS certificate update cryptography delay

certificate update step	operation	entity involved	cryptography delay
step(2)	OBU_n certificate verification	RSU_j	$4T_{pair} + 2T_{mul}$
	calculation of the shared key k_{jn}	RSU_j	T_{pair}
step(3)	calculation of N_{cert} partial public keys	RSU_j	$N_{cert}T_{mul}$
	calculation of N_{cert} partial secret keys	RSU_j	$N_{cert}T_{mul}$
step(4)	generation of N_{cert} final public keys	OBU_n	$N_{cert}T_{mul}$
step(5)	calculation of $\{U'_{n,r} \mid 1 \leq r \leq N_{cert}\}$	RSU_j	$N_{cert}T_{mul}$
	calculation of $\{L'_{n,r} \mid 1 \leq r \leq N_{cert}\}$	RSU_j	$N_{cert}T_{mul}$

Let $T_{cert-DCS}, T_{cert-ECPP}$, and $T_{cert-CA}$ denote the time from the moment an OBU requests N_{cert} new certificates from an RSU to the moment it receives the required certificates in the DCS scheme, ECPP protocol, and the classical PKI, respectively. We consider the cryptography delay only due to the pairing and point multiplication operations on an elliptic curve as they are the most time consuming operations in the schemes under consideration. Let T_{pair} and T_{mul} denote the time required to perform a pairing operation and a point multiplication, respectively. In [26], T_{pair} and T_{mul} are found for an MNT curve with embedding degree $k = 6$ to be equal to 4.5 $msec$, and 0.6 $msec$, respectively. Let $T_{crypt-DCS}$ and $T_{crypt-ECPP}$ denote the total incurred cryptography delay from the moment an OBU requests N_{cert} new certificates from an RSU to the moment it receives the required certificates in the DCS scheme, and ECPP protocol, respectively. It should be noted that the cryptography delay (T_{crypt}) is part of the certificate update delay (T_{cert}) in any of the schemes under consideration. Table 3.2 gives the cryptography delay incurred in each step of the DCS certificate update algorithm, shown in Fig. 3.4, from the moment an OBU requests N_{cert} new certificates from an RSU, i.e., step (2), until it receives the required



(a) City street map

(b) RSUs connection pattern

Figure 3.5: Simulation scenario

certificates, i.e., by the end of step (5). According to Table 3.2, we have

$$T_{crypt-DCS} = 5T_{pair} + (2 + 5N_{cert})T_{mul} \tag{3.13}$$

In the ECPP protocol [24], an RSU generates only one certificate for an OBU requesting certificate update. However, the ECPP protocol can be easily extended to enable an RSU to generate a bundle of N_{cert} certificates for the requesting OBU similar to the DCS scheme. In the case where ECPP protocol generates N_{cert} for the requesting protocol, we have

$$T_{crypt-ECPP} = (3 + 5N_{cert})T_{pair} + (4 + 9N_{cert})T_{mul} \tag{3.14}$$

We have conducted two ns-2 [47] simulations to respectively compare certificate update delay of the DCS scheme with that of the ECPP protocol and the classical PKI for the city street scenario shown in Fig. 3.5(a). The adopted simulation parameters are given in Table 3.3. The mobility traces adopted in this simulation are generated using TraNS [48].

Table 3.3: NS-2 simulation parameters for DCS

simulation area	13.4 $Km \times 12.3$ Km
simulation time	100 sec
max. vehicle speed	60 Km/h
OBU transmission range	300 m
MAC protocol	802.11a
OBU information dissemination interval	300 $msec$
wired channel capacity	100 Mbps
wireless channel capacity	6 Mbps
number of RSUs	576
distribution of RSUs	uniform

We use the IEEE 802.11a standard, which is the basis of DSRC, to simulate the Medium Access Control (MAC) protocol for VANETs [48][49]. VANETs have two types of links: wireless links connecting OBUs to each other and to the RSUs and wired links connecting the RSUs in one domain and the CA responsible for that domain as shown in Fig. 3.3 (we consider only the domain of CA_i in Fig. 3.3). According to the DSRC specifications, each wireless data channel in VANET has a bandwidth of 10 MHz corresponding to channel data rate in the range of 3 $Mbps - 27$ $Mbps$ [50]. We select a data rate of 6 $Mbps$ for the wireless channels in VANET. We consider the links of the Ethernet connecting the RSUs and CA_i to have data rate of 100 $Mbps$. The RSUs connection pattern employed in our simulation is shown in Fig. 3.5(b). The adopted RSU connection considers a well deployed VANET, where the RSUs are uniformly distributed with the distance between any pair of adjacent RSUs is 500 m. CA_i is located at the top left corner of the city scenario shown in Fig. 3.5(a). To simulate real-life VANET scenarios, we conduct the certificate update scenarios imposed on VANET safety-related applications, where each OBU has to

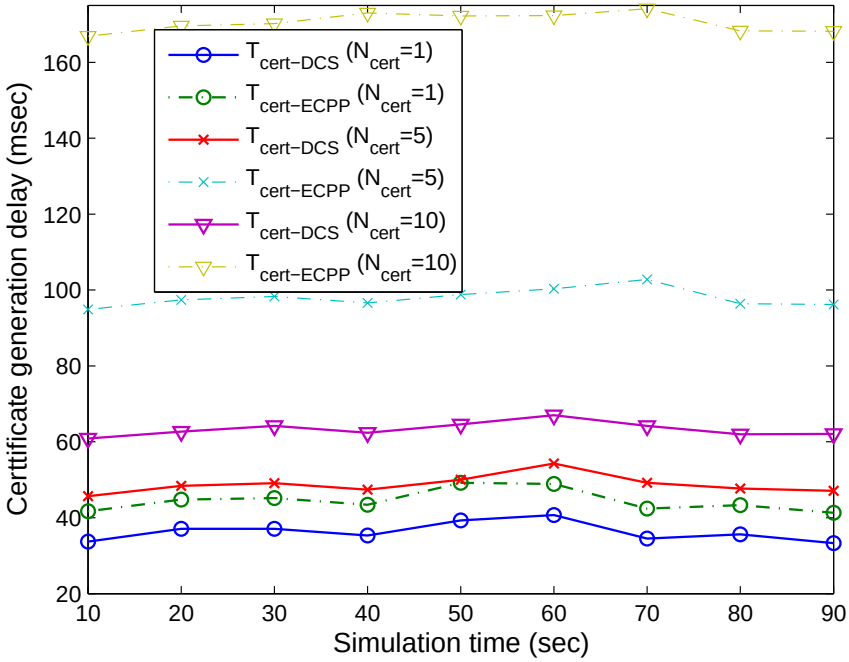


Figure 3.6: Certificate update delay for DCS and ECPP

disseminate information about the road condition every 300 *msec* according to DSRC.

The first simulation is conducted to compare the certificate update delay in the DCS scheme with that in the ECPP protocol. Fig. 3.6 shows the certificate update delay in *msec* for the DCS scheme and the ECPP protocol vs. the simulation time in *sec*. In the conducted simulation, we consider N_{cert} to be constant for all the OBUs, where we consider values of N_{cert} equal 1, 5, and 10 certificates. In addition, an OBU sends a certificate update request every 10 *sec* during the simulation, and the corresponding certificate update delay is measured. The variations in $T_{cert-DCS}$ and $T_{cert-ECPP}$ are due to the variations of the distance separating the OBU requesting the certificate update and the RSU issuing the certificate. Table 3.4 shows the average values of the certificate update delay shown in Fig.

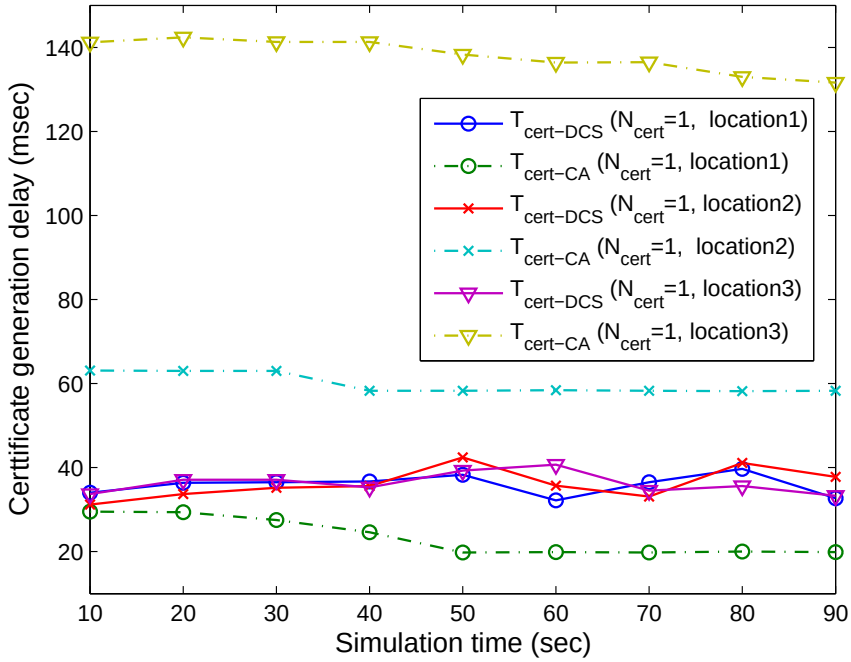


Figure 3.7: Certificate update delay for DCS and classical PKI

3.6. It can be seen from Table 3.4 that the DCS scheme outperforms the ECPP protocol and the percentage of the delay-saving obtained by DCS compared to ECPP increases with N_{cert}. It should be noted that the average values for $T_{cert-DCS}$ and $T_{cert-ECPP}$ in Table 3.4 are independent on the density of the RSUs as only one RSU is involved in each certificate update process. Therefore, the RSUs density has no effect on the certificate update delay.

The second simulation is conducted to compare the certificate update delay of the DCS scheme with that of the classical PKI [2] under a well-deployed VANET. The classical PKI certificate update requires each OBU requesting certificate update to contact the CA through the RSUs as the CA is the only entity responsible for generating the certificates. ECDSA [51] is the classical PKI digital signature method chosen by the WAVE standard,

Table 3.4: Average certificate update delay

N_{cert}	avg. $T_{cert-DCS}$ (msec)	avg. $T_{cert-ECPP}$ (msec)	delay-saving
1	36.2	44.4	18.5%
5	48.8	98	50.2%
10	63.3	170.5	62.9%

where a certificate and signature verification takes $4T_{mul}$, and a signature generation takes T_{mul}.

We consider two certificate update scenarios shown in Fig. 3.3 as follows. The first scenario is the classical PKI certificate update, where OBU_m (shown in red) needs to update its certificates. Hence, it should send a certificate update request to CA_i via the nearest RSU, which in this case is RSU_1. After the request reaches RSU_1, it will be forwarded through the RSUs' Ethernet to CA_i, where the request message experiences a delay of $4T_{mul}$ at each intermediate RSU, as each RSU has to verify the certificate and the signature of the sender before forwarding the request, otherwise, a denial of service attack can be easily launched by sending faked requests, which can overwhelm CA_i. When the certificate update request reaches CA_i, it has to verify the request which takes $4T_{mul}$, and generate new N_{cert} certificates for OBU_m which takes $N_{cert}T_{mul}$. Then, CA_i forwards the new certificates to RSU_l which in turn forwards them to OBU_m. In the second scenario, OBU_m updates its certificates directly from RSU_1 as proposed by the DCS scheme.

Fig. 3.7 shows the classical PKI certificate update delay $T_{cert-CA}$ and the DCS certificate update delay $T_{cert-DCS}$ in $msec$ vs. the simulation time. We conducted simulation for the two certificate update scenarios triggered by OBU_m for N_{cert} equal 1 at three different locations: location1, location2, and location3 corresponding to initial distances of $2.7Km$, $4.7Km$, and $10.3Km$, respectively, from CA_i at the beginning of the simulation. The certificate update process is triggered every 10 sec during the simulation and the

corresponding certificate update delay is measured. The variations in $T_{cert-CA}$ are due to the number of the intermediate RSUs existing in the connection between CA_i and OBU_m. It can be seen that $T_{cert-DCS}$ is almost the same for the three locations, and is confined within the range $31msec - 43msec$. This is due to the fact that the DCS scheme is independent on CA_i. On the other hand, it can be seen that $T_{cert-CA}$ increases with the distance from CA_i. Consequently, the delay-saving of the proposed DCS scheme compared to the classical PKI certificate update increases with distance from the CA. For example, the average certificate update delay is 59.87 $msec$ for location2, while that for the DCS scheme is 36.2 $msec$. Consequently, the DCS scheme decreases the certificate update delay by 39.54% compared to the classical PKI in that case. From the aforesaid discussion, it can be seen that even under a well-deployed VANET the DCS scheme outperforms the classical PKI in terms of certificate update delay, which directly translates into a better certification service. In addition, since in the classical PKI, all certificates updates are handled by the CA, it is expected that the certificate update delay from the CA increases in real-life large scale VANETs.

3.7.2 Successful Certification Ratio

When an OBU_m requests N_{cert} certificates from an RSU_l, RSU_l should process the request, generate the required certificates, and deliver them to OBU_m before OBU_m moves out of the communication range of RSU_l, otherwise, the certificate update process fails. Therefore, if the number of certificate update requests is large, the RSU will not be able to process all the requests and some requests may be dropped. To calculate the maximum number of certificates that an RSU can generate within its coverage range, we adopt the

following formula [24]

$$NC_{max} = \frac{R}{\overline{S} \cdot T_{cert}} \tag{3.15}$$

where NC_{max} is the maximum number of certificates an RSU can generate within its coverage range R, $\overline{S}$ is the average speed of the OBUs within R, and T_{cert} is the average certificate update delay of the scheme under consideration.

Successful Certification Ratio (SCR) is the metric usually used to evaluate the efficiency of authentication algorithms [52]. SCR is defined as the ratio of the number of successful certificate generations (NC_s) to the number of total certificate requests (NC_t). Hence, we have

$$SCR = \begin{cases} 1 & \text{if } NC_t \leq NC_{max} \\ \frac{NC_s}{NC_t} & \text{if } NC_t > NC_{max} \end{cases} \tag{3.16}$$

We consider an RSU with $R = 600m$ (corresponding to omnidirectional communication range with radius 300 m according to DSRC), and the average speed of OBUs is $\overline{S} = 60\ Km/h$. Fig. 3.8 shows the successful certification ratio for the DCS scheme and the ECPP protocol [24] for values of N_{cert} equal 1, 5, and 10 certificates vs. the total number of certificate requests, where we used the values of T_{cert} in Table 3.4. It should be noted that in the cases where $N_{cert} > 1$, each request in Fig. 3.8 is corresponding to generating N_{cert} certificates. It can be seen that DCS gives a higher SCR than the ECPP protocol. Also, the SCR for DCS with $N_{cert} = 10$ is even higher than that of the ECPP with $N_{cert} = 5$. Since DCS can handle a larger number of certificates requests than ECPP for the same duration, the DCS is more suitable for the requirement of vehicular networks.

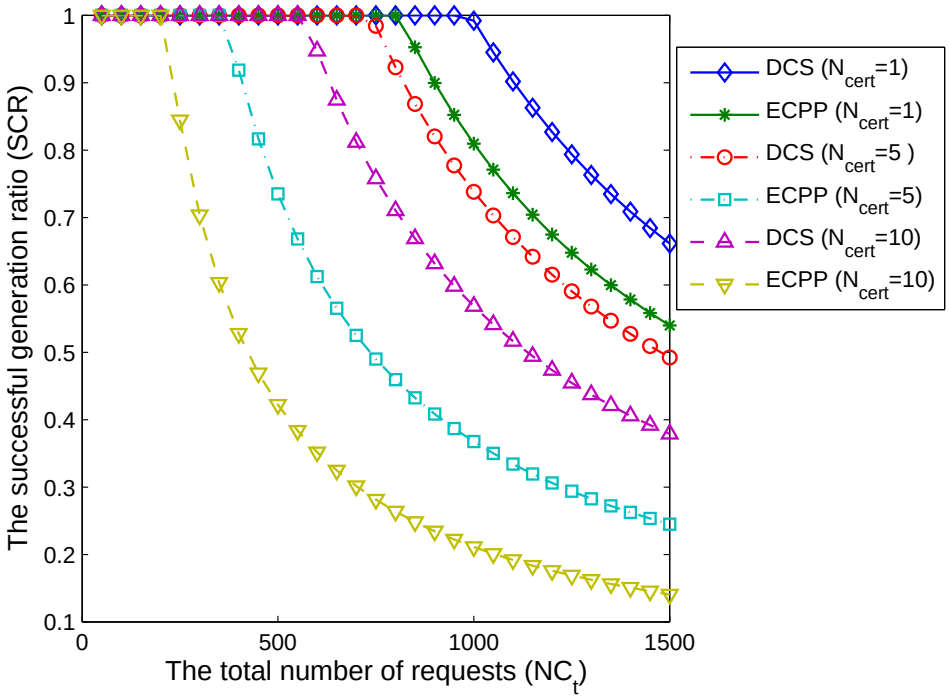


Figure 3.8: Successful certification ratio

Table 3.5: Example of the required $density_{RSU}$ in DCS for $\overline{vperiod} = 1min$ and $\overline{S} = 60Km/h$

state	New York	Hawaii
rural roads length (km)	106014	3285
urban roads length (km)	77033	3701
$\overline{N}_{cert}$ (rural)	20	20
$\overline{N}_{cert}$ (urban)	10	10
$density_{RSU}$ (rural)	0.05	0.05
$density_{RSU}$ (urban)	0.1	0.1
number of required RSUs (rural)	5301	165
number of required RSUs (urban)	7074	371
total number of RSUs	13005	536

3.7.3 The Required RSUs Density in DCS

In this section, we give a rough estimate of the required RSUs density in the DCS scheme. It is more meaningful to express the RSUs density ($density_{RSU}$) as the number of RSUs per road unit length (Km) instead of per unit area (Km^2) as RSUs are implemented only on the roads, and a road width is generally much smaller than its length. The average distance D_{RSU} the OBUs can move without the need to contact an RSU is

$$D_{RSU} = \frac{1}{60}\overline{N}_{cert}\ \overline{vperiod}\ \overline{S}\ \ (Km), \tag{3.17}$$

where $\overline{N}_{cert}$ is the average number of the generated certificates per OBU from the RSUs, $\overline{vperiod}$ is the average validity period of the OBUs certificates in min, and $\overline{S}$ is the average speed of the OBUs in Km/h. It should be noted that the parameters in eq. (3.17) are corresponding to only one domain. Since D_{RSU} can be interpreted as the road distance between two adjacent RSUs. Consequently, the required RSU density ($density_{RSU}$) for the DCS scheme can be calculated as

$$density_{RSU} = \frac{1}{D_{RSU}} = \frac{60}{\overline{N}_{cert}\ \overline{vperiod}\ \overline{S}}\ \ (/Km) \tag{3.18}$$

Eq. (3.18) can be used in the design phase of the DCS scheme to calculate the number of RSUs needed for the operation of the DCS scheme.

Table 3.5 gives an example of the required densities and numbers of RSUs for New York and Hawaii states for $\overline{vperiod} = 1min$ and $\overline{S} = 60Km/h$. New York has an area of $141299Km^2$ while that for Hawaii is $28311Km^2$ [53]. The total length of the urban and rural roads is obtained from [54]. Since the density of the OBUs in an urban road is higher than that in a rural road, it will not be cost-effective to implement RSUs in rural

Table 3.6: RSU_j certificate size in DCS

parameter	PK_{j_i}	U_j	V_j	PID_j	Q_i	$cert_{RSU_{ji}}$
size in bytes	21	21	21	8	21	92

Table 3.7: OBU_m certificate size in DCS

parameter	$PK_{m_{ji}}$	$U_m^{\backprime}$	$V_m^{\backprime}$	$vperiod$	PID_m	$cert_{RSU_{ji}}$	$cert_{OBU_{mji}}$
size in bytes	21	21	21	4	8	92	167

roads with a density equal to that in urban roads. Therefore, we select $\overline{N}_{cert}$ for rural and urban roads to be 20 and 10, respectively. The total number of the required RSUs can be decreased by increasing the validity period ($\overline{vperiod}$) of the certificates of the OBUs or increasing N_{cert}. However, increasing $\overline{vperiod}$ increases the probability of being tracked, i.e., lowering the privacy protection level. Also, increasing the number of certificates (N_{cert}) generated from RSUs decreases the SCR as shown in Fig. 3.8. A compromise between the privacy protection level and the SCR of RSUs should be made according to the required RSUs density. It should be noted that each CA can change the minimum and maximum bound to the value of the certificate validity period according to the required level of privacy protection, and broadcast these bounds to the RSUs in its domain through its local Ethernet.

3.7.4 Communication Overhead

We consider the Tate pairing implementation on an MNT curve with embedding degree 6, where $\mathbb{G}_1$ is represented by 161 bits. Accordingly, each point on this MNT curve is represented by 21 bytes. Table 3.6 and Table 3.7 give each parameter and the corresponding size in bytes for an RSU and OBU certificate, respectively. The last column in each table gives the total size of the certificate under consideration. It can be seen that an RSU has

a certificate size of 92 bytes, while that for an OBU is 167 bytes.

It is indicated in section 3.5.1 that an OBU_m with $cert_{OBU_{mji}}$ can generate a valid signature (U'', V'') for an arbitrary message M. Since U'' and V'' are points on the elliptic curve, the signature size in DCS is 42 bytes. Consequently, the communication overhead incurred in a signed message transmitted by an OBU is 209 bytes, which is the certificate size plus the signature size, compared to an overhead of 189 bytes in the ECPP protocol. According to WAVE [2], the maximum payload data size in a signed message is 65.6 Kbytes. Consequently, the ratio of the communication overhead incurred by the DCS scheme to the payload data size is 0.3%, which means that the DCS scheme is feasible with respect to the incurred communication overhead.

3.7.5 OBU Message Signing Delay

In DCS, the signature of an OBU_m with $cert_{OBU_{mji}}$ on an arbitrary message M is (U'', V''). The cryptography operation involved in calculating either U'' or V'' is point multiplication. Therefore, the total delay for signing a message in DCS is $2T_{mul}$. The second column in Table 3.8 gives the message signing delay for ECDSA, BLS, CAS, ECPP, and DCS. BLS is a pairing-based aggregate signature [55]. CAS is a certificateless aggregate signature scheme [56], which is the basis of the DCS batch verification scheme.

It can be seen that ECDSA and ECPP give the lowest message signing delay, and DCS gives the second lowest delay. The effect of the message signing delay is alleviated by the fact that an OBU has to disseminate only one signed message every 300 $msec$, which means that an OBU has a time window of 300 $msec$ to prepare a signature on a message. The DCS scheme has a message signing delay of 1.2 $msec$, which can be neglected compared to the time window an OBU has to sign a message.

Table 3.8: Signing and verification delay

Method	message signing	one signature and certificate verification	K signatures and certificates verifications
ECDSA	T_{mul}	$4T_{mul}$	$4KT_{mul}$
BLS	$T_{mul} + T_{mtp}$	$4T_{pair} + 2T_{mtp}$	$(2K+2)T_{pair} + 2KT_{mtp}$
CAS	$2T_{mul} + T_{mtp}$	$5T_{pair} + 2T_{mtp}$	$(4K+1)T_{pair} + 2KT_{mtp}$
ECPP	T_{mul}	$3T_{pair} + 11T_{mul}$	$3KT_{pair} + 11KT_{mul}$
DCS	$2T_{mul}$	$5T_{pair} + 3T_{mul}$	$5T_{pair} + 3KT_{mul}$

3.7.6 Batch Verification Delay

We compare the verification delay of the DCS batch signature and certificate verification scheme with ECDSA, BLS, CAS, and ECPP.

The time needed to verify one ECDSA signature is $2T_{mul}$, and that for BLS is $2T_{pair} + T_{mtp}$, where T_{mtp} is a map to point hash function. T_{mtp} is found for an MNT curve to be 3.9 $msec$ [57]. We consider the verification delay for a certificate sent with a message signature for ECDSA and BLS to be equal to that of a signature verification. The time needed to verify one CAS signature is $3T_{pair} + 2T_{mtp}$. For CAS, there is no certificate; however, to verify the sender, a check process must be performed which takes $2T_{pair}$. For ECPP, the total verification delay of a certificate and signature is $3T_{pair} + 11T_{mul}$. For the DCS scheme, the verification delay of a certificate and message signature requires $5T_{pair} + 3T_{mul}$, where $5T_{pair}$ corresponds to the pairing operations in the left and right hand sides of eq. (3.12), and $3T_{mul}$ corresponds to the point multiplication operations in $\overline{U}$, $\overline{U}^{\backprime}$, and $\overline{U}^{\backprime\backprime}$. Table 3.8 shows a summary of the verification delays for ECDSA, BLS, CAS, ECPP, and the DCS schemes.

Fig. 3.9 shows the verification delay in $msec$ vs. the number of the received messages.

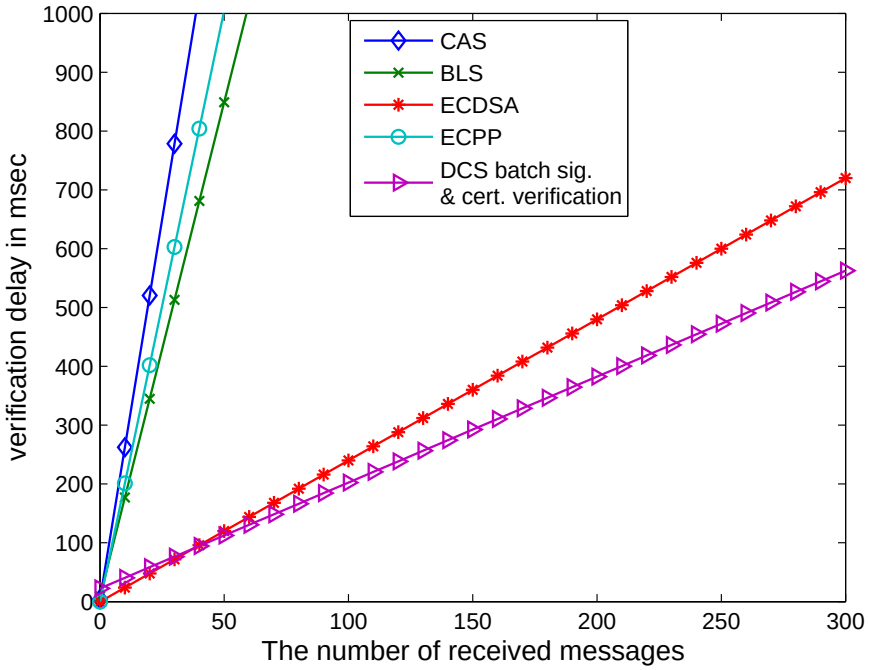


Figure 3.9: Verification delay comparison between different schemes

It can be seen that the DCS scheme has the lowest verification delay. Also, from Table 3.8 and the values of T_{pair}, T_{mtp}, and T_{mul}, the most time-consuming operation in the signature verification process of the schemes under consideration is the pairing operation. Hence, the reason for the superiority of the DCS is that the number of the pairing operations required for signatures verification is independent on the number of the signatures to be verified. The maximum number of signatures and certificates that can be verified simultaneously in 300 $msec$ is 11, 14, 17, 124, and 154 messages for CAS, ECPP, BLS, ECDSA, and the DCS schemes, respectively. The number of signatures and certificates that the DCS scheme can verify is greater than that of the ECDSA by 24.2%. Fig. 3.10 shows the delay for batch signature verification, batch certificate verification, and simultaneous batch signature and certificate verification. The maximum number of certificates that can be verified

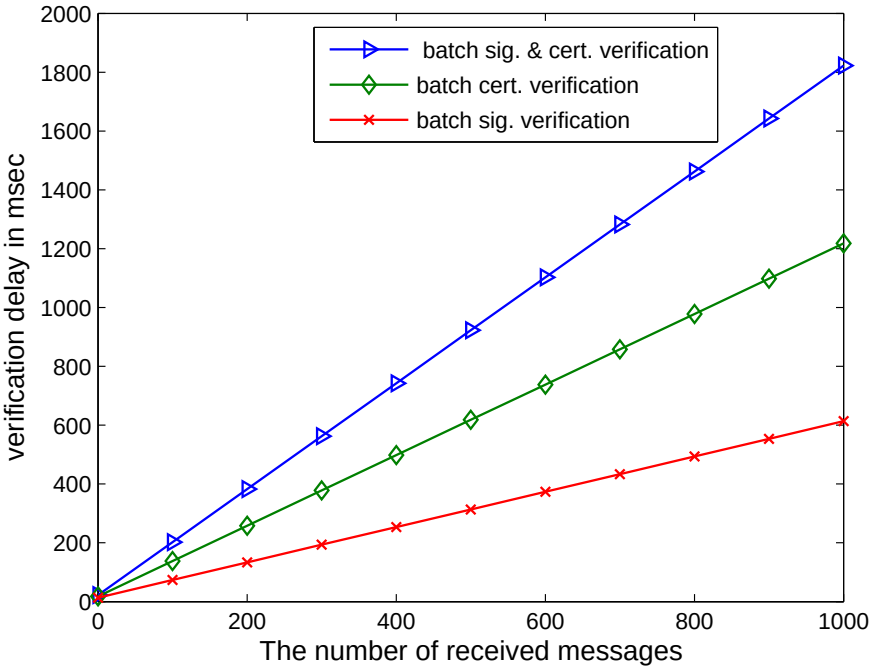


Figure 3.10: Verification delay of the different batch schemes of DCS

aggregately within 300 ms is 234 certificates, while that for signatures is 477 signatures.

To further evaluate the DCS batch verification scheme, we conduct ns-2 [47] simulation using the same parameters in Table 3.3 except for simulation area and time, which become 7.4 $Km \times 7.4$ Km and 30 sec, respectively. In this simulation, we are interested in the message loss incurred by OBUs due to V2V communications only, i.e., we do not consider the implementation of RSUs. The average message loss ratio is defined as the average ratio between the number of messages dropped every 300 $msec$, due to signatures and certificates verification delay, and the total number of messages received every 300 $msec$. According to DSRC, each OBU has to disseminate information about the road condition every 300 $msec$. In order to react properly and instantly to the varying road conditions, each OBU should

verify the messages received during the last 300 *msec* before disseminating a new message about the road condition. Therefore, we chose to measure the message loss ratio every 300 *msec*. Fig. 3.11 shows the analytical and simulated average message loss ratio vs. the average number of OBUs within the communication range of each OBU for DCS, ECPP, ECDSA, BLS, and CAS, respectively. It can be seen that the simulated average message loss ratio closely follows the analytical message loss ratio which is calculated based on the maximum number of messages that can be verified within 300 *msec* in the schemes under consideration. The difference between the analytical and simulations results stems from observing that some zones in the simulated area become more congested than other zones, thus, some OBUs experience higher message loss than other OBUs, which on the average leads to that difference between the analytical and simulation results. Also, the proposed DCS batch verification provides the lowest message loss ratio, and the message loss ratio increases as the number of OBUs within communication range increases. The reason of the superiority of the DCS scheme is that it can aggregately verify a number of signatures higher than that of ECPP, ECDSA, BLS, or CAS.

3.7.7 Additional GPS Memory Requirements

In the DCS scheme, the GPS receiver in each OBU is required to be loaded with the geographic coordinates of the RSUs, which incurs additional memory requirements. According to [2], each latitude or longitude coordinate of the geographic location of an RSU is represented by 4 bytes. With the results obtained in section 3.7.3, the number of RSUs in a CA domain is in the order of 10^4. Consequently, the memory size required to save the coordinates of the RSUs in a domain requires 0.08 Mbytes. Most of the currently available GPS receivers have sufficient memory storage to meet this requirement.

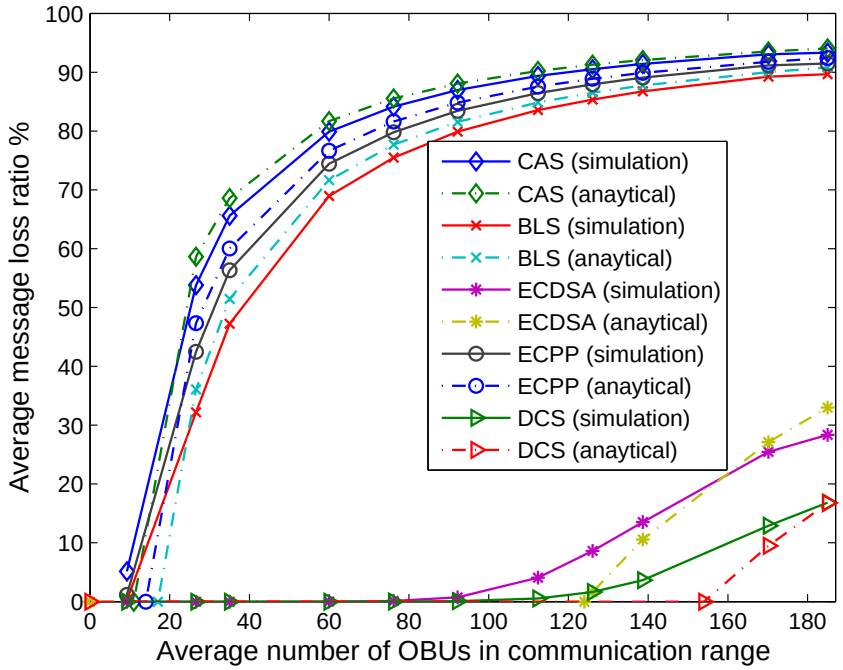


Figure 3.11: Comparison between message loss ratio for different schemes

3.8 Summary

In this Chapter, we have proposed an efficient distributed certificate service (DCS) scheme for vehicular communications, which offers a flexible interoperability to avoid the key escrow issue in different administrative authorities and an efficient distributed algorithm for any OBUs to update or revoke its certificate from the available RSUs in a timely manner. In addition, with the batch verification, the entities in the DCS scheme can rapidly verify a mass of message signatures and certificates simultaneously. Therefore, the proposed DCS scheme can significantly reduce the complexity of certificate management, and achieve excellent efficiency and scalability, especially when it is deployed in heterogeneous vehicular networks.

Chapter 4

EDR: Efficient Decentralized Revocation Protocol for Vehicular Ad Hoc Networks

In this Chapter, we propose an Efficient Decentralized Revocation (EDR) protocol based on a novel pairing-based threshold scheme and a probabilistic key distribution technique. Because of the decentralized nature of the EDR protocol, it enables a group of legitimate vehicles to perform fast revocation of a nearby misbehaving vehicle. Consequently, the EDR protocol improves the safety levels in VANETs as it diminishes the revocation vulnerability window existing in the conventional Certificate Revocation Lists (CRLs). By conducting detailed performance evaluation, the EDR protocol is demonstrated to be reliable, efficient, and scalable.

4.1 Introduction

According to [2], VANET will rely on the Public Key Infrastructure (PKI) as a comprehensive method to secure its operation. In PKI, a central Certification Authority (CA) issues an authentic digital certificate for each node in the network. Efficient certificate management is essential for reliable and robust operation of any PKI. A critical part of any certificate management scheme is the revocation of misbehaving nodes. Certificate revocation can be centralized or decentralized. For centralized revocation, a central entity, such as the CA, is the only entity in the network that can take the revocation decision for a certain node. For decentralized revocation, the node revocation is done by the neighboring nodes of the misbehaving node.

According to the IEEE 1609.2 standard [2], vehicular networks depend on Certificate Revocation Lists (CRLs) and short-lifetime certificates to achieve revocation. In such case, to revoke a vehicle, a CRL has to be issued by the CA and broadcast by the infrastructure RSUs. The network scale of VANETs is expected to be very large. Hence, the distribution of CRLs is prone to long delays [11] [12]. Moreover, centralizing the revocation decision to the CA renders the CA a bottleneck and a single point of failure. In addition, during the early deployment of VANETs, it is expected that RSUs will not be uniformly distributed in the network. Hence, CRL is not proper for applications requiring fast revocation of misbehaving vehicles. Revocation can also be achieved by relying on certificates with short-lifetimes, where a certificate is automatically revoked after its lifetime expires. In VANETs, each vehicle takes life-critical actions based on the received messages from its neighboring vehicles. Hence, VANETs cannot depend solely on the short-lifetime certificates, as a misbehaving vehicle can harm other vehicles until its certificate lifetime expires.

For a practical revocation method, it is required that revocation of misbehaving vehicles

should take place as fast as possible to prevent these vehicles from jeopardizing the safety of other vehicles. Also, the revocation should be done in a decentralized way to alleviate the load on the CA. In addition, the revocation method should be independent of RSUs, which may not be uniformly distributed in the network. Finally, the revocation method should not contradict with other security requirements so that it can be efficiently integrated with other security mechanisms. To address the aforesaid challenges, we propose an efficient decentralized revocation (EDR) protocol for VANETs, which enables a group of neighboring vehicles to revoke a nearby misbehaving vehicle. The EDR protocol is independent of the RSUs and the CA, which makes it suitable for the early deployment phase of VANETs, where a nonuniform RSUs distribution is expected. Also, EDR distributes the revocation load to all the vehicles, thus, avoiding overwhelming the CA. Moreover, it achieves fast revocation of misbehaving vehicles, thus, decreasing the time window during which a misbehaving vehicle can broadcast malicious messages. Consequently, the EDR protocol increases the security level provided by VANETs. In addition, the revocation messages, broadcast by the vehicles, have a security strength equivalent to that of the revocation messages issued by the CA. The EDR protocol has a modular nature which makes it integrable with any PKI system. It can also be used as a stand alone revocation protocol or integrated with the CRL technique to compensate the absence of the RSUs in some areas.

The remainder of the Chapter is organized as follows. In section 4.2, preliminaries are presented. The proposed EDR protocol is presented in section 4.3. The performance analysis of the EDR protocol is discussed in section 4.4. Section 4.5 evaluates the EDR protocol from the security point of view. Section 4.6 outlines how DCS and EDR can be integrated followed by the summary in section 4.7.

4.2 Preliminaries

In this section, we present the system model and security model adopted by the EDR protocol.

4.2.1 System Model

As shown in Fig. 4.1, the system model under consideration consists of the followings.

- A Certification Authority (CA), which is responsible for generating initial certificates for all the vehicles in the network, and it also acts as a Key Distribution Center (KDC). Therefore, the CA is also responsible for distributing keys to all the vehicles in the network;
- Road-Side Units (RSUs), which are fixed units distributed in the network. RSUs can communicate securely with the CA;
- Vehicles, which can communicate either with other vehicles through Vehicle-to-Vehicle (V2V) communications or with the infrastructure RSUs through Vehicle-to-Infrastructure (V2I) communications.

It should be noted that the system model under consideration is mainly a PKI system, where each vehicle has a short-lifetime certificate used to secure its communication with other entities in the network.

4.2.2 Security Model

In this section, we outline the security model adopted by the EDR protocol as follows.

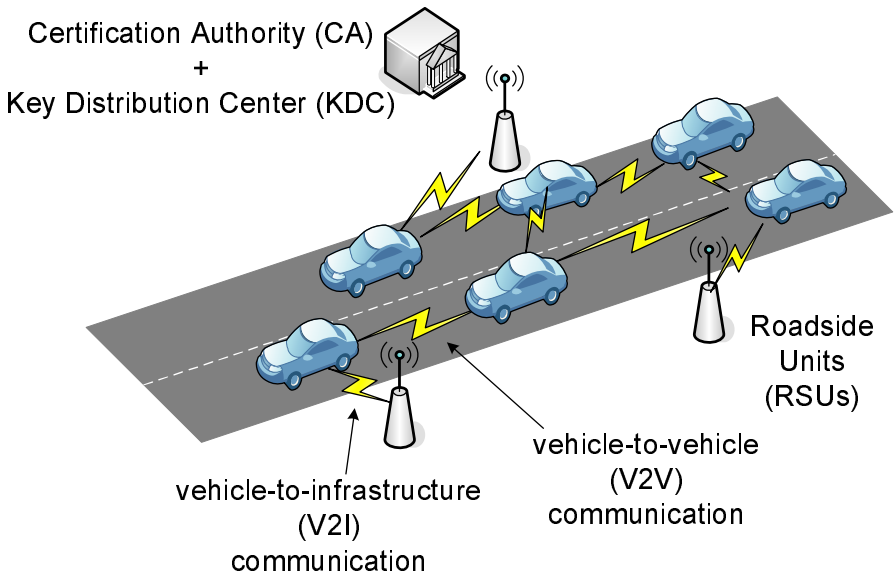


Figure 4.1: The system model.

- The CA is fully trusted by all the network entities. In addition, it has sufficient physical securing mechanisms such that it cannot be compromised by any attacker regardless of his capabilities;
- The RSUs are fixed in place, and they are fully controlled by the CA. Moreover, the CA can instantly quarantine any compromised RSU;
- Vehicles have abundant resources in computation and storage. In addition, vehicles can move freely in the network, and they can be easily compromised by an attacker;
- Revoked vehicles can collude together trying to revoke a legitimate innocent vehicle;
- Legitimate vehicles do not have sufficient incentives to disclose security materials to the revoked vehicles, i.e., legitimate vehicles cannot collaborate with the revoked

vehicles.

4.2.3 Security Objectives

On the design of the EDR protocol, we aim at achieving the following security objectives.

- *Resistance to forging attacks*: The generated revocation messages in the EDR protocol should be unforgeable such that any entity in the network must not be able to generate a fake revocation message even if it has previously generated revocation messages.
- *Resistance to collusion attacks*: The revoked vehicles must not be able to collude together to revoke an innocent vehicle.
- *Resistance to internal revocation-denial attacks*: A legitimate vehicle should not be able to deliberately fail the revocation process of a misbehaving vehicle.
- *Resistance to external revocation-denial attacks*: An external attacker is defined as the attacker who has neither a valid certificate nor valid keys. An external attacker must be able to neither share illegitimately in any revocation process nor fail the revocation process of a misbehaving vehicle.

4.3 The EDR Protocol

The revocation process of any misbehaving vehicle consists of two phases as follows. The first phase is the detection of the misbehaving vehicle, which can be performed by any group vehicles by voting [35] on the misbehavior of a nearby vehicle or by any intrusion detection

system. The second phase is the actual revoking of the misbehaving vehicle whether this is done done using centralized mechanism via the CA or decentralized method via a group of vehicles. In this phase, a revocation message is generated including the certificate of the misbehaving vehicle.

In this Chapter, we present the EDR protocol which is mainly focusing on the second phase of the revocation process, i.e., the actual revoking of misbehaving vehicles. The proposed EDR protocol is based on probabilistic random key distribution technique and a novel pairing-based threshold scheme.

4.3.1 System Initialization

The system is initialized as follows.

1. The CA issues a short-lifetime certificate for each vehicle in the network. Each vehicle can update its certificate from either the RSUs or the CA;

2. Initially, the CA selects a generator $P \in \mathbb{G}_1$ of order q, and key pool consisting of l keys, where each key $k_j \in \mathbb{Z}_q$ has a fixed identity $j \in \{1, 2, \cdots, l\}$. Each vehicle in the network randomly picks from the key pool a key set (R) consisting of m distinct keys;

3. The CA selects x random revocation secret keys $SK_{SHARE} = \{s_1, s_2, s_3, \ \cdots \ , s_x\} = \{s_i | 1 \leq i \leq x\}$ from the key pool such that $\sum_{i=1}^{x} s_i \mod q = S$, where S is the secret key of the CA to sign a message, and for all $i \in [1, x] \ \exists \ j \in [1, l]$, such that $s_i = k_j$. It should be noted that S and the revocation secret keys are equivalent to a threshold scheme [58], where the key S is equivalent to the secret to be shared between multiple entities, and the revocation secret keys are

equivalent to the shadows. Also, the CA calculates the revocation public keys $PK_{SHARE} = \{PK_{share_1},\ PK_{share_2}, \cdots,\ PK_{share_x}\} = \{PK_{SHARE_i} | 1 \leq i \leq x\} = \{\frac{1}{s_1}P,\ \frac{1}{s_2}P, \cdots,\ \frac{1}{s_x}P\} = \{\frac{1}{s_i}P | 1 \leq i \leq x\}$ corresponding to the revocation secret keys $SK_{SHARE} = \{s_i | 1 \leq i \leq x\}$. In addition, the CA calculates its public key $P_\circ = SP$ corresponding to the private key S, and chooses a hash function $H : \{0,1\}^* \rightarrow \mathbb{G}_1$;

4. The CA announces $H,\ P_\circ,\ PK_{SHARE}$, and the keys identities (j's) corresponding to the revocation secret keys SK_{SHARE} to all the vehicles.

After the system is initialized, each vehicle should have the following information:

- A short-lifetime certificate;
- A set of m keys;
- The keys identities (j's) corresponding to the revocation secret keys $SK_{SHARE} = \{s_i | 1 \leq i \leq x\}$;
- The revocation public keys $PK_{SHARE} = \{PK_{SHARE_i} | 1 \leq i \leq x\}$;
- The hash function H, P, and the public key $P_\circ$.

The main idea of the proposed protocol is to use the revocation secret keys $SK_{SHARE} = \{s_i | 1 \leq i \leq x\}$ to revoke the PKI certificate of any misbehaving vehicle.

4.3.2 The Revocation Process

A misbehaving vehicle can be revoked as follows.

1. When a vehicle exhibits a misbehavior, its neighbors vote to revoke the misbehaving vehicle. The proposed protocol has a modular nature which makes it integrable with any voting scheme, e.g., the voting scheme proposed in [35]. When the voting exceeds a predefined threshold, the misbehaving vehicle should be revoked;

2. The vehicle, which accumulates votes exceeding the defined threshold to revoke a vehicle, takes the role of the revocation coordinator, or one of the neighbors of the misbehaving vehicle volunteers to take the role of the revocation coordinator;

3. The revocation coordinator broadcasts to its one-hop neighboring vehicles a request to share in the revocation process, and a message msg containing the certificate of the misbehaving vehicle, the reason for revocation, the current time-stamp, the revocation coordinator signature on the entire message msg, and the revocation coordinator certificate;

4. Any vehicle receiving the request and the message msg, verifies the signature of the revocation coordinator on msg using the revocation coordinator's public key contained in its certificate, and checks the time-stamp to ensure the freshness of the message msg. In addition, it searches its key set (R) for revocation secret keys belong to $SK_{SHARE} = \{s_i | 1 \leq i \leq x\}$. For each possessed revocation secret key, it calculates its revocation share as $Rev_i = s_i H(msg) \in \mathbb{G}_1$, where $i \in \{1, 2, 3, \cdots, x\}$, and sends $(i||Rev_i)$ to the revocation coordinator;

5. When the revocation coordinator receives any revocation share $(i||Rev_i)$ calculated by a revocation secret key i, it uses the corresponding revocation public key ($PK_{share_i} = \frac{1}{s_i}P$) to verify the received revocation share by checking that $\hat{e}(Rev_i, PK_{share_i}) =$

$\hat{e}(H(msg), P)$. This verification holds since

$$\begin{aligned} \hat{e}(Rev_i, PK_{share_i}) &= \hat{e}(s_i H(msg), \frac{1}{s_i} P) \\ &= \hat{e}(H(msg), P)^{s_i \cdot \frac{1}{s_i}} \\ &= \hat{e}(H(msg), P) \end{aligned} \tag{4.1}$$

If the revocation share Rev_i does not pass the verification, it is immediately rejected and dropped. Instead of verifying the revocation shares one by one, the revocation coordinator can wait until the revocation shares corresponding to all the x revocation secret keys are received, then, it can simultaneously verify all the x revocation shares by checking that

$$\hat{e}(\sum_{i=1}^{x} Rev_i, P) = \hat{e}(H(msg), P_\circ) \tag{4.2}$$

This verification holds since

$$\begin{aligned} \hat{e}(\sum_{i=1}^{x} Rev_i, P) &= \hat{e}(Rev_1 + Rev_2 + \cdots + Rev_x, P) \\ &= \hat{e}(Rev_1, P)\hat{e}(Rev_2, P) \cdots \hat{e}(Rev_x, P) \\ &= \hat{e}(s_1 H(msg), P)\hat{e}(s_2 H(msg), P) \cdots \hat{e}(s_x H(msg), P) \\ &= \hat{e}(H(msg), P)^{s_1}\hat{e}(H(msg)P, P)^{s_2} \cdots \hat{e}(H(msg), P)^{s_x} \\ &= \hat{e}(H(msg), P)^{s_1 + s_2 + \cdots + s_x} \\ &= \hat{e}(H(msg), P)^{S} \\ &= \hat{e}(H(msg), SP) \\ &= \hat{e}(H(msg), P_\circ) \end{aligned}$$

6. When the revocation coordinator receives and correctly verifies all the required re-

vocation shares, i.e., $Rev_1, Rev_2, \cdots, Rev_x$, the revocation coordinator computes the total revocation message signature as

$$Rev = \sum_{i=1}^{x} Rev_i$$

The total revocation message signature Rev can be verified as follows

$$\hat{e}(Rev, P) = \hat{e}(H(msg), P_\circ) \tag{4.3}$$

The proof of eq. (4.3) follows directly from the proof of eq. (4.2). It should be noted that the CA is also able to revoke any vehicle using its secret revocation key (S) by directly calculating the total revocation message signature $Rev = SH(msg)$. The total revocation message signature issued by the revocation coordinator is identical to that issued by the CA. Hence, the revocation message signature Rev, generated by either the CA or the revocation coordinator, can be verified by any vehicle using the CA public key $P_\circ$ as indicated in eq. (4.3). As a result, a vehicle verifying Rev does not notice any difference between the verification of the revocation messages transmitted by the revocation coordinators and those transmitted by the CA.

7. The revocation coordinator broadcasts a certificate revocation message $Cert_{rev} = \{msg||Rev||T_{stamp}||\ sgn_{coord}\}$ to the neighboring vehicles, where T_{stamp} is the current time stamp, and sgn_{coord} is the signature of the revocation coordinator on $(msg||Rev||T_{stamp})$. Note that the certificate of the revocation coordinator is included in the message msg;

8. Any vehicle receiving $Cert_{rev}$, checks the freshness of the time stamp T_{stamp} compared to that in msg to ensure that the revocation process is done in a timely manner,

verifies the signature of the coordinator sgn_{coord} using the coordinator's public key included in its certificate, and validates Rev as shown in eq. (4.3). Any vehicle verifying $Cert_{rev}$ correctly forwards it to other vehicles. The dissemination of $Cert_{rev}$ continues until the lifetime of the revoked certificate ends;

9. If any RSU captures the message $Cert_{rev}$, it checks the message validity and then, forwards the message to the CA, which keeps a list of the revoked vehicles.

Since the message $Cert_{rev}$ is broadcast to all the vehicles in the neighborhood of the revoked vehicle, all the neighboring vehicles ignore the messages from the revoked vehicle.

4.3.3 Vehicles Rekeying

All the keys of the revoked vehicles are considered compromised. The rekeying process is triggered by the CA when the number of the compromised keys in the key pool or when the number of the compromised revocation secret keys exceeds a predefined threshold. All the legitimate vehicles must update their compromised keys securely [59]. The rekeying process is as follows.

1. The CA searches its database to determine the identity (M) of the non-compromised key k_M that is shared by the majority of the non-revoked vehicles. The CA then generates an intermediate key $k_{im} = f(k_M) \in \mathbb{Z}_q^*$, where f is a family of pseudo-random functions, which is unique and publicly known to all the network entities. This intermediate key is used by all the vehicles to update their compromised keys. Also, the CA calculates the updated revocation public key(s) corresponding to the compromised revocation secret key(s) $s_i = k_j$ as $PK'_{share_i} = \frac{1}{f_{k_{im}}(s_i)}P$, and its new

secret key $S^{\backprime} = \sum_{i=1}^{x} s_i^{\backprime} \mod q$, where

$$s_i^{\backprime} = \begin{cases} f_{k_{im}}(s_i), & \text{if } s_i \text{ is compromised} \\ s_i, & \text{otherwise} \end{cases}$$

In addition, the CA calculates its new public key $P_{\circ}^{\backprime} = S^{\backprime}P$. After that, the CA broadcasts a key update message

$$Kmsg = (M||IDrev_{vehicle}||IDrev_{key}||\{PK_{share_i}^{\backprime}\}||P_{\circ}^{\backprime})$$

where $IDrev_{vehicle}$ is the identities of the revoked vehicles, $IDrev_{key}$ is the identities of the revoked keys, $\{PK_{share_i}^{\backprime}\}$ is the set of updated revocation public keys, and $P_{\circ}^{\backprime}$ is the CA new public key corresponding to the new secret key $S^{\backprime}$. The CA also sends with the previous message its signature $sgn_{Kmsg} = SH(Kmsg)$ on the message $Kmsg$;

2. After receiving the message $Kmsg$ and the signature sgn_{Kmsg}, each vehicle verifies the received message as follows $\hat{e}(sgn_{Kmsg}, P) = \hat{e}(H(Kmsg), P_{\circ})$. This verification holds since

$$\begin{aligned} \hat{e}(sgn_{Kmsg}, P) &= \hat{e}(SH(Kmsg), P) \\ &= \hat{e}(H(Kmsg), SP) \\ &= \hat{e}(H(Kmsg), P_{\circ}) \end{aligned}$$

If the message is correctly verified, the vehicle checks if it has k_M or not. If yes, the vehicle computes the intermediate key k_{im} independently;

3. If a vehicle v does not have the key k_M, it will not be able to update its compromised

keys and must get k_{im} from its neighboring vehicles. The vehicle v broadcasts its certificate and a request to get k_{im}, and starts its own timer;

4. Any neighboring vehicle of the vehicle v having k_{im} uses the public key of the vehicle v, included in its certificate, to encrypt the intermediate key k_{im} and sends the encrypted k_{im} to the vehicle v;

5. If the vehicle v receives the encrypted k_{im}, it uses its secret key to decrypt k_{im}. Otherwise, if the timer of the vehicle v is timed out without receiving the required data, then go to step 3;

6. The revoked vehicles cannot compute k_{im} since they do not have k_M. Also, they cannot receive k_{im} from other vehicles since the key update message contains the identities of the revoked vehicles, which prevents others from forwarding k_{im} to them;

7. If a vehicle possesses a key k_j that is contained in the revoked vehicles key sets, i.e., compromised key, it updates the compromised key as follows

$$k_j^{\backslash} = f_{k_{im}}(k_j);$$

8. After performing the key set update, each vehicle erases k_{im}, the original compromised revocation public keys PK_{share_i}'s, and the original compromised keys k_j's.

Remarks

- Note that if a vehicle missed a rekeying process, it is still able to share in the upcoming revocation processes since only the compromised keys are updated, hence, it can use its non-compromised revocation secret key(s) in the future. However, if the number

of the missed rekeying processes increases, it may be necessary for the vehicle to contact the CA through RSUs to get the required security materials to update its key set.

- It is clear that only one key update message is broadcast after several revocations took place. Consequently, the number of the messages broadcast by the CA is substantially reduced compared with the centralized revocation scheme, where the CA has to broadcast a message for each revocation process. It should be noted that the rekeying process can be done after every revocation process to increase the security level of the proposed protocol. However, this results in increasing the communication overhead of the rekeying process.

- It should be noted that the EDR protocol has a modular nature, which makes it integrable with any PKI system, e.g., the DCS scheme proposed in Chapter 3. In other words, the proposed protocol does not require any modification to the core of the PKI architecture but all that is needed to implement the proposed protocol is to add a KDC to the CA.

- The EDR protocol can be used as a stand alone revocation method or it can co-exist with the conventional CRL revocation method, where the proposed protocol helps in revoking the misbehaving vehicles in geographic areas where RSUs are not available.

- The EDR protocol is suitable not only for VANETs but also for any type of networks employing PKI.

Table 4.1: EDR notations

Symbol	Notation
l	the key pool size of the key server
m	the key set size stored in each vehicle
N	the number of the collaborating vehicles to revoke a vehicle
x	the number of the revocation secret keys
P_x	the probability of having at least one key of x
P_{rev}	the revocation success probability
w	the number of revoked keys
P_{half}	the probability that at least half of the x keys are safe when there are w keys revoked
$P_{rev}(w)$	the probability that the revocation is successful and it is performed with at least $(x/2)$ non-compromised revocation secret keys

4.4 Performance Evaluation

In this section, we evaluate the performance of the EDR protocol in terms of its feasibility and reliability. The notations used throughout the rest of this section are given in Table 4.1.

4.4.1 The Probability of Having at Least One Revocation Secret Key

The probability P_x of having at least one key of the revocation secret keys (x) in the key set of a vehicle can be calculated as

$$P_x = 1 - \frac{\binom{l-x}{m}}{\binom{l}{m}} \tag{4.4}$$

Fig. 4.2 shows P_x as a function in x. It can be seen that P_x increases as x and m increase and l decreases. This can be explained as follows. For a fixed x, the probability that a vehicle has at least one revocation secret key increases with the number of keys (m) a vehicle gets from the key pool. Similar analogy applies to the number of the revocation secret keys x and the key pool size l. Therefore, a desired value for P_x can be achieved by appropriately selecting the values of $l,\ m$, and x.

4.4.2 Revocation Success Probability

In this section, we are interested in calculating the revocation success probability P_{rev}, which is defined as the probability that any N collaborating vehicles have all the required revocation secret keys (x) to revoke a vehicle. The revocation success probability P_{rev} can be calculated as

$$P_{rev} = \left(1 - \frac{\binom{l-1}{m}^N}{\binom{l}{m}^N}\right)^x \tag{4.5}$$

Fig. 4.3 shows the relation between the revocation success probability (P_{rev}) and the number of the collaborating vehicles (N) for different values of $l,\ m$, and x. It can be seen that for constant $l,\ m$, and x, P_{rev} increases as N increases. Generally speaking, the value of N should be set according to the real-life measurements of the average number of vehicles within the communication range of a vehicle. Also, it can be seen that P_{rev} increases as the vehicle key set size (m) increases and the size of the key pool (l) decreases. This is due to the fact that increasing m or decreasing l increases the probability P_x of having

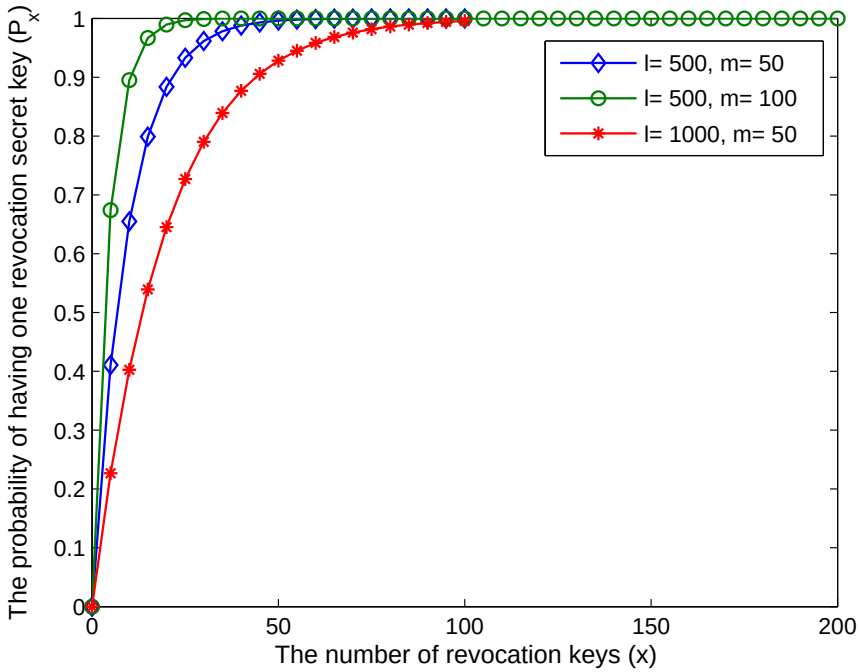


Figure 4.2: The probability (P_x) of having at least one key out of x in the key set of a vehicle.

at least one revocation secret key in the key set of a vehicle, as it can seen in Fig. 4.2, hence, increasing the probability for each vehicle of the neighbors of a misbehaving vehicle to share in the revocation process, which increases the revocation success probability P_{rev}. However, increasing the value of m results in increasing the vulnerability of the system because the more keys a single vehicle has, the more information an attacker can get by compromising a single vehicle. Also, decreasing the value of l results in lowering the security because an attacker gets more information about the key pool if a few number of vehicles are compromised.

From the above discussion, the values of l, m, N, x, and the desired security level should be carefully selected in order to get the desired value of P_{rev}.

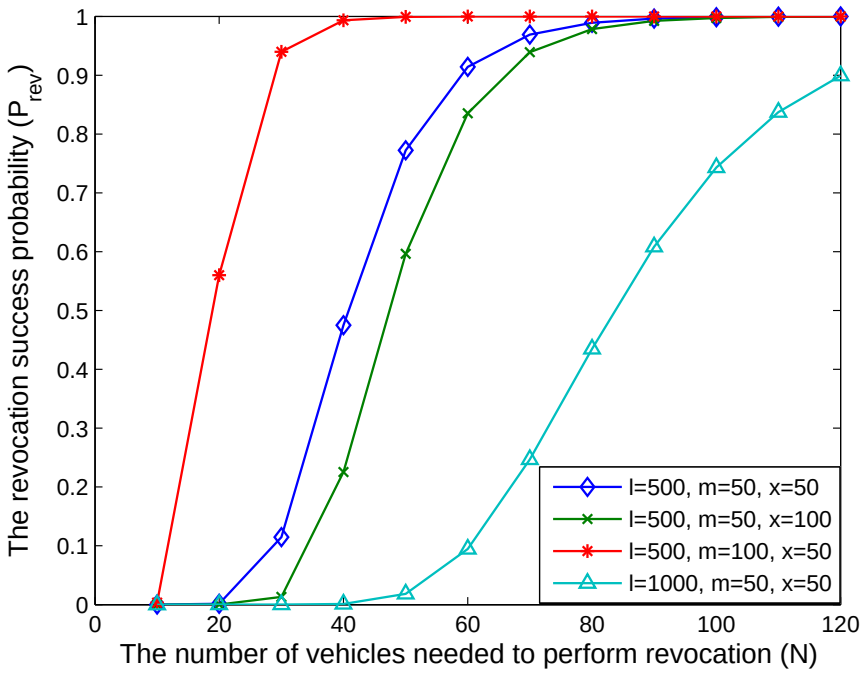


Figure 4.3: The revocation success probability P_{rev}.

4.4.3 The Impact of the Number of the Revoked Keys

In this section, we study the effect of revoking w keys on the safety of the revocation secret keys and the revocation success probability.

To ensure the correctness of the revocation process, we set the following requirement: at least half of the revocation secret keys sharing in the revocation of a vehicle must be non-compromised. It should be noted that the keys of any revoked vehicle are considered compromised. The probability P_{half} that at least half of the x revocation secret keys are safe can be calculated as a function of the number of the revoked keys w as follows

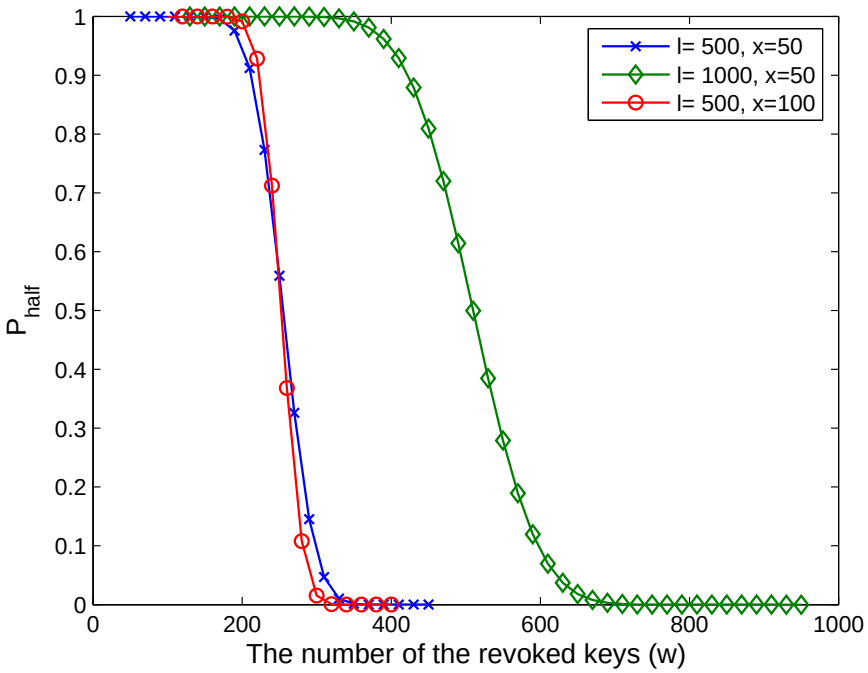


Figure 4.4: The probability that at least half of the revocation secret keys are safe.

$$P_{half} = \sum_{i=x/2}^{x} \frac{\binom{l-w}{i} \cdot \binom{w}{x-i}}{\binom{l}{x}} \tag{4.6}$$

Fig. 4.4 shows the relation between P_{half} and the number of the revoked keys w. It can be seen that changing x has a slight effect on P_{half} because the number of the revocation secret keys (x) is relatively small compared with the number of keys (l) in the key pool, which alleviates the effect of revoking keys from the key pool on the safety of the revocation secret keys. Also, P_{half} decreases as w increases, and P_{half} increases as l increases.

The probability $P_{rev}(w)$ that the revocation is successful and it is performed by at least $(x/2)$ non-compromised revocation secret keys is as follows.

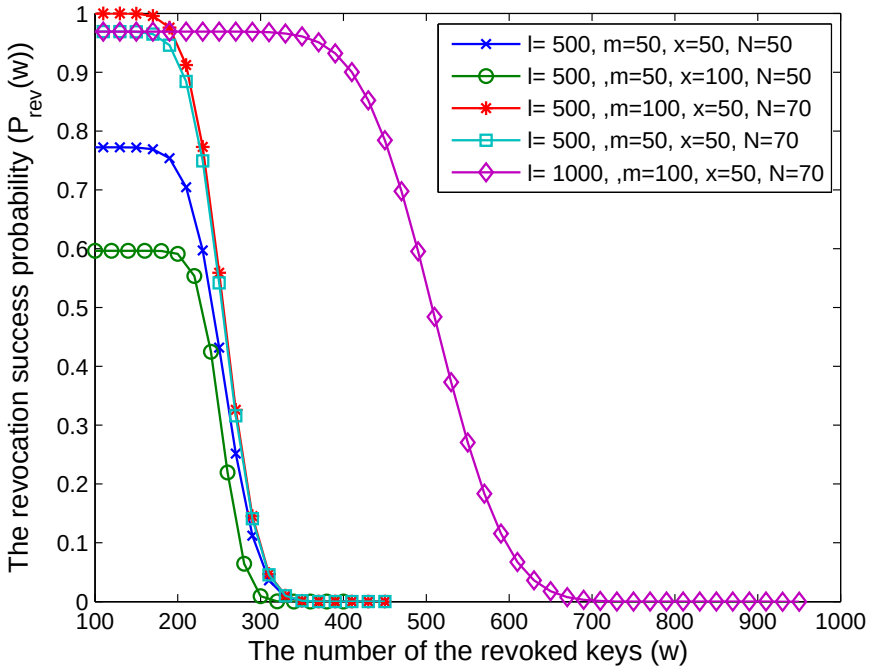


Figure 4.5: The revocation success probability with at least half of the revocation secret keys being safe.

$$
\begin{aligned}
P_{rev}(w) &= P_{rev} \cdot P_{half} \\
&= \left(1 - \frac{\binom{l-1}{m}^N}{\binom{l}{m}^N}\right)^x \cdot \sum_{i=x/2}^{x} \frac{\binom{l-w}{i} \cdot \binom{w}{x-i}}{\binom{l}{x}}
\end{aligned} \tag{4.7}
$$

Fig. 4.5 shows the relation between $P_{rev}(w)$ and w. It can be seen that $P_{rev}(w)$ decreases as w increases. Also, $P_{rev}(w)$ increases as m and N increase, and $P_{rev}(w)$ decreases as l and x increase.

Table 4.2: NS-2 simulation parameters for EDR

simulation area	13.4 $Km \times 12.3$ Km
simulation time	100 sec
max. vehicle speed	60 Km/h
vehicle transmission range	300 m
vehicle information dissemination interval	300 $msec$
number of vehicles	4486
wired channel capacity	100 Mbps
wireless channel capacity	6 Mbps
number of RSUs	576
distribution of RSUs	uniform
key pool size l	500
vehicle key set size m	100
number of revocation secret keys x	20

4.4.4 Revocation Delay

In this section, we evaluate the revocation delay of the EDR protocol and the conventional CRL by conducting ns-2 [47] simulation for the city street scenario shown in Fig. 4.6. The adopted simulation parameters are given in Table 4.2. The mobility traces adopted in this simulation are generated using TraNS [48].

VANETs have two types of links: wireless links connecting vehicles to each other and to the RSUs and wired links connecting the RSUs and the CA as shown in Fig. 4.7. According to the DSRC specifications, each wireless data channel in VANET has a bandwidth of 10 MHz corresponding to channel data rate in the range of 3 $Mbps - 27$ $Mbps$ [50]. We select a data rate of 6 $Mbps$ for the wireless channels in VANET. RSUs are connected via Ethernet to the CA [2]. We consider the links of the Ethernet connecting the RSUs and CA

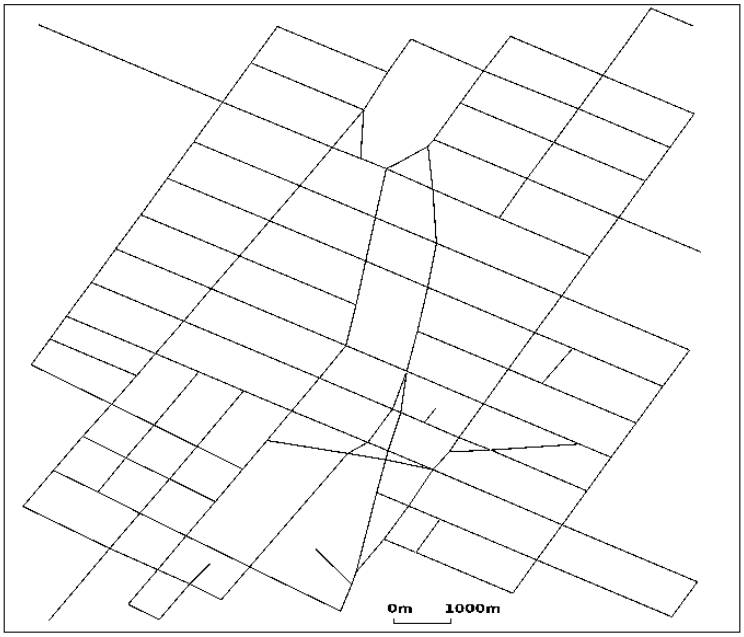


Figure 4.6: A city street simulation scenario

to have data rate of 100 $Mbps$. The RSUs connection pattern employed in our simulation is shown in Fig. 4.8. The adopted RSU connection considers a well deployed VANET, where the RSUs are uniformly distributed with the distance between any pair of adjacent RSUs is 500 m. The CA is located at the top left corner of the city scenario shown in Fig. 4.6. To simulate real-life revocation scenarios, we conduct revocation scenarios imposed on VANET safety-related applications, where each vehicle has to disseminate information about the road condition every 300 $msec$ according to DSRC.

In the conducted simulation, we consider the cryptography delay only due to pairing and point multiplication operations on an elliptic curve as they are the most time consuming operations in the proposed protocol and the conventional CRL. Let T_{pair} and T_{mul} denote the time required to perform a pairing operation and a point multiplication, respectively.

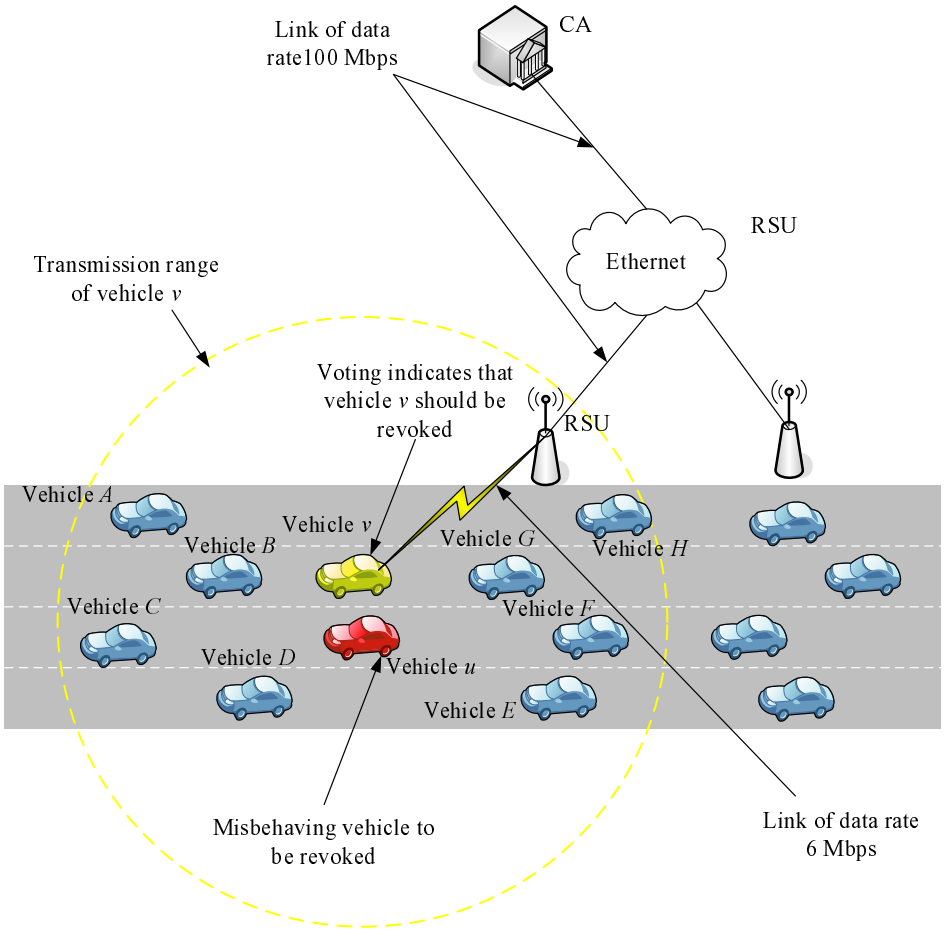


Figure 4.7: Different revocation scenarios

In [26], T_{pair} and T_{mul} are found for an MNT curve with embedding degree $k = 6$ to be equal to 4.5 $msec$, and 0.6 $msec$, respectively. ECDSA [51] is the digital signature method chosen by the VANET standard IEEE1609.2, where a certificate and signature verification takes $4T_{mul}$, and a signature generation takes T_{mul}.

We consider two revocation scenarios as shown in Fig. 4.7. The first scenario is the conventional CRL revocation method combined with a generic voting scheme. In Fig.

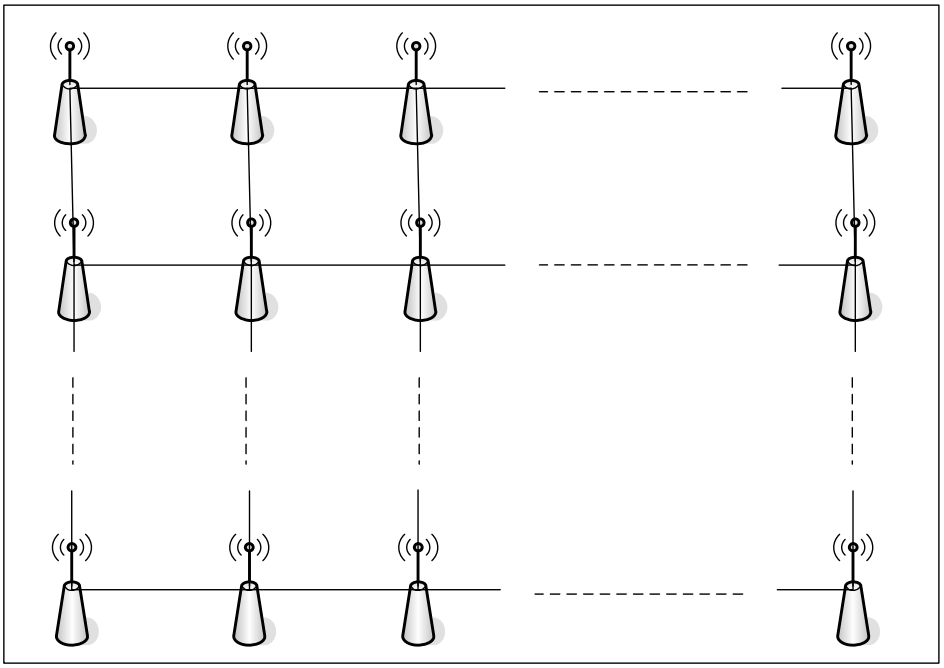


Figure 4.8: RSUs connection pattern

4.7, vehicle u is misbehaving, and the accumulation of the votes in vehicle v reaches the threshold where a revocation of vehicle u should be performed. Hence, vehicle v should send a revocation request to the CA via the nearest RSU. After the request reaches the nearest RSU, the request will be forwarded through the RSUs' Ethernet to the CA, where the request message experiences a delay of $4T_{mul}$ at each intermediate RSU, as each RSU has to verify the certificate and the signature of the sender before forwarding the request. When the revocation request reaches the CA, it has to verify the request which takes $4T_{mul}$, and generate a new signed CRL which takes T_{mul}. In VANETs, the most important issue in any revocation method is the delay of delivering the revocation message to the neighboring vehicles of a misbehaving vehicle to prevent that misbehaving vehicle from jeopardizing the safety of its neighbors. Consequently, the CRL total revocation delay T_{CRL} is the delay

from the moment a vehicle issues a revocation request until the moment the new CRL is broadcast in the geographic area containing vehicle u.

The second scenario is the EDR protocol. In Fig. 4.7, when the accumulation of the votes in vehicle v exceeds the threshold where a revocation of vehicle u should be performed, vehicle v acts as the revocation coordinator and sends a revocation request to the neighboring vehicles located within one hop connectivity (vehicles A, B, $C, \cdots$, and H in Fig. 4.7). Any vehicle receiving the revocation request and having a revocation secret key verifies the request which takes $4T_{mul}$, calculates its revocation share Rev_i which takes T_{mul}, and broadcasts its revocation share. When the revocation coordinator receives the required revocation shares to calculate the final revocation message of vehicle u, it verifies all the revocation shares using eq. (4.2) which takes $2T_{pair} + T_{mul}$, then, it calculates the final revocation message. Finally, vehicle v broadcasts the final revocation messages to its neighboring vehicles. Consequently, the EDR revocation delay T_{EDR} is the delay from the moment the revocation coordinator issues a revocation request until the moment the revocation of vehicle v is broadcast in the geographic area containing vehicle u.

Fig. 4.9 shows the CRL revocation delay T_{CRL} and the EDR revocation delay T_{EDR} in $msec$ vs. the simulation time. We conducted simulation for the two revocation scenarios triggered by the vehicle v at three different locations: location1, location2, and location3 corresponding to initial distances of $2.7Km$, $4.7Km$, and $10.3Km$, respectively, from the CA at the beginning of the simulation. The revocation process is triggered every 10 sec during the simulation and the corresponding revocation delay is measured. The variations in T_{CRL} is due to the number of the intermediate RSUs existing in the connection between the CA and the vehicle sending the revocation request. Also, the variations in T_{EDR} are due to the variation in the number of the neighboring vehicles of the revocation coordinator. It can be seen that T_{EDR} is almost the same for the three locations, and is confined within the

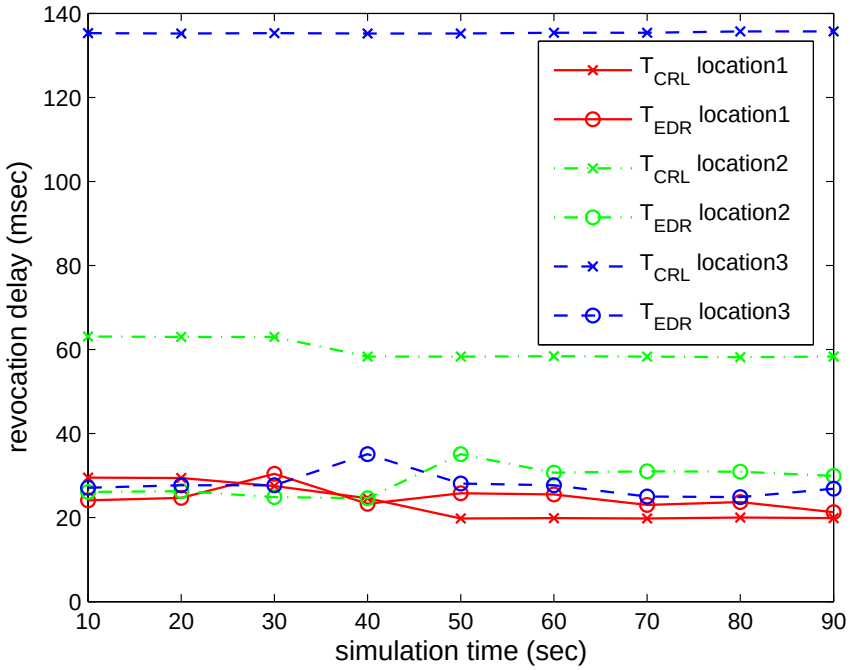


Figure 4.9: The revocation delay for different revocation scenarios

range $21msec - 35msec$. This is due to the fact that the proposed protocol is independent on the CA. On the other hand, it can be seen that T_{CRL} increases with the distance from the CA. Consequently, the delay-saving of the proposed EDR protocol compared to the conventional CRL revocation increases with distance from the CA. For example, the average CRL revocation delay is 59.87 $msec$ for location2, while that for the EDR protocol is 28.83 $msec$. Consequently, the EDR protocol decreases the revocation delay by 51.85% compared to the conventional CRL in that case. It should be noted that T_{EDR} and T_{CRL} are corresponding to the vulnerability window that a misbehaving vehicle has until it is revoked for the EDR protocol and CRL, respectively. During the vulnerability window, the misbehaving vehicle can still jeopardize the safety of the neighboring vehicles. It can be seen that the EDR protocol has a small vulnerability window compared to the CRL

technique, which increases the safety level in VANET.

4.5 Security Analysis

In this section we analyze the proposed protocol against the achieved security objectives in section 4.2.3. It should be noted that these security objectives are the most common revocation attacks.

Resistance to forging attacks

To forge the revocation share $Rev_i = s_i H(msg)$ of any vehicle, an attacker has to solve the following ECDLP problem: given $H(msg)$ and Rev_i, find s_i such that $Rev_i = s_i H(msg)$. Similar analogy applies to finding the CA secret key S from the total revocation message signature $Rev = SH(msg)$. Since ECDLP is a hard computational problem, i.e., it cannot be solved in a sub-exponential time, the revocation shares and the total revocation message signature Rev are unforgeable. Similarly, finding the CA secret value S from $P_\circ = SP$ is ECDLP problem, which makes it unforgeable. Furthermore, the revocation request, sent by the revocation coordinator to his neighboring vehicles, is unforgeable since this request is signed by the revocation coordinator. From the aforementioned discussion, the EDR protocol is resistant to forging attacks.

Resistance to collusion attacks

According to the EDR protocol, the rekeying process is performed before the number of the compromised revocation secret keys exceed half the total number of the revocation secret keys. Therefore, it is guaranteed that the revoked vehicles can never have all the

revocation secret keys, hence, they cannot collude to revoke any vehicle. Consequently, the EDR protocol is resistant to collusion attacks. Moreover, the key update in each vehicle depends mainly on the intermediate key k_{im} which cannot be generated by any revoked vehicle. Also, any compromised vehicle cannot lead to the old k_{im}'s since after each rekeying process, each vehicle erases the current k_{im}. As a result, the revoked vehicles are able to neither update their keys nor share in future revocation processes.

Resistance to internal revocation-denial attacks

If a legitimate vehicle deliberately sends an erroneous revocation share in order to fail the revocation process, the revocation coordinator immediately detects and discards the erroneous revocation share as it will fail to pass the revocation share verification in eq. (4.1). In addition, since the EDR protocol adopts a probabilistic key distribution technique, the same revocation secret key may be found with more than one vehicle. Consequently, the revocation coordinator may receive multiple copies of the same revocation share Rev_i alleviating the effect of a vehicle intentionally choosing not to send its revocation share. Hence, the EDR protocol exhibits robust performance against internal revocation-denial attacks.

Resistance to external revocation-denial attacks

If an external attacker tries to send a fake revocation share during the revocation process, the revocation coordinator will immediately detect and exclude the fake revocation share as it will fail to pass the revocation share verification in eq. (4.1). Consequently, the EDR protocol is resistant to external revocation-denial attacks.

4.6 Integrating DCS and EDR

The EDR protocol presented in this Chapter can be integrated with the DCS scheme presented in Chapter 3 as follows.

- During the system initialization in the DCS scheme, the MA generates a key pool, selects x random revocation secret keys from that key pool, and calculates the corresponding revocation public keys as indicated in subsection 4.3.1.
- The MA copies the key pool and the random revocation secret keys in each CA.
- During the initialization of the OBUs, each OBU randomly selects m keys from the key pool of its in charge CA. Also, each CA loads in each OBU the keys identities corresponding to the revocation secret keys and the revocation public keys.
- Whenever a misbehaving vehicle is detected, the revocation process can be done as indicated in subsection 4.3.2.
- Since the key pool and the revocation secret keys are the same in each CA, the OBUs can roam between different domains and still be able to share in the revocation process in any domain.
- The rekeying process is preformed individually by each CA.
- The integrated DCS and EDR offer an efficient PKI management.

4.7 Summary

In this Chapter, we have proposed a robust and efficient decentralized revocation (EDR) protocol for VANETs, which substantially reduces the complexity of the certificate revo-

cation problem, while achieving fast revocation of the misbehaving vehicles. The EDR protocol decreases the vulnerability window that a misbehaving vehicle has resulting in higher safety level for VANET. The EDR protocol is resistant to the most known revocation attacks. In addition, it can be efficiently integrated with any PKI, such as the DCS scheme proposed in Chapter 3, and/or any misbehavior detection scheme for VANETs. The DCS and EDR schemes can provide the targeted efficient PKI management to provide most of the security requirements for VANETs.

Chapter 5

Complementing Public Key Infrastructure

As discussed in Chapter 2, PKI is a prominent candidate for securing VANETs, however, it cannot provide important security services such as location privacy and expedite authentication. In this Chapter, we propose complementing PKI with group communication to achieve location privacy and expedite message authentication. In specific, the proposed complemented PKI features the following. First, it employs a probabilistic key distribution to establish a shared secret group key between non-revoked OBUs. Second, it uses the shared secret group key to perform expedite message authentication (EMAP) which replaces the time-consuming CRL checking process by an efficient revocation checking process. Third, it uses the shared secret group key to provide novel location privacy preservation through random encryption periods (REP) which ensures that the requirements to track a vehicle are always violated. Moreover, in case of revocation an OBU can calculate the new group key and update its compromised keys even if the OBU missed

previous rekeying process. By conducting detailed analysis and simulation, the proposed complemented PKI is demonstrated to be reliable, efficient, and scalable.

5.1 Introduction

Public Key Infrastructure (PKI) is a well-recognized solution to secure VANETs. However, PKI cannot provide vital security services for VANETs such as location privacy and expedite authentication.

Traditional PKI uses Certificate Revocation Lists (CRLs) for managing the revoked certificates. In PKI, each entity in the network holds an authentic certificate, and every message should be digitally signed before its transmission. A CRL, usually issued by a Trusted Authority (TA), is a list containing all the revoked certificates. In a PKI system, the authentication of any message is performed by first checking if the sender's certificate is included in the current CRL, i.e., checking its revocation status, then, verifying the sender's certificate, and finally verifying the sender's signature on the received message. In Chapter 3, we proposed batch verification to expedite the second and the third parts. The first part of the authentication, which checks the revocation status of the sender in a CRL, may incur long delay depending on the CRL size and the employed mechanism for searching the CRL. Unfortunately, the CRL size in VANETs is expected to be large for the following reasons: (1) To preserve the privacy of the drivers, i.e., to abstain the leakage of the real identities and location information of the drivers from any external eavesdropper [6][7], each OBU should be preloaded with a set of anonymous digital certificates, where the OBU has to periodically change its anonymous certificate to mislead attackers [4]. Consequently, a revocation of an OBU results in revoking all the certificates carried by that OBU leading to a large increase in the CRL size; (2) The scale of VANET is very

large. According to the United States Bureau of Transit Statistics, there are approximately 251 million OBUs in the Unites States in 2006 [60]. Since the number of the OBUs is huge and each OBU has a set of certificates, the CRL size will increase dramatically if only a small portion of the OBUs is revoked. To have an idea of how large the CRL size can be, consider the case where only 100 OBUs are revoked, and each OBU has $25,000$ certificates [61]. In this case, the CRL contains 2.5 million revoked certificates. According to the employed mechanism for searching a CRL, the Wireless Access in Vehicular Environments (WAVE) standard [2] does not state that either a non-optimized search algorithm, e.g., linear search, or some sort of optimized search algorithm such as binary search, will be used for searching a CRL. In this Chapter, we consider both non-optimized and optimized search algorithms.

According to the Dedicated Short Range Communication (DSRC) [3], which is part of the WAVE standard, each OBU has to broadcast a message every 300 $msec$ about its location, velocity, and other telematic information. In such scenario, each OBU may receive a large number of messages every 300 $msec$, and it has to check the current CRL for all the received certificates, which may incur long authentication delay depending on the CRL size and the number of received certificates. The ability to check a CRL for a large number of certificates in a timely manner leads an inevitable challenge to VANETs.

To ensure reliable operation of VANETs and increase the amount of authentic information gained from the received messages, each OBU should be able to check the revocation status of all the received certificates in a timely manner. Most of the existing works overlooked the authentication delay resulting from checking the CRL for each received certificate.

Moreover, as discussed in Chapter 2, PKI and anonymous authentication cannot provide location privacy as an eavesdropper may still able to track a vehicle [30][62][6][63][64][7][21].

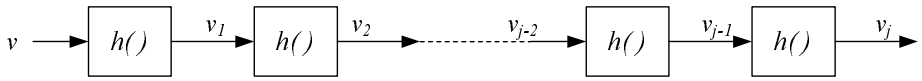


Figure 5.1: Hash chain

Consequently, PKI should be complemented to provide location privacy. One possible solution to achieve location privacy for VANETs is to prevent attackers from gaining any useful information from the periodically multicasted messages.

The remainder of the chapter is organized as follows. In section 5.2, preliminaries are presented. The proposed protocols are introduced in section 5.3. The security analysis and performance evaluation are presented in sections 5.7 and 5.8, respectively. Section 5.9 discusses how to complement DCS with EMAP and REP. The summary of the Chapter is introduced in section 5.10.

5.2 Preliminaries

5.2.1 Hash Chains

A hash chain [65] is the successive application of a hash function $h : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ with a secret value as its input. A hash function is easy and efficient to compute, but it is computationally infeasible to invert. Fig. 5.1 shows the application of a hash chain to a secret value v, where $v_0 = v, v_i = h(v_{i-1}) \; \forall \; 1 \leq i \leq j$.

5.2.2 Search Algorithms

The WAVE standard does not consider a specific mechanism for searching CRLs to check the revocation status of certificates. The most common search algorithms [66] include non-optimized search algorithms such as linear search algorithm, and optimized search algorithms such as binary search algorithm and lookup hash tables. The basic concept of each algorithm is as follows.

Linear Search Algorithm

In the linear search algorithm, the revocation status of a certificate is checked by comparing the certificate with each entry in the CRL. If a match occurs, the certificate is revoked and vice versa.

Binary Search Algorithm

The binary search algorithm works only on sorted lists. Consequently, upon receiving a new CRL, each OBU has to maintain a sorted (with respect to the certificate's identity) database of the revoked certificates included in previous CRLs and the recently received CRL. The main idea of the binary search algorithm is to cancel out half of the entries under consideration after each comparison in the search process. In the binary search, the revocation status of a certificate is checked by comparing the identity of the certificate with middle value (which in this case will be the median value) of the sorted database. If the identity of the certificate is greater than the median value, the right half of the database will be considered in the next comparison process and vice versa. This process continues until a match is found, i.e., the certificate is revoked, or the process is finished without finding a match which means that the certificate is unrevoked.

Lookup Hash Tables

In this approach, the set of all possible certificates ($\mathcal{U}$) is mapped using a hash function into a table of n entries. To check the revocation status of a certificate, the hash of the certificate's identity is the index of the entry in the lookup table which should be checked to determine the revocation status of the certificate. If nil is found in that entry, the certificate under consideration is unrevoked and vice versa. Since VANETs scale is very large and each OBU has a set of certificates, the size of $\mathcal{U}$ will be huge compared to the size (n) of the lookup table. Consequently, the probability of hash collisions will be high, which directly translates to a high probability of false positives. Here, a false positive means that the certificate of an innocent OBU is falsely considered revoked which results in rejecting all the messages containing the certificate of that OBU. The rejected messages may include a warning from dangerous situations. Hence, rejecting these messages may deprive the recipient OBU from taking the appropriate countermeasures to ensure its safety. Accordingly, lookup hash tables may not be practical for VANETs. Hence, lookup hash tables will not be considered in this Chapter. It should be noted that hash functions which map an input to one entry of possible n entries used in the lookup tables, are different from cryptographic hash functions which map an input to a unique output. Throughout the rest of the Chapter, the considered hash functions are cryptographic hash functions.

5.2.3 Location Privacy Threat Model

We consider an external passive global observer, which can overhear and correlate any message broadcast in clear in the network. The anonymity set is defined as the set of all possible OBUs which simultaneously change their anonymous certificate between two observation points controlled by an attacker [29]. Consider an OBU moving between two

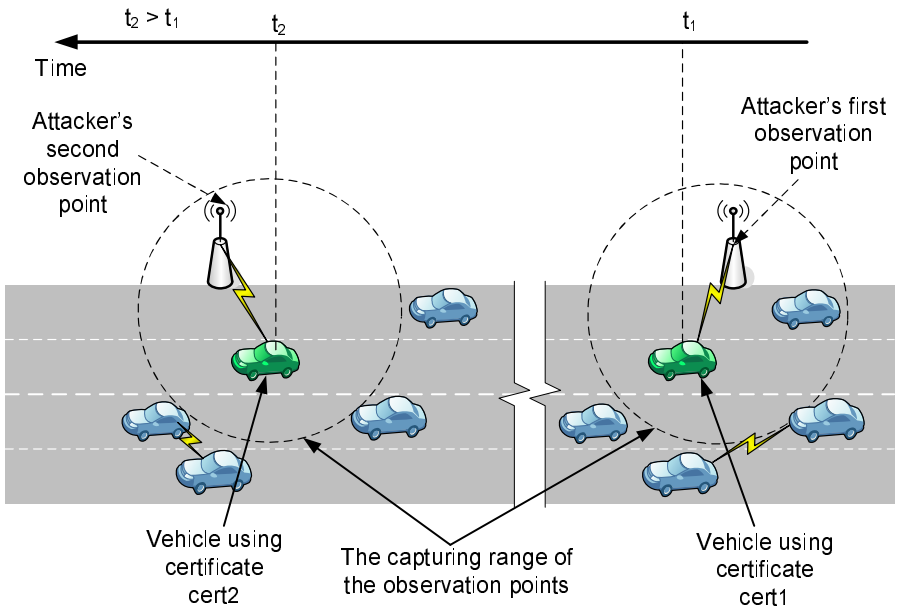


Figure 5.2: The threat model

observation points controlled by the global observer as shown in Fig. 5.2 [30]. The observer can track an OBU if two anonymous certificates can be correctly correlated. This correlation can be achieved by capturing at least one message at each observation point from the OBU, while it is moving with the same speed and in the same lane for some distance between observation points controlled by the observer. For example, a message is captured at the first observation point at time $t1$ from an OBU moving with speed v, in lane L, and using anonymous certificate $cert1$. Given the speed of the OBU and the distance between the two observation points, the observer can expect the time $t2$ to receive a message from that OBU at the second observation point. If a message is captured at the second observation point at time $t2$ from an OBU moving with the same speed v, in lane L, and using anonymous certificate $cert2$, the observer can conclude that $cert1$ and $cert2$ belong to the same OBU. Also, if the OBU under attack is the only OBU, which changes

its certificate in the area between the two observation points, i.e., it has anonymity set size equals one, the observer can track that OBU even if it changes its speed or lane. It can be seen from the threat model that location privacy can be achieved only if the anonymity set size is greater than one and the OBUs, which changed their certificates, change their speeds and/or their lane locations. In addition, it can be seen that periodically changing the anonymous certificate of each OBU in PKI security architectures is insufficient to provide location privacy for VANETs. Consequently, PKI architectures should be combined with other methods to achieve robust location privacy.

5.3 The Proposed Complemented PKI

In this section, we introduce the proposed security architecture.

5.3.1 System Model

As shown in Fig. 5.3, the system model under consideration consists of the followings.

- A Trusted Authority (TA), which is responsible for providing anonymous certificates and distributing secret keys to all OBUs in the network;
- Roadside units (RSUs), which are fixed units distributed all over the network. The RSUs can communicate securely with the TA;
- On-Board Units (OBUs), which are embedded in vehicles. OBUs can communicate either with other OBUs through V2V communications or with RSUs through V2I communications.

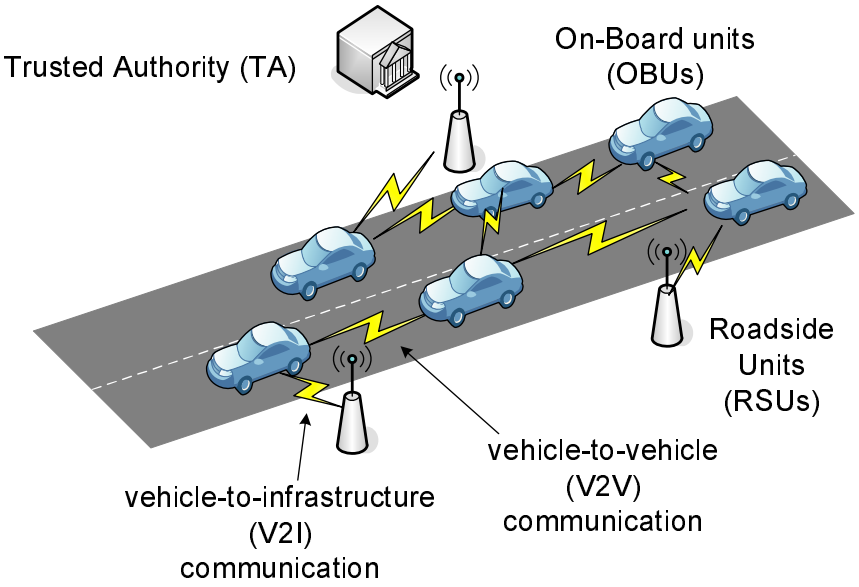


Figure 5.3: The system model

According to the WAVE standard [2], each OBU is equipped with a Hardware Security Module (HSM), which is a tamper-resistant module used to store the security materials, e.g., secret keys, certificates, etc., of the OBU. Also, the HSM in each OBU is responsible for performing all the cryptographic operations such as signing messages, verifying certificates, keys updating, etc. We consider that legitimate OBUs cannot collude with the revoked OBUs as it is difficult for legitimate OBUs to extract their security materials from their HSMs. Finally, we consider that a compromised OBU is instantly detected by the TA.

5.3.2 System Initialization

The TA initializes the system by executing Algorithm 4. In step (20), it should be noted

Algorithm 4 System initialization

Select two generators $P,\ Q \in \mathbb{G}_1$ of order q,
for $i \leftarrow 1, l$ **do**
    Select a random number $k_i \in \mathbb{Z}_q^*$
    Set the secret key $K_i^- = k_i Q \in \mathbb{G}_1$
    Set the corresponding public key $K_i^+ = \frac{1}{k_i} P \in \mathbb{G}_1$
end for
Select an initial secret group key $K_g \in \mathbb{G}_2$ ▷ to be shared between all the non-revoked OBUs
Select a master secret key $s \in \mathbb{Z}_q^*$
Set the corresponding public key $P_\circ = sP$
Choose hash functions $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ and $h : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$
Select a secret value $v \in \mathbb{Z}_q^*$ and set $v_\circ = v$
for $i \leftarrow 1, j$ **do** ▷ to obtain a set V of hash chain values
    Set $v_i = h(v_{i-1})$
end for
for all OBU_u in the network, TA **do**
    for $i \leftarrow 1, m$ **do**
        Select a random number $a \in [1, l]$
        Upload the secret key $K_a^- = k_a Q$ and the corresponding public key $K_a^+ = \frac{1}{k_a} P$ in HSM_u which is the HSM embedded in OBU_u
    end for
    Generate a set of anonymous certificates $CERT_u = \{cert_u^i(PID_u^i, PK_u^i, sig_{TA}(PID_u^i || PK_u^i)) | 1 \leq i \leq C\}$ ▷ for privacy-preserving authentication
    Upload $CERT_u$ in HSM_u of OBU_u
end for
Announce $H,\ h,\ P,\ Q,$ and $P_\circ$ to all the OBUs

that: PK_u^i denotes the i^{th} public key for OBU_u, where the corresponding secret key is SK_u^i; PID_u^i denotes the i^{th} pseudo identity for OBU_u, where the TA is the only entity that can relate PID_u^i to the real identity of OBU_u; $sig_{TA}(PID_u^i||PK_u^i)$ denotes the TA signature on the concatenation ($||$) of PID_u^i and PK_u^i; and C is the number of certificates loaded in each OBU.

After the system is initialized, the TA has the followings:

- A secret key pool $U_s = \{K_i^- = k_i Q | 1 \leq i \leq l\}$;
- The corresponding public key set $U_p = \{K_i^+ = \frac{1}{k_i} P | 1 \leq i \leq l\}$;
- A master secret key s and the corresponding public key $P_\circ$;
- The secret group key K_g;
- A set of hash chain values $V = \{v_i | 0 \leq i \leq j\}$, where j is large enough to accommodate with the number of revocation processes occur during the life-time of the network;
- The public parameters H, h, P, and Q.

Also, each OBU will have the followings:

- A set of anonymous certificates ($CERT_u$) used to achieve privacy-preserving authentication;
- A set of secret keys RS_u consisting of m keys randomly selected from U_s, i.e., $RS_u \subset U_s$;
- The set of the public keys RP_u corresponding to the keys in RS_u, i.e., $RP_u \subset U_p$;

- The secret key K_g, which is shared between all the legitimate OBUs;
- The hash function H, h, P, Q, and the public key $P_\circ$.

Note that the system model under consideration is mainly a PKI system, where each OBU_u has a set of anonymous certificates ($CERT_u$) used to secure its communications with other entities in the network. In specific, the public key PK_u, included in the certificate $cert_u$, and the secret key SK_u are used for verifying and signing messages, respectively. Also, each OBU_u is pre-loaded with a set of asymmetric keys (secret keys K^-'s in RS_u and the corresponding public keys K^+'s in RP_u). Those keys are necessary for generating and maintaining a shared secret group key K_g between unrevoked OBUs. It should be noted also that the secret group key K_g can be used in any application to secure the communication between OBUs.

5.4 Expedite Message Authentication (EMAP)

The proposed EMAP uses a fast *HMAC* function and the established shared secret group key K_g.

5.4.1 Message Authentication

Since we adopt a generic PKI system, the details of the TA signature on a certificate and an OBU signature on a message are not discussed in this Chapter for the sake of generality. We only focus in how to accelerate the revocation checking process, which is conventionally performed by checking the CRL for every received certificate. The message signing and verification between different entities in the network are performed as follows.

Message Signing

Before any OBU_u broadcasts a message $\mathcal{M}$, it calculates its revocation check REV_{check} as $REV_{check} = HMAC(K_g, PID_u||T_{stamp})$ [1], where T_{stamp} is the current time stamp, and $HMAC(K_g, PID_u||T_{stamp})$ is the hash message authentication code on the concatenation of PID_u and T_{stamp} using the secret key K_g. Then, OBU_u broadcasts $(\mathcal{M}||T_{stamp}||cert_u(PID_u, PK_u, sig_{TA}(PID_u||PK_u))||sig_u(\mathcal{M}||T_{stamp})||REV_{check})$, where $sig_u(\mathcal{M}||T_{stamp})$ is the signature of OBU_u on the concatenation of the message $\mathcal{M}$ and T_{stamp}.

Message Verification

Any OBU_y receiving the message $(\mathcal{M}||T_{stamp}||cert_u(PID_u, PK_u, sig_{TA}(PID_u||PK_u))||sig_u(\mathcal{M}||T_{stamp})||REV_{check})$ can verify it by executing Algorithm 5. In step (5), OBU_y calculates $HMAC(K_g, PID_u||T_{stamp})$ using its K_g on the concatenation $PID_u||T_{stamp}$, and compares the calculated $HMAC(K_g, PID_u||T_{stamp})$ with the received REV_{check}.

5.5 Achieving Location Privacy Using Random Encryption Period (REP)

Each OBU has a set of anonymous certificates used to achieve authentication, non-repudiation, and liability. According to the previous threat model, although each OBU periodically changes its anonymous certificate to protect its privacy, it still can be tracked by a global observer. To overcome this tracking attack, when an OBU needs to change its certificate,

[1] It should be noted that throughout the rest of the Chapter the superscript i will be removed from PID_u^i and PK_u^i for the ease of presentation.

Algorithm 5 Message verification

Require: $(\mathcal{M}||T_{stamp}||cert_u(PID_u, PK_u, sig_{TA}(PID_u||PK_u))||sig_u(\mathcal{M}||T_{stamp})||\ REV_{check})$ and K_g

Check the validity of T_{stamp}
if invalid **then**
    Drop the message
else
    Check $REV_{check} \stackrel{?}{=} HMAC(K_g, PID_u||T_{stamp})$
    if invalid **then**
        Drop the message
    else
        Verify the TA signature on $cert_{OBU_u}$
        if invalid **then**
            Drop the message
        else
            Verify the signature $sig_u(\mathcal{M}||T_{stamp})$ using OBU_u public key (PK_u)
            if invalid **then**
                Drop the message
            else
                Process the message
            end if
        end if
    end if
end if

it uses the secret group key shared between legitimate OBUs to surround itself by an encrypted zone with the aid of its neighboring OBUs.

5.5.1 Random Encryption Periods

From the threat model previously discussed, location privacy can be achieved only if the anonymity set size of the OBUs changing their certificates is greater than one, and those OBUs change their speeds and/or their lane locations. The main idea of the proposed

random encryption periods is to provide location privacy for an OBU changing its certificate by ensuring that the aforementioned conditions are met. Random encryption period is triggered when an OBU needs to change its certificate. The random encryption period uses the secret group key K_g, shared between all the OBUs, to create an encryption zone around the OBU which needs to change its certificate as follows:

- Any OBU_i, before changing its certificate, sends a message $msg = \{request_{REP} || PID_i || T_{REP}\}$ to its neighbors moving in the same direction to announce itself as the random encryption period coordinator, where $request_{REP}$ is a request to start a random encryption period in the transmission range of OBU_i, PID_i is the pseudo ID of OBU_i, and T_{REP} is a random time specifying the encryption duration;
- All the OBUs receiving msg start encrypting their broadcast messages using the group key K_g. We term the OBUs encrypting their messages as the encryption group;
- After encryption starts, OBU_i starts monitoring all the OBUs in the encryption group. Also, it changes its certificate;
- Any OBU in the encryption group checks the remaining validity period of its current certificate. If the remaining validity period is less than T_{REP}, it changes its certificate immediately;
- OBU_i monitors the encryption group for the following conditions:
  1. more than one OBU in the encryption group change their certificates;
  2. the OBUs which changed their certificates change their speeds;
  3. the OBUs which changed their certificates change their lanes or directions;

If the first condition and either the second or the third condition is met by the end of T_{REP}, OBU_i terminates the encryption period by broadcasting a message informing the encryption group to stop encrypting their messages. It should be noted that the required anonymity set size can be increased in the first condition to increase the location privacy level;

- If the conditions to terminate the encryption period are not met before T_{REP}, OBU_i broadcasts another msg requesting to extend the encryption period.

It should be noted that any legitimate OBU outside the encryption group can decrypt the received messages since it has K_g. Also, it can be seen that random encryption period prevents the global observer from overhearing messages in the areas where a certificate update takes place, hence, decreasing the probability of tracking an OBU.

5.6 Revocation and Rekeying

The revocation is triggered by the TA when there is an OBU_u to be revoked. The certificates of OBU_u must be revoked. In addition, the secret key set RS_u of OBU_u and the current secret group key K_g are considered revoked. Hence, a new secret group key $\tilde{K_g}$ should be securely distributed to all the non-revoked OBUs. Also, each non-revoked OBU should securely update the compromised keys in its key sets RS and RP [59], which is called the rekeying process. The revocation and rekeying processes are as follows.

1. The TA searches its database to determine the identity (M) of the non-compromised secret key $K_M^- = k_M Q$ that is shared by the majority of the non-revoked OBUs, and finds the corresponding public key $K_M^+ = \frac{1}{k_M} P$. The TA then selects a random

number $t \in \mathbb{Z}_q^*$, and calculates the intermediate key $K_{im} = tK_M^+ = \frac{t}{k_M}P \in \mathbb{G}_1$, and the new secret group key $\tilde{K_g}$ as follows

$$\begin{aligned} \tilde{K_g} &= \hat{e}(K_M^-, K_{im}) \\ &= \hat{e}(k_M Q, \frac{t}{k_M} P) \\ &= \hat{e}(Q, P)^{k_M \cdot \frac{t}{k_M}} \\ &= \hat{e}(Q, P)^t \end{aligned} \tag{5.1}$$

Also, it selects the value v_{j-ver} of the hash chain values, where v_j is the last value in the hash chain as shown in Fig. 5.1, and ver is an integer indicating the revocation version, i.e., the number of the revocation processes performed since the network initialization. The value v_{j-ver} is used by all the OBUs to update their compromised secret keys and the corresponding public keys. After that, the TA prepares a key update message $Kmsg = (ver||M||IDrev_{key}||K_{im}||enc_{\tilde{K_g}}(v_{j-ver}))$, where $IDrev_{key}$ is a list of the identities of the revoked keys, and $enc_{\tilde{K_g}}(v_{j-ver})$ is the symmetric encryption of v_{j-ver} using the key $\tilde{K_g}$. Finally, the TA broadcasts the following message $REV_{msg} = (CRL||Kmsg||sig_{TA}(CRL||Kmsg))$, where CRL is a list of the certificates of the revoked OBUs, and $sig_{TA}(CRL||Kmsg) = sH(CRL||Kmsg)$ is the TA signature on $CRL||Kmsg$;

2. After receiving the message REV_{msg}, each OBU_y executes Algorithm 6;

3. In Algorithm 6 step (1), OBU_y verifies the signature $sig_{TA}(CRL||Kmsg)$ by checking

that $\hat{e}(sig_{TA}(CRL||Kmsg), P) \stackrel{?}{=} \hat{e}(H(CRL||Kmsg), P_\circ)$. This check follows since

$$\begin{aligned}\hat{e}(sig_{TA}(CRL||Kmsg), P) &= \hat{e}(sH(CRL||Kmsg), P) \\ &= \hat{e}(H(CRL||Kmsg), sP) \\ &= \hat{e}(H(CRL||Kmsg), P_\circ)\end{aligned}$$

Algorithm 6 Processing revocation messages

Require: $REV_{msg} = (CRL||Kmsg||sig_{TA}(CRL||Kmsg))$ and $P_\circ$

1: Verify $sig_{TA}(CRL||Kmsg)$ by checking $\hat{e}(sig_{TA}(CRL||Kmsg), P) \stackrel{?}{=} \hat{e}(H(CRL||Kmsg), P_\circ)$
2: **if** invalid **then**
3: Exit
4: **else**
5: Run Algorithm 7 to get $\tilde{K_g}$ and v_{j-ver}
6: Run Algorithm 8 to update the key set of OBU_y
7: **end if**
8: Store ver and $IDrev_{key}$
9: Erase K_{im}, the hash chain values, and the original compromised secret and public keys.

4. OBU_y has to execute Algorithm 7 to get $\tilde{K_g}$ and v_{j-ver}. If OBU_y has K_M^-, it can independently calculate $\tilde{K_g}$ according to step (2). Otherwise, OBU_y gets $\tilde{K_g}$ from its neighboring OBUs as indicated in steps (5-15);

5. In Algorithm 7, the revoked OBUs cannot compute $\tilde{K_g}$ since they do not have K_M^-. Also, they cannot receive $\tilde{K_g}$ from other OBUs since the recent CRL sent in REV_{msg} contains the certificates of the revoked OBUs, which stops others from forwarding $\tilde{K_g}$ to them;

Algorithm 7 Obtaining $\tilde{K_g}$ and v_{j-ver}

1: **if** K_M^- exists in RS_y **then**
2: Set the new secret group key $\tilde{K_g} = \hat{e}(K_M^-, K_{im})$
3: Decrypt $enc_{\tilde{K_g}}(v_{j-ver})$ using $\tilde{K_g}$ to get v_{j-ver}
4: **else**
5: Broadcast a signed request and $cert_y(PID_y, PK_y, sig_{TA}(PID_y||PK_y))$ to get $\tilde{K_g}$ from neighboring OBUs
6: Start a timer T_1
7: Any neighboring OBU of OBU_y having $\tilde{K_g}$ verifies the signature and certificate of OBU_y, ensures that $cert_y$ is not in the recent CRL, uses the public key (PK_y) of OBU_y included in $cert_y$ to encrypt $\tilde{K_g}$, and sends the encrypted $\tilde{K_g}$ to OBU_y
8: **if** the encrypted $\tilde{K_g}$ is received **then**
9: Decrypt $\tilde{K_g}$ using the secret key corresponding to PK_y
10: Decrypt $enc_{\tilde{K_g}}(v_{j-ver})$ using $\tilde{K_g}$ to get v_{j-ver}
11: **else**
12: **if** T_1 is timed out **then**
13: Go to 5
14: **end if**
15: **end if**
16: **end if**

6. OBU_y has to execute Algorithm 8 to update its key sets RS_y and RP_y. If OBU_y did not miss any previous revocation messages, it updates its key sets as indicated in steps (3-4). If OBU_y missed a number of previous revocation messages, it can update its key sets as indicated in steps (9-22). It should be noted that in step (14), $ver_{|missed}$ and $IDrev_{key|missed}$ denote the revocation version and the list of identities of the revoked keys of a missed revocation process, respectively;

7. It should be noted that in Algorithm 7 step (7) and in Algorithm 8 step (16) one of the communicating parties do not have the new key $\tilde{K_g}$. Accordingly, the OBUs must use the CRL to check that the certificates of the communicating parties are not previously revoked.

Algorithm 8 Rekeying OBU_y (Updating the key sets of OBU_y)

Require: $\tilde{K_g}$ and v_{j-ver}

if not previously missing any revocation message **then**
  if possesses compromised secret keys $\{K_i^-\} = \{k_iQ\}$ in $IDrev_{key}$ **then**
    Update the secret key K_i^- as $\tilde{K}_i^- = v_{j-ver}K_i^- = v_{j-ver}k_iQ$
    Update the corresponding pubic keys $\tilde{K}_i^+ = \frac{1}{v_{j-ver}}K_i^+ = \frac{1}{v_{j-ver}k_i}P$
  else
    Exit
  end if
else
  Set $n = ver$
  while $n \neq v_{ver_{last}}$ **do** ▷ ver_{last} is the last received revocation version
    Set $v_{j-n+1} = h(v_{j-n})$
    Set $n = ver + 1$
  end while ▷ this loop outputs $\{v_{j-ver+1}, v_{j-ver+2}, \cdots, v_{ver_{last}-1}\}$
  Broadcast a signed request to the neighboring OBUs requesting $ver_{|_{missed}}$ and $IDrev_{key|_{missed}}$ for all the missed revocation processes
  for each received signed value of $ver_{|_{missed}}$ **do**
    Verify the signature and certificate of the sender and, ensures that the certificate of the sender is not in the recent CRL
    Find the value of $v_{j-ver_{|_{missed}}}$ from $\{v_{j-ver+1}, v_{j-ver+2}, \cdots, v_{ver_{last}-1}\}$
    for each possessed key $K_i^- = k_iQ \in IDrev_{key|_{missed}}$ **do**
      Update the secret key K_i^- as $\tilde{K}_i^- = v_{j-ver_{|_{missed}}}K_i^- = v_{j-ver_{|_{missed}}}k_iQ$
      Update the corresponding public key as $\tilde{K}_i^+ = \frac{1}{v_{j-ver_{|_{missed}}}}K_i^+ = \frac{1}{v_{j-ver_{|_{missed}}}k_i}P$
    end for
  end for
end if

Remarks

- To perform the revocation and rekeying process the TA needs to send one message only irrespective of the number of the OBUs in the network.
- An important feature of the proposed complemented PKI is that it enables an OBU to update its compromised keys corresponding to previously missed revocation processes provided that it picks one revocation process in the future. To the best of our knowledge, this is the first work to propose a rekeying mechanism capable of updating compromised keys corresponding to previously missed rekeying processes.
- Note that the proposed complementing mechanisms have a modular feature, which makes them integrable with any PKI system. In other words, the proposed complementing mechanisms for PKI do not require any modification to the core of the PKI architecture. It only needs a key distribution module to be added to the TA during the system initialization.
- The complementing mechanisms are suitable for not only VANETs but also any type of networks employing PKI.
- Algorithms 6-8 are executed through the HSM module in each OBU.

5.7 Security Analysis

In this section, we analyze the security of the proposed complementing mechanisms of PKI against some common attacks.

Resistance to forging attacks

To forge the revocation check $REV_{check} = HMAC(K_g, PID_u || T_{stamp})$ of any OBU_u, an attacker has to find the current K_g, which is equivalent to finding t in the following ECDLP problem: given $K_{im} = tK_M^+ = \frac{t}{k_M}P$ and $K_M^+ = \frac{1}{k_M}P$, find t such that $K_{im} = tK_M^+$. Similar analogy applies to finding the TA secret key s from the TA message signature $sgn_{Kmsg} = sH(Kmsg)$. Since ECDLP is a hard computational problem [40], i.e., it cannot be solved in a sub-exponential time, the revocation check and the TA message signature sgn_{Kmsg} are unforgeable. Similarly, finding the TA secret value s from $P_\circ = sP$ is ECDLP problem, which makes it unforgeable. From the aforementioned discussion, it is concluded that EMAP is resistant to forging attacks.

Resistance to replay attacks

Since in each message an OBU includes the current time stamp in the revocation check value $REV_{check} = HMAC(K_g, PID_u||T_{stamp})$, an attacker cannot record REV_{check} at time T_i and replay it at a later time T_{i+1} to pass the revocation checking process as the receiving OBU compares the current time T_{i+1} with that included in the revocation check. Consequently, EMAP is secure against replay attacks.

Forward secrecy

Since the values of the hash chain included in the revocation messages are released to non-revoked OBUs starting from the last value of the hash chain, and given the fact that a hash function is irreversible, a revoked OBU cannot use a hash chain value $v_{j-ver+1}$ received in a previous revocation process to get the current hash chain value v_{j-ver}. Consequently, a revoked OBU cannot update its secret key set (RS). Accordingly, a revoked OBU can

neither get K_M^- necessary to independently calculate the new secret key $\tilde{K_g}$ nor get $\tilde{K_g}$ from the neighboring OBUs since the certificates of the revoked OBUs are in the up-to-date CRL which prevents unrevoked OBUs from forwarding $\tilde{K_g}$ to the revoked OBUs. As a result, the proposed complemented PKI guarantees forward secrecy.

Resistance to colluding attacks

For a colluding attack, a legitimate OBU colludes with a revoked OBU by releasing the current secret group key $\tilde{K_g}$ such that the revoked vehicle can use this key to pass the revocation check process by calculating the correct HMAC values for the transmitted messages. All the security materials of an OBU are stored in its tamper-resistant HSM. In addition, all the keys update processes in Algorithms 6-8 are executed in the HSM, which means that the new secret group key $\tilde{K_g}$ is stored in the HSM, and it cannot be transmitted in clear under any circumstances. Note that in Algorithm 7 step (7) the HSM only sends $\tilde{K_g}$ encrypted with the public key included in the certificate of the OBU requesting $\tilde{K_g}$ after checking that the certificate of that OBU is not in the CRL. Accordingly, only that OBU is the entity that can decrypt and obtain $\tilde{K_g}$ using its secret key which is exclusively known to itself. Since it is infeasible to extract the security materials from the tamper-resistant HSM, an unrevoked OBU cannot collude with a revoked OBU by passing the new secret group key $\tilde{K_g}$ to the revoked OBU. Hence, the proposed complemented PKI is secure against colluding attacks.

5.8 Performance Evaluation

In this section we evaluate the performance of the proposed complemented PKI from different aspects.

5.8.1 Computation Complexity of Revocation Status Checking

We are interested in the computation complexity of the revocation status checking process which is defined as the number of comparison operations required to check the revocation status of an OBU. Let N_{rev} denote the total number of revoked certificates in a CRL. To check the revocation status of an OBU_u using the linear search algorithm, an entity has to compare the certificate identity of OBU_u with every certificate of the N_{rev} certificates in the CRL, i.e., the entity performs one-to-one checking process. Consequently, the computation complexity of employing the linear search algorithm to perform a revocation status checking for an OBU is $O(N_{rev})$. In the binary search algorithm, the certificate identity of OBU_u is compared to the certificate identity in the middle of the sorted CRL. If the certificate identity of OBU_u is greater than that of the entry in the middle, then half of the CRL with identities lower than that of OBU_u are discarded from the upcoming comparisons. If the certificate identity of OBU_u is lower than that of the entry in the middle, then half of the CRL with identities higher than that of OBU_u are discarded. The checking process is repeated until a match is found or the CRL is finished. It can be seen that at each step in the binary search method half of the entries considered in the search is discarded. Thus, the computation complexity of the binary search algorithm to perform a revocation status checking for an OBU is $O(\log N_{rev})$ [66]. In EMAP, the revocation checking process requires only one comparison between the calculated and received values of REV_{check}. As a result, the computation complexity of EMAP is $O(1)$, which is constant and independent of the number of revoked certificates. In other words, EMAP has the lowest computation complexity compared with the CRL checking processes employing linear and binary search algorithms.

5.8.2 Authentication Delay

We compare the message authentication delay employing the CRL with that employing EMAP to check the revocation status of an OBU. As stated earlier, the authentication of any message is performed by three consecutive phases: checking the sender's revocation status, verifying the sender's certificate, and verifying the sender's signature. For the first authentication phase which checks the revocation status of the sender, we employ either the CRL or EMAP. For EMAP, we adopt the Cipher Block Chaining Advanced Encryption Standard (CBC-HMAC AES) [67] and Secure Hash Algorithm 1 SHA-1 [68] as the *HMAC* functions. We consider the pseudo identity (PID) of OBU and the time stamp (T_{stamp}) having equal lengths of 8 bytes. We adopt the Crypto++ library [69] for calculating the delay of the *HMAC* functions, where it is compiled on Intel Core2Duo 2 GHz machine. The delay incurred by using CBC-HMAC AES and SHA-1 to calculate the revocation check ($REV_{check} = HMAC(K_g, PID_u||T_{stamp})$) is 0.23 μsec and 0.42 μsec, respectively. Also, we have simulated the linear and binary CRL checking process using C++ programs compiled on the same machine. The linear CRL checking program performs progressive search on a text file containing the unsorted identities of the revoked certificates, while the binary CRL checking program performs a binary search on a text file containing the sorted identities of the revoked certificates. For the second and third authentication phases, we employ Elliptic Curve Digital Signature Algorithm (ECDSA) [51] to check the authenticity of the certificate and the signature of the sender. ECDSA is the digital signature method chosen by the WAVE standard. In ECDSA, a signature verification takes $2T_{mul}$, where T_{mul} denotes the time required to perform a point multiplication on an elliptic curve. Consequently, the verification of a certificate and message signature takes $4T_{mul}$. In [26], T_{mul} is found for a supersingular curve with embedding degree $k = 6$ to be equal to 0.6 $msec$.

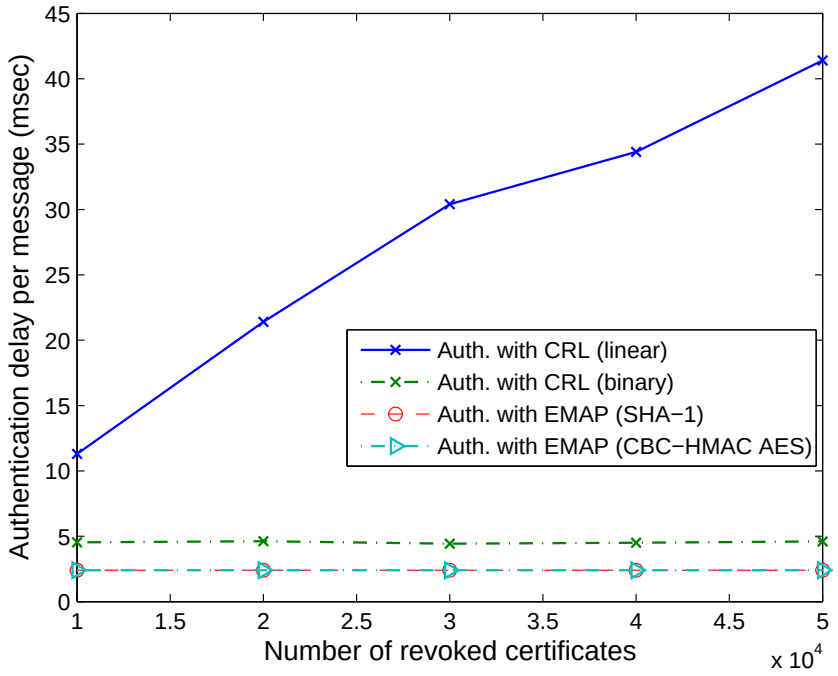


Figure 5.4: Authentication delay per message

Fig. 5.4 shows a comparison between the authentication delay per message using EMAP, linear CRL checking process, and binary CRL checking process vs. the number of the revoked certificates, where the number of the revoked certificates is an indication of the CRL size. It can be seen that the authentication delay using the linear CRL checking process increases with the number of revoked certificates, i.e., with the size of the CRL. Also, the authentication delay using the binary CRL checking process is almost constant. This can be explained as follows: the number of revoked certificates in the conducted simulation ranges from 10000 to 50000 revoked certificates; This is respectively corresponding to 14 to 16 comparison operations. Since the range of the number of the comparison operations is very small, the authentication delay is almost constant. The authentication delay using EMAP is constant and independent of the number of revoked certificates. Moreover,

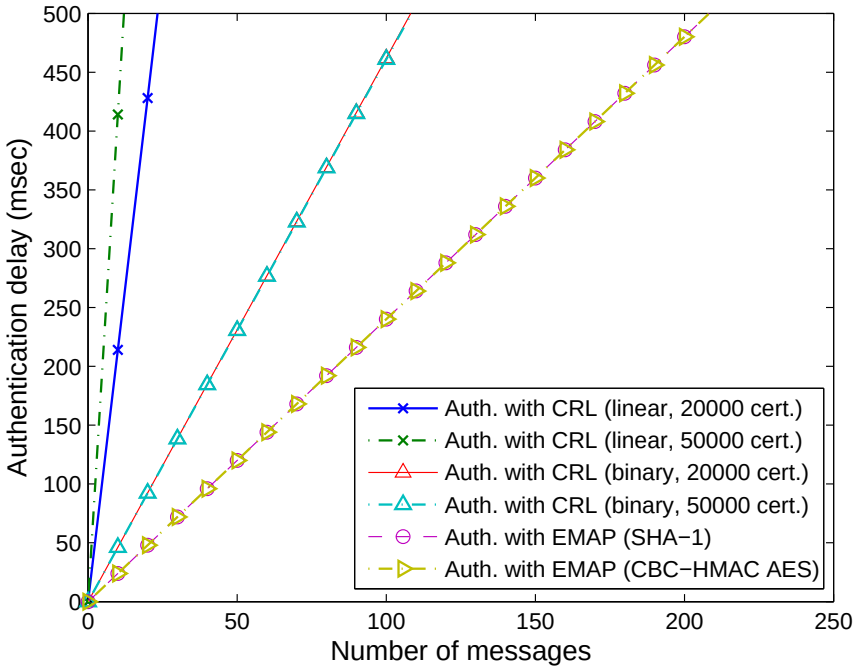


Figure 5.5: Total authentication delay vs. the number of the received messages

the authentication delay using EMAP outperforms that using the linear and binary CRL checking processes. For example, the authentication delay per message using the linear CRL checking process, the binary CRL checking process, and EMAP (SHA-1) for a CRL including 20000 revoked certificates are 21.4 *msec*, 4.62 *msec*, and 2.4004 *msec*, respectively. Consequently, EMAP (SHA-1) expedites the message authentication by 88.78% and 48.04% compared to that using the linear and binary CRL checking processes, respectively. Fig. 5.5 shows the total authentication delay in *msec* vs. the number of messages to be authenticated using EMAP and the linear and binary CRL checking processes. It can be seen that as the CRL size increases the number of messages that can be verified within a specific period is significantly decreased using the linear CRL checking process. Also, for a constant authentication delay, EMAP outperforms the linear and binary CRL

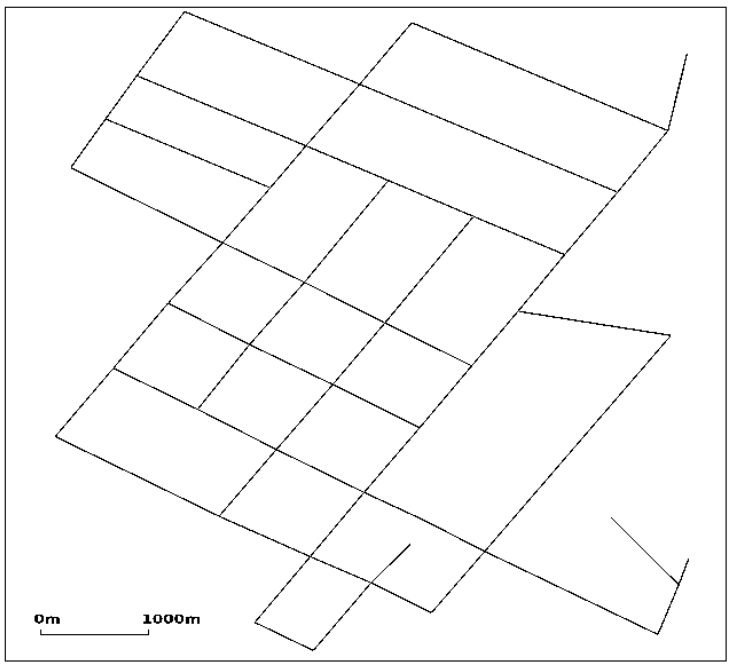


Figure 5.6: A city street simulation scenario

checking processes. The maximum number of messages that can be verified simultaneously in 300 *msec* is 14, 64, and 124 messages for message authentication employing linear CRL checking, binary CRL checking, and EMAP, respectively, where the considered CRL includes $20,000$ certificates. The number of messages that can be verified using EMAP within 300 *msec* is greater than that using linear and binary CRL checking by 88.7% and 48.38%, respectively.

5.8.3 End-to-end delay

To further evaluate EMAP, we have conducted ns-2 [47] simulation for the city street scenario shown in Fig. 5.6. The adopted simulation parameters are given in Table 5.1.

Table 5.1: NS-2 simulation parameters for the proposed complemented PKI

Simulation area	7.4 $Km \times 7.4$ Km
Simulation time	30 sec
Max. OBU speed	60 Km/h
OBU transmission range	300 m
OBU information dissemination interval	300 $msec$
MAC protocol	802.11a
Wireless channel capacity	6 Mbps

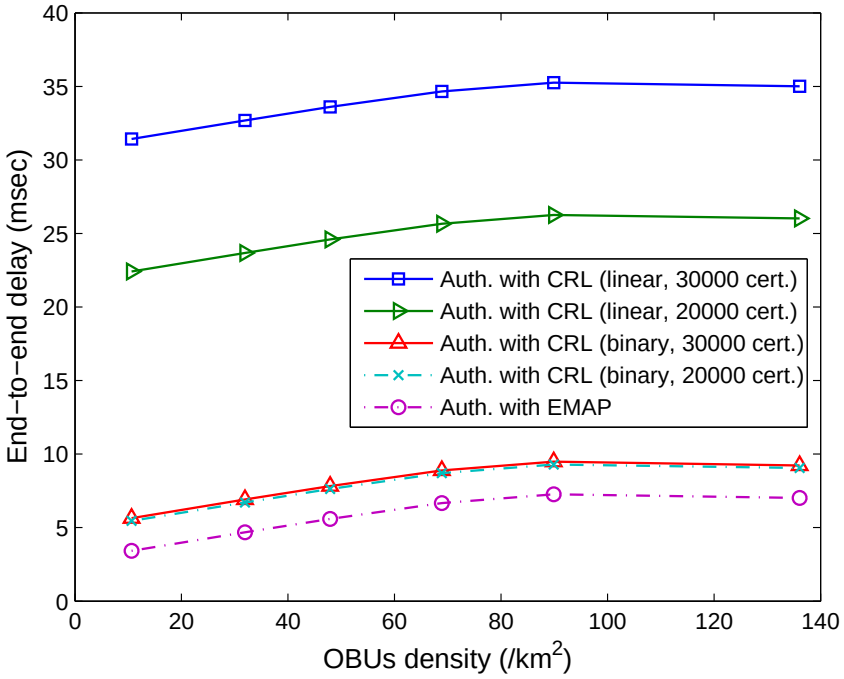


Figure 5.7: End-to-end delay vs. OBUs density

We select the dissemination of the road condition information by an OBU every 300 $msec$ to conform with the DSRC standards. The mobility traces adopted in this simulation are generated using TraNS [48]. We are interested in the end-to-end delay, which is defined as

the time to transmit a message from the sender to the receiver. Fig. 5.7 shows the end-to-end delay in *msec* vs. the OBUs density, by employing authentication using the proposed EMAP (SHA-1), the linear CRL checking, and binary CRL checking, respectively. In the simulation, we consider CRLs containing 20000 and 30000 revoked certificates, respectively, and the OBUs density as the number of OBUs per km^2. It can be seen that the end-to-end delay increases with the OBUs density because the number of the received packets increases with the OBUs density resulting in longer waiting time for the packets to be processed by the application layer in each OBU. In addition, the end-to-end delay tends to be constant for high OBUs densities as the number of received packets reaches the maximum number of packets an OBU can verify within a specific duration. The end-to-end delay also increases with the number of revoked certificates included in the CRL for the linear CRL checking process. However, the end-to-end delay is almost constant with the CRL size using the binary checking process as the number of comparison operations needed to check CRLs with 20000 and 30000 certificates is almost the same. From Fig. 5.7, employing the proposed EMAP in authentication reduces the end-to-end delay compared with that using either the linear or the binary CRL checking process.

5.8.4 Message Loss Ratio

The average message loss ratio is defined as the average ratio between the number of messages dropped every 300 *msec*, due to the message authentication delay, and the total number of messages received every 300 *msec* by an OBU. It should be noted that we are only interested in the message loss incurred by OBUs due to V2V communications. According to DSRC, each OBU has to disseminate a message containing information about the road condition every 300 *msec*. In order to react properly and instantly to the varying road

conditions, each OBU should verify the messages received during the last 300 $msec$ before disseminating a new message about the road condition. Therefore, we chose to measure the message loss ratio every 300 $msec$. Fig. 5.8 shows the analytical and simulated average message loss ratio vs. the average number of OBUs within the communication range of each OBU for message authentication employing CRL linear checking, CRL binary checking, and EMAP, respectively, for a CRL containing $20,000$ certificates. It can be seen that the simulated average message loss ratio closely follows the analytical message loss ratio which is calculated based on the maximum number of messages that can be authenticated within 300 $msec$. The difference between the analytical and simulations results stems from observing that some zones in the simulated area become more congested than other zones, thus, some OBUs experience higher message loss than other OBUs, which leads to that difference between the analytical and simulations results. It can also be seen that the message loss ratio increases with the number of OBUs within communication range for all the protocols under considerations. In addition, the message authentication employing EMAP significantly decreases the message loss ratio compared to that employing either the linear or binary CRL revocation status checking. The reason of the superiority of EMAP is that it incurs the minimum revocation status checking delay compared to the linear and binary CRL revocation checking processes.

5.8.5 Communication Overhead

In EMAP, each OBU_u broadcasts a signed message on the form $(\mathcal{M}||T_{stamp}||\ cert_u(PID_u, PK_u, sig_{TA}(PID_u||PK_u))||sig_u(\mathcal{M}||T_{stamp})||REV_{check})$ to its neighboring OBUs. A signed message in the WAVE standard should include the certificate of the sender, a time stamp, and the signature of the sender on the transmitted message. Consequently, the additional

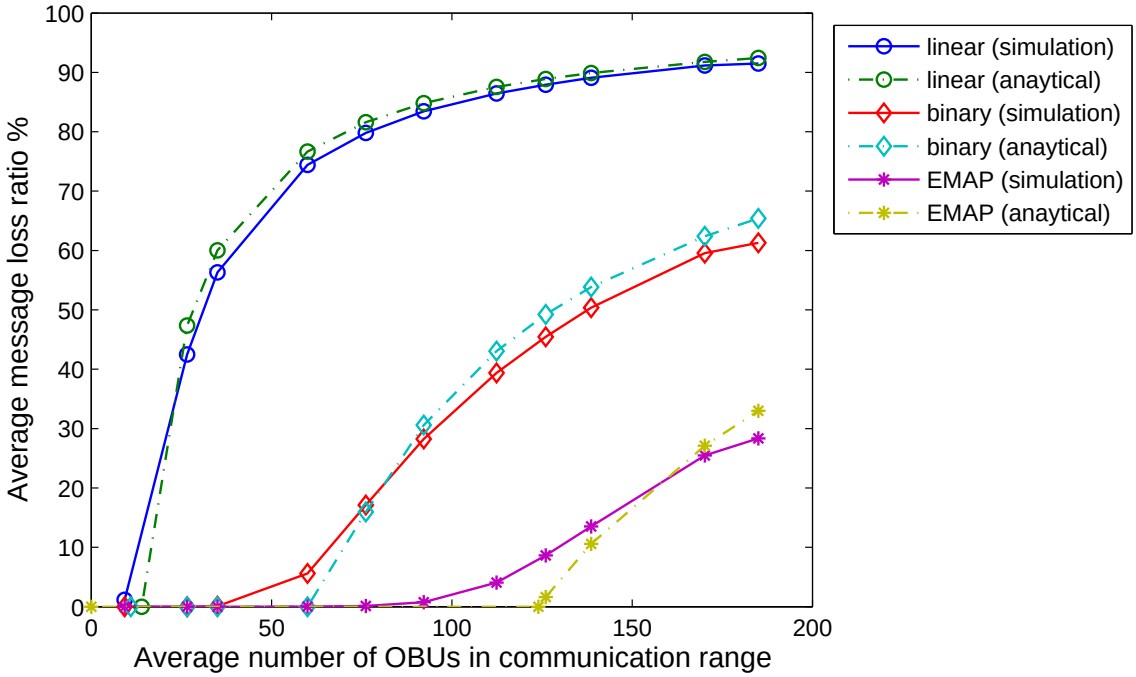


Figure 5.8: Comparison between message loss ratio for different schemes

communication overhead incurred in EMAP compared to that in the WAVE standard is mainly due to REV_{check}. The length of REV_{check} depends on the employed hash function. For example, when SHA-1 is employed in EMAP for calculating REV_{check}, this is corresponding to an additional overhead of 20 bytes [68]. The total overhead incurred in a signed message in the WAVE standard is 181 bytes [2]. Consequently, the total overhead in EMAP (SHA-1), assuming the same message format of the WAVE standard, is 201 bytes. In WAVE [2], the maximum payload data size in a signed message is 65.6 Kbytes. Accordingly, the ratio of the communication overhead in a signed message to the payload data size is 0.28% and 0.31% for the WAVE standard and EMAP, respectively. EMAP incurs 0.03% increase in the communication overhead compared to the WAVE standard,

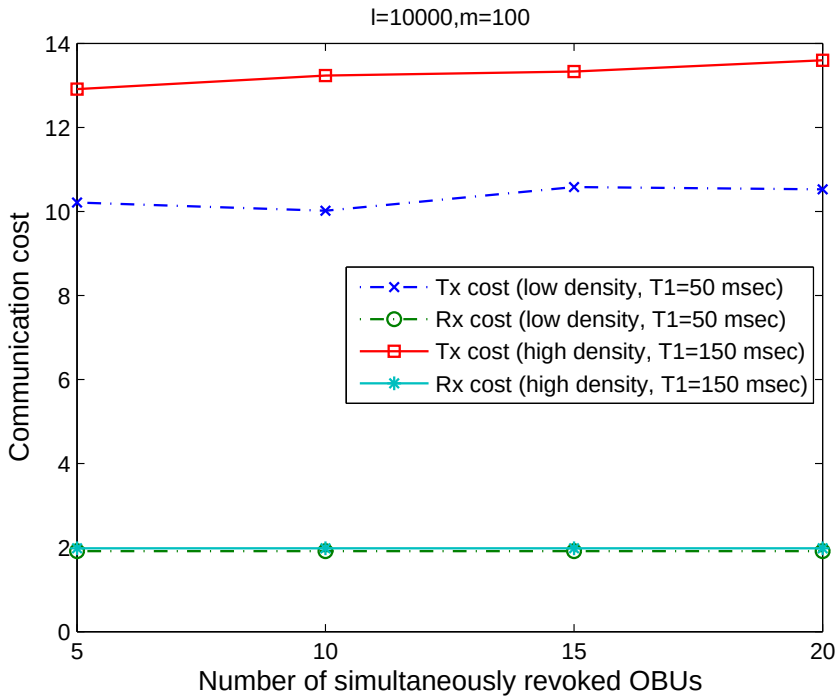


Figure 5.9: Communication cost of updating K_g

which is acceptable with respect to the gained benefits from EMAP.

5.8.6 Communication Cost of Updating the Secret Group Key (K_g)

We are interested in the communication cost of updating the secret group key (K_g), which is the average number of messages an OBU has to transmit and receive after triggering the revocation process to get the new secret group key ($\tilde{K_g}$) and distribute $\tilde{K_g}$ to its unrevoked neighboring OBUs. We have conducted ns-2 [47] simulation for the city street scenario shown in Fig. 5.6, for two scenarios: low and high OBUs densities corresponding to OBUs densities of 32.5 $/km^2$ and 91.5 $/km^2$, respectively. We use the same simulation parameters

given in Table 5.1. We select the dissemination of the road condition information by an OBU every 300 *msec* to conform with the DSRC standards. We consider the TA having a key pool of size $l = 10000$, and each OBU having a key set of size $m = 100$. In the proposed complemented PKI, an OBU_y not having K_M^- will send a request message to its neighboring OBUs to get the new secret group key ($\tilde{K_g}$) and start timer T_1, where OBU_y will retry to get $\tilde{K_g}$ if T_1 is expired before getting $\tilde{K_g}$. In the conducted simulation, we set T_1 to be 50 *msec* and 150 *msec* for the low and high OBUs densities, respectively. Also, we only consider the case that an OBU without K_M^-, can get the new secret group key $\tilde{K_g}$ from another OBU through a single hop.

Initially, the percentage of OBUs having the key K_M^- is 1.97% and 1.56% for the low and high OBUs densities, respectively. After the broadcast of the revocation message REV_{msg}, only the OBUs having K_M^- are able to independently calculate the new secret group key $\tilde{K_g}$, and they will deliver $\tilde{K_g}$ to other OBUs through V2V communication. Fig. 5.9 shows the average communications cost vs. the number of simultaneously revoked OBUs. It can be seen that in each scenario, the communication cost (transmit or receive) is almost constant with respect to the number of simultaneously revoked OBUs. This is due to the fact that revoking the key sets of the revoked OBUs does not revoke the key K_M^- which is shared by the majority of OBUs. Consequently, the percentage of OBUs initially having K_M^- will not change. It can be seen that the communication cost is equal with respect to the number of received messages in both low and high OBUs densities. Also, the communication cost of the transmitted messages is higher than that of the received messages. This is due to the fact that a request broadcast by an OBU_y to get the new secret group key ($\tilde{K_g}$) is received by all the neighboring OBUs, and each OBU of the neighboring OBUs will send $\tilde{K_g}$ to OBU_y. As a result, a number of OBUs, requesting $\tilde{K_g}$, in some geographic area, will cause all the neighboring OBUs to broadcast $\tilde{K_g}$ as many times as the number of

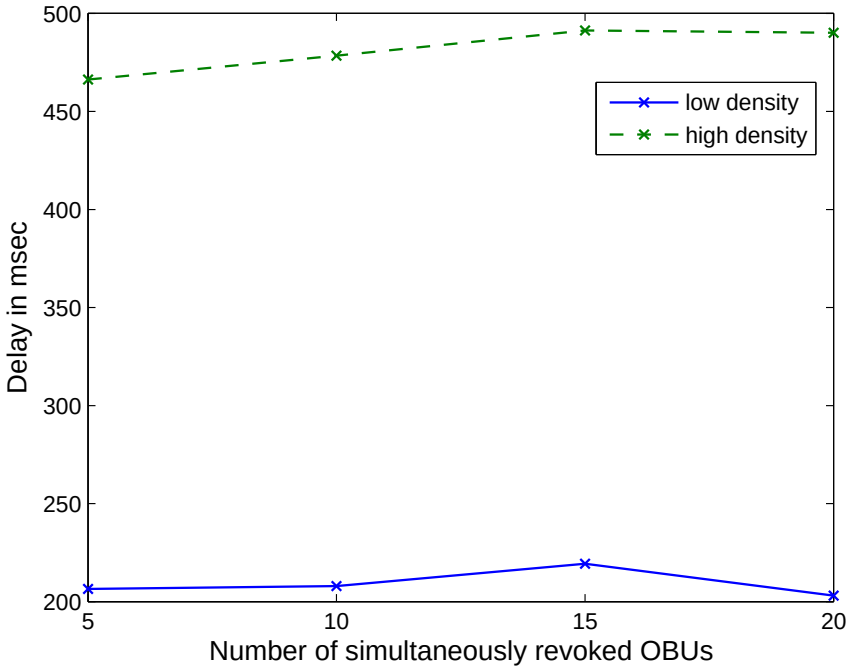


Figure 5.10: Incurred delay to obtain $\tilde{K_g}$

OBUs requesting $\tilde{K_g}$ in that area. We have tried several values for the timer T_1, and the considered values for T_1 give the best results. If we select smaller T_1, the transmission cost will increase since each OBU not having K_M^- will send requests to get $\tilde{K_g}$ at a higher rate, and hence, more replies will be transmitted by the OBUs. It can also be seen that the low OBU density scenario incurs lower transmission communication cost than the high OBU density scenario since the number of OBUs in the low density scenario is lower than that in the high OBU density scenario.

Table 5.2: Simulation parameters for REP

Simulation parameter	Highway model	Manhattan model
road length (km)	3	3x3
arrival rate (/msec)	0.01	0.012
OBUs speed(km/h)	80	50
simulation time (min)	4	6

5.8.7 Incurred Delay to Obtain the New Secret Group Key ($\tilde{K_g}$)

We are interested in the average delay for an OBU without K_M^- to get the new secret group key $\tilde{K_g}$ from its neighboring OBUs after the revocation message REV_{msg} is delivered to all the OBUs in the simulated area. We conducted ns-2 simulation for the low and high OBUs densities scenarios considered in the previous subsection. Initially, the percentage of OBUs having the key K_M^-, and capable of independently calculating $\tilde{K_g}$, is 1.97% and 1.56% for the low and high OBUs densities scenarios, respectively. Fig. 5.10 shows the average delay in *msec*, incurred by an OBU from the moment the revocation message REV_{msg} is received by all the OBUs in the simulated area until it gets the new secret group key $\tilde{K_g}$, vs. the number of simultaneously revoked OBUs. It can be seen that the incurred delay to get $\tilde{K_g}$ is confined to a small range in each scenario. Also, the delay of obtaining $\tilde{K_g}$ in the high OBU density scenario is higher than that in the low OBU density scenario as the value of T_1 in the high OBU density scenario is higher than that in the low OBU density scenario. However, for both low and high OBU densities, the delay of getting $\tilde{K_g}$ is less than 1 *sec*, which indicates that the proposed complemented PKI is feasible and reliable.

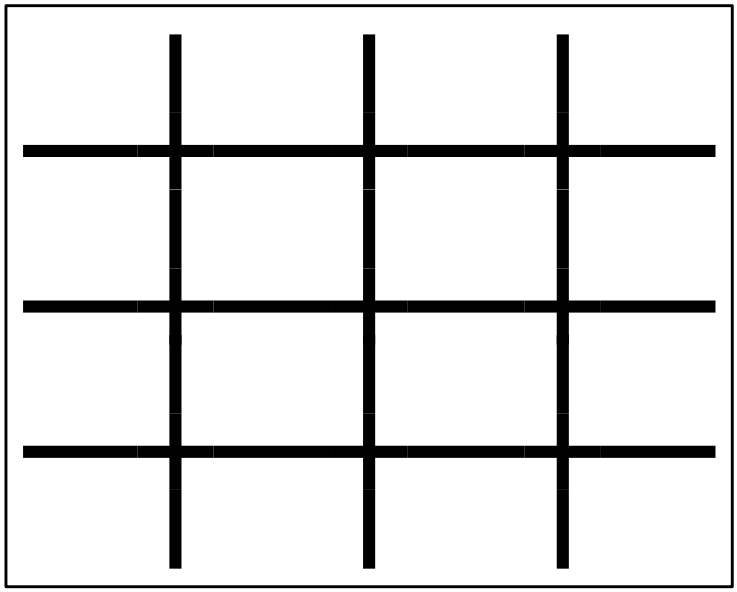


Figure 5.11: Manhattan road model

5.8.8 Anonymity Set Size

In this section, we evaluate the achieved anonymity set size under REP. We simulate a highway and Manhattan mobility models using Matlab. The simulation parameters are given in Table. 5.2. In each mobility model, OBUs arrive according to a Poisson random process, where the arriving OBU randomly selects an entry port from predefined entry ports. When an OBU reaches one of the output entries, it disappears. The highway model consists of 6 lanes (3 in each direction). For the Manhattan mobility model shown in Fig. 5.11, each street consists of 4 lanes (2 in each direction). At each intersection, each OBU has a probability of 25% to turn right or left.

We use the highway and Manhattan mobility models to investigate the achieved average anonymity set size using REP. Each OBU has a probability of 0.05% to change its certificate every 300 *msec*. Also, we consider that the criteria to terminate the encryption period is

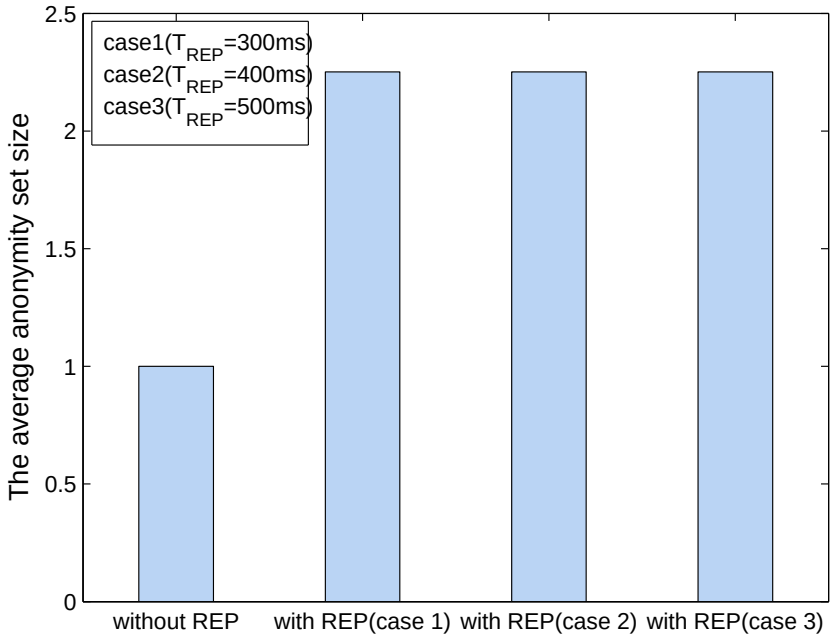


Figure 5.12: The average anonymity set size for highway mobility model

that the anonymity set size is greater than one. Figs. 5.12-5.17 show the average anonymity set size, the average REP duration, and the ratio between the number of anonymity sets of different sizes to the total number of anonymity sets for the highway and Manhattan mobility models, respectively. In Figs. 5.12-5.13, we compare the average anonymity set size with and without REP. Also, we simulate REP for three cases corresponding to initial T_{REP} values of 300 $msec$, 400 $msec$, and 500 $msec$, respectively. It can be seen that without using REP (which is the current normal mode of VANETs), the average anonymity set size is one since the global observer can capture all the broadcast messages. As a result, the location privacy of drivers is vulnerable to the tracking attack previously discussed. In addition, it can be seen that with using REP, the average anonymity set size is always greater than two, which decreases the probability of being tracked by a global observer.

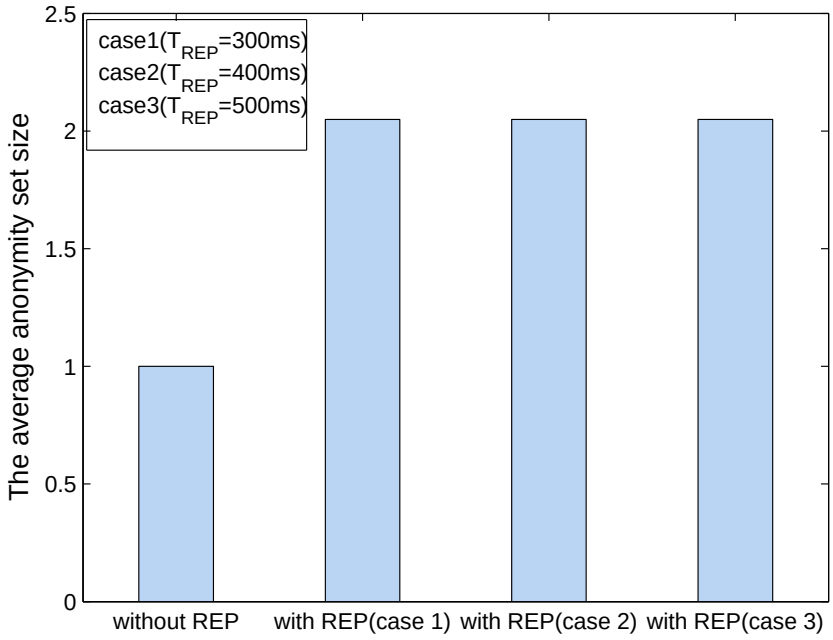


Figure 5.13: The average anonymity set size for Manhattan mobility model

It can be seen from Figs. 5.14-5.15 that the average REP duration for the Manhattan model is higher than that for the highway model as the OBUs density in the Manhattan model is less than that for the highway model. The reason can be explained as follows. First, although the arrival rate for the Manhattan model is higher than that for the highway model, the arriving OBUs are distributed between 6 streets (4 lanes each, i.e., 24 lanes) for the Manhattan model, while the arriving OBUs are distributed between 6 lanes in the highway model. Second, the simulation area for the Manhattan model is larger than that for the highway model. From the aforementioned explanation, the OBUs density in the Manhattan model is lower than that in the highway model. Hence, the OBUs in the Manhattan model needs more time to have anonymity set size greater than one.

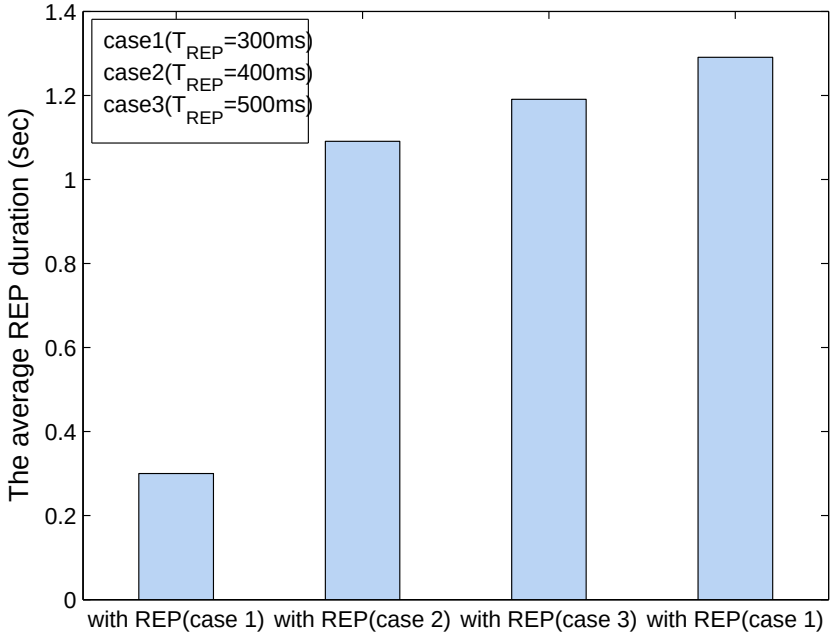


Figure 5.14: The average REP duration for highway mobility model

Figs. 5.16-5.17 show the ratio between the anonymity sets of different sizes and the total number of anonymity sets. For both Manhattan model and highway model, it can be seen that without using REP, the anonymity sets of size one are 100% of the total anonymity sets. In addition, when using REP in the highway model, the ratio of the anonymity sets of size greater than one are 100%. For the Manhattan model, when using REP, 98.24% of the total anonymity sets achieves set size greater than one, while 1.76% of the anonymity sets are still having size one by the end of the simulation time.

Since the random silent periods proposed by Sampigethaya et al. [7] only considers location privacy for non-safety applications in vehicular networks, and REP is mainly used for providing location privacy for safety applications, the only work that can be compared

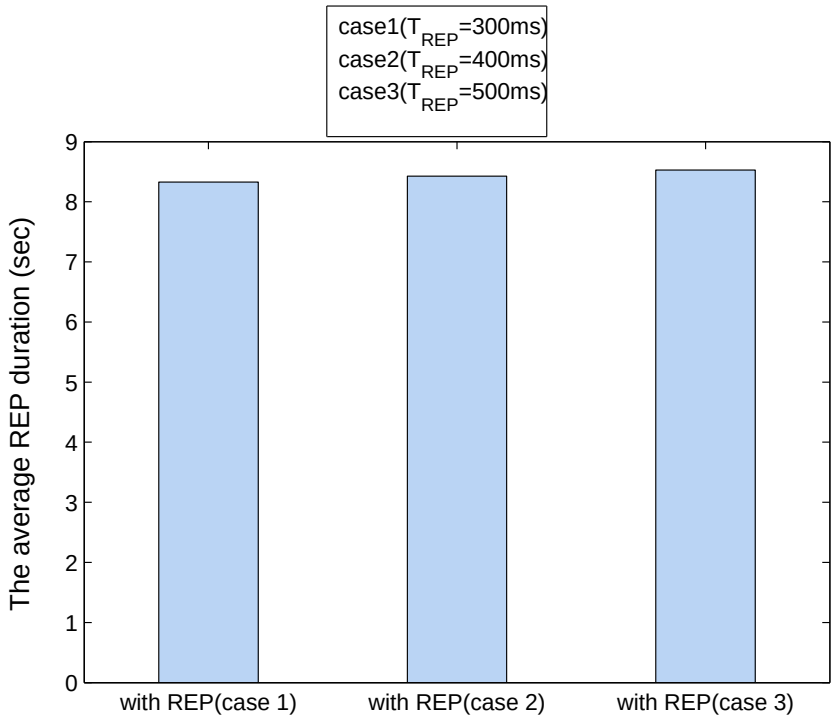


Figure 5.15: The average REP duration for Manhattan mobility model

to REP is the Cryptographic MIX-zones (CMIXes) approach [33]. In addition, the authors of CMIXes demonstrate that CMIXes can be applied in a city scenario with the help of RSUs at some selected intersection, however, they do not explain how CMIXes can be applied in a highway scenario. Hence, we only consider CMIXes in the Manhattan model. In the conducted simulation for CMIXes, we consider that there is an RSU with a coverage area of 300 m [3] at each intersection. In addition, when an OBU_i enters the coverage area of an RSU_j at an intersection, it is forced to change its certificate according to [33]. Moreover, we consider all the OBUs within the coverage area of RSU_j to be in the anonymity set of OBU_i. When OBU_i leaves the coverage area of RSU_j, the anonymity set corresponding to OBU_i is terminated, and a new anonymity set is generated at RSU_j for the first OBU enters the coverage area of RSU_j. Fig. 5.17 shows a comparison of

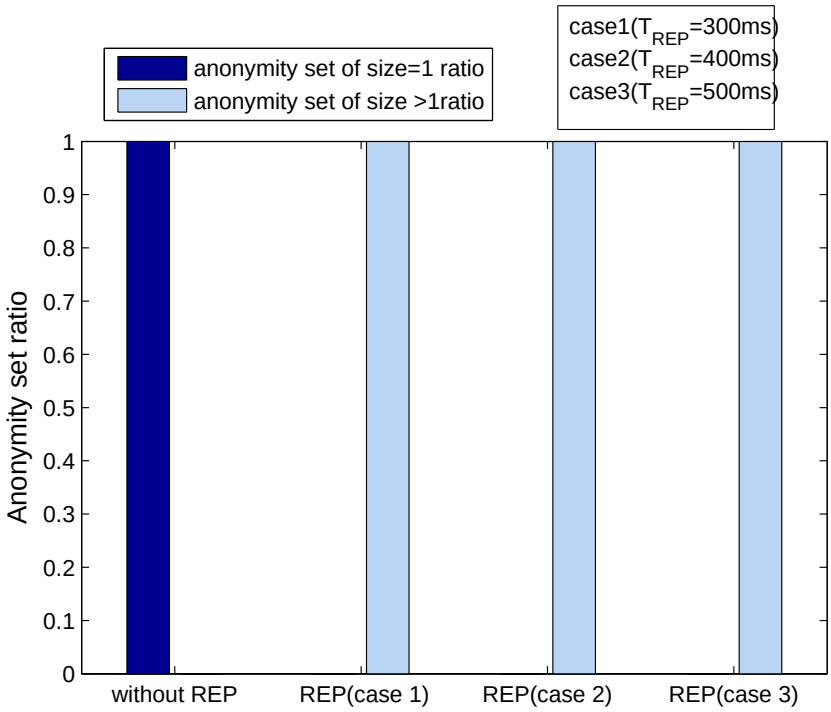


Figure 5.16: The impact of REP on the anonymity set size for highway mobility model

the ratio between the anonymity sets of different sizes and the total number of anonymity sets in the Manhattan model for the case without and with REP, and with CMIXes. It can be seen that for CMIXes, only 25.42% of the total anonymity sets achieves set size greater than one, while 74.58% of the anonymity sets are still having size one by the end of the simulation time, which clearly indicates that the proposed REP protocol outperforms CMIXes. Another advantage of the REP scheme is that it does not rely on the existence of RSUs compared to CMIXes. Therefore, REP is more suitable for the early deployment stage of VANETs where the RSUs will be sparsely distributed.

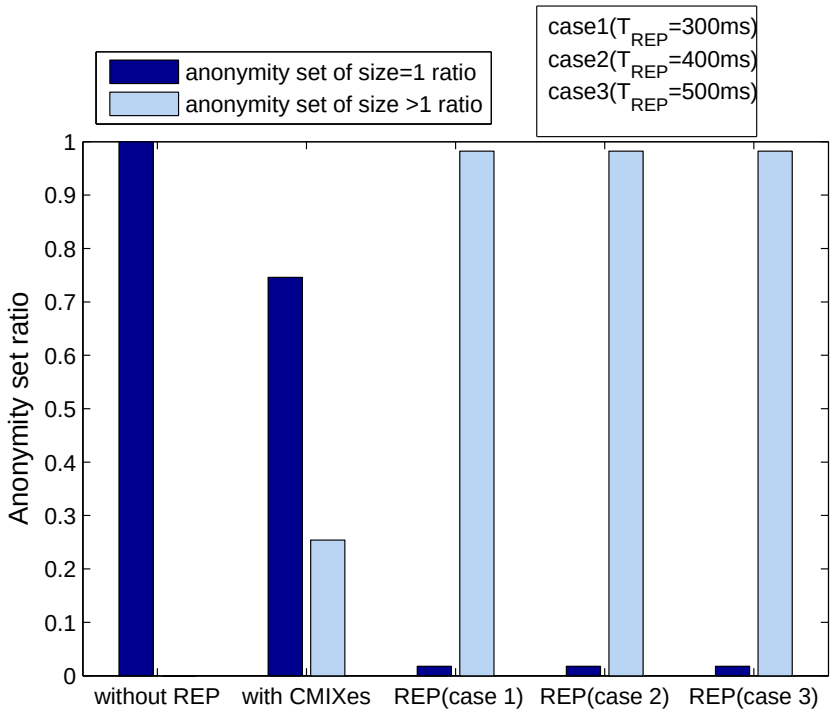


Figure 5.17: The impact of REP on the anonymity set size for Manhattan mobility model

5.9 Complementing DCS with EMAP and REP

The DCS scheme presented in Chapter 3 can be complemented with EMAP and REP as follows.

- During the system initialization in the DCS scheme, the MA generates a secret key pool, calculates the corresponding public keys as indicated in subsection 5.3.2, and selects a secret group key K_g. It should be noted that the key pool for complementing DCS is different from that required to integrate EDR with DCS, i.e., each CA will have two key pools: one for EDR, and the other one for complementing DCS with EMAP and REP.

- The MA copies the key pool, the corresponding public keys, and the secret group key K_g in each CA.
- During the initialization of the OBUs, each OBU randomly selects m secret keys, their corresponding public keys from the key pool of its in charge CA. Also, the CA loads the secret group key K_g in each OBU.
- Whenever an OBU or RSU wants to sign a message, it generates the signature on the message as indicated in subsection 3.5.1, and then calculates the revocation check as indicated in subsection 5.4.1.
- Whenever an OBU wants to change its current certificate, it employs REP as indicated in subsection 5.5.1 to preserve its location privacy.
- When a CA initiates a rekeying process, it proceeds as indicated in Algorithm 5.6, and sends the revocation version *ver* to the MA. The MA, broadcasts *ver* to all the CAs such that when a rekeying process is performed in another CA, this CA should consider $ver + 1$ as its revocation version. Since the key pool is the same in each CA and the future revocation version is always incremental compared to the current revocation version experienced by the OBUs, the OBUs can roam between different domains and still be able to use EMAP, REP, and update its randomly selected keys.

5.10 Summary

In this Chapter, we have proposed complemented PKI, which employs a probabilistic key distribution to establish a shared secret group key between the group members. The

proposed complemented PKI expedites message authentication by replacing the time-consuming CRL checking process with a fast revocation checking process employing *HMAC* function. The expedite message authentication (EMAP) can significantly decrease the message loss ratio due to message verification delay compared to the conventional authentication methods employing CRL checking. In addition, the proposed complemented PKI provides location privacy for VANETs using random encryption periods (REP). For an OBU changing its certificate, REP triggers encryption zone around the OBU to violate the conditions required to launch a tracking attack, and it creates an ambiguity to any external observer. Extensive analysis and evaluation of the proposed complemented PKI have been performed to demonstrate its reliability and security. Moreover, the proposed complementing mechanisms (EMAP and REP) have modular natures which enable their integration with any PKI, e.g., the DCS scheme presented in Chapter 3.

Chapter 6

Conclusions and Future Work

In this Chapter, we present our conclusions and outline our future research directions.

6.1 Conclusions

In this thesis, we have proposed a set of protocols to provide security and privacy preservation for VANETs. Our results have shown that the proposed protocols can achieve the targeted security and privacy preservation requirements. In addition, the detailed performance evaluation and security analysis have indicated that the proposed protocols are secure and efficient. The achievements accomplished in this thesis can be summarized as follows.

- In Chapter 2, we surveyed the state of the art in security and privacy preservation in VANETs. We have identified the challenges for securing VANETs by discussing security threats and requirements in VANETs. Moreover, we have detailed the necessary security mechanisms to achieve anonymous authentication and location privacy

for preserving the users' privacy in VANETs. In addition, we have described the different revocation mechanisms in VANETs.

- In Chapter 3, we have proposed an efficient distributed certificate service (DCS) scheme for vehicular communications, which offers a flexible interoperability to avoid the key escrow issue in different administrative authorities and an efficient distributed algorithm for any OBU to update or revoke its certificate from the available RSUs in a timely manner. In addition, with the batch verification, the entities in the DCS scheme can rapidly verify a mass of message signatures and certificates simultaneously. Therefore, the proposed DCS scheme can significantly reduce the complexity of certificate management, and achieve excellent efficiency and scalability, especially when it is deployed in heterogeneous vehicular networks.

- In Chapter 4, we have proposed a robust and efficient decentralized revocation (EDR) protocol for VANETs, which substantially reduces the complexity of the certificate revocation problem, while achieving fast revocation of the misbehaving vehicles. The EDR protocol decreases the vulnerability window that a misbehaving vehicle has resulting in higher safety level for VANET. The EDR protocol is resistant to the most known revocation attacks. In addition, it can be efficiently integrated with any PKI and/or any misbehavior detection scheme for VANETs.

- In Chapter 5, we have proposed complemented PKI, which employs a probabilistic key distribution to establish a shared secret group key between the group members. The proposed complemented PKI expedites message authentication by replacing the time-consuming CRL checking process with a fast revocation checking process employing *HMAC* function. The expedite message authentication (EMAP) can significantly decrease the message loss ratio due to message verification delay compared

to the conventional authentication methods employing CRL checking. In addition, the proposed complemented PKI provides location privacy for VANETs using random encryption periods (REP). For an OBU changing its certificate, REP triggers encryption zone around the OBU to violate the conditions required to launch a tracking attack, and it creates an ambiguity to any external observer. Extensive analysis and evaluation of the proposed complemented PKI have been performed to demonstrate its reliability and security. Moreover, the proposed complementing mechanisms (EMAP and REP) have modular natures which enable their integration with any PKI, e.g., the DCS scheme presented in Chapter 3.

6.2 Future Work

In this thesis, we have proposed a set of security protocols that can achieve reliable and secure vehicular communications. Although the achievements accomplished in this thesis, the research done in the thesis can only provide the security requirements necessary for deploying the basic VANET applications and there still many open research areas that should be tackled. In the following subsections, we indicate our future work.

6.2.1 Reputation-Based Scheme for VANETs

In decentralized reputation-based schemes, each node assigns a reputation value for each neighboring node based on the received messages from the other nodes. The applications of reputation-based schemes in VANETs are very promising [70]. For example, a reputation-based scheme can be used to complement the secure routing in VANETs to help a vehicle in selecting the most reliable forwarder available among its neighboring vehicles. Also, a

reputation-based scheme can be used to check the consistency of the received data. In safety applications, e.g., lane change assistance, if the consistency of the received message cannot be guaranteed, the vehicles receiving this message can be subject to fatal actions. It is expected that anonymous authentication will be employed in VANETs [2], where vehicle periodically changes its identity to preserve its privacy. Consequently, deploying a reputation-based scheme in VANETs will be very challenging. We believe that the main challenge is how to link the reputation of a vehicle to its periodically changing identities. In the future, we will propose a reputation-based scheme suitable for VANETs. In addition, we are going to explore the different VANET applications that reputation-based schemes can be used in to enhance their security.

6.2.2 Privacy-Preservation in Position-Based Routing in VANETs

Position-based routing schemes requires the vehicles to periodically reveal their locations in order to properly route the messages between the source and the destination. This requirement violates the location privacy of the users and it may be the cause that users abstain from using the multi-hop applications in VANETs. Preserving the location privacy in position-based routing protocols is an open research topic. The main challenge in this research topic is how to route the messages between the source and the destination without revealing their exact locations. In our future work, we are going to investigate how to solve this problem.

6.2.3 Integrating Electronic Health With Vehicular Networks

In emergency situation such as accidents, health records are needed on the spot to provide the necessary medical care, which may be essential to save lives. Through personal health

records, electronic health (ehealth) can provide this medical service on the roads [71]. On the other hand, body area network is a set of sensors mounted on the body of the patient to monitor his/her health conditions. To enable the aforementioned medical service on the road, ehealth and body area network should be integrated with location based routing in VANET. Such integration is very important for providing health reports in case of accidents. Also, vehicles can display warning messages for the drivers in case their blood pressure, blood sugar level, etc., reach a dangerous level. In our future work, we need to investigate this open research area.

References

[1] "Statistics Canada." [Online]. Available: http://www.statcan.ca/english/freepub/50-501-XIE/sect3.pdf

[2] "IEEE trial-use standard for wireless access in vehicular environments - security services for applications and management messages," *IEEE Std 1609.2-2006*, 2006.

[3] "Dedicated short range communications (DSRC)." [Online]. Available: http://www.leearmstrong.com/DSRC/DSRCHomeset.htm

[4] M. Raya and J.-P. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.

[5] F. Li and Y. Wang, "Routing in vehicular ad hoc networks: A survey," *IEEE Vehicular Technology Magazine*, vol. 2, no. 2, pp. 12–22, 2007.

[6] P. Papadimitratos, A. Kung, J. P. Hubaux, and F. Kargl, "Privacy and identity management for vehicular communication systems: a position paper," *Proc. Workshop on Standards for Privacy in User-Centric Identity Management, Zurich, Switzerland*, July 2006.

[7] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "CARAVAN: Providing location privacy for VANET," *Proc. Embedded Security in Cars (ESCAR)*, November 2005.

[8] J. P. Hubaux, "The security and privacy of smart vehicles," *IEEE Security and Privacy*, vol. 2, pp. 49–55, 2004.

[9] M. Shi, X. Shen, and J. Mark, "IEEE 802.11 roaming and authentication in wireless LAN/cellular mobile networks," *IEEE Wireless Communications*, vol. 11, no. 4, pp. 66–75, 2004.

[10] Y. Jiang, C. Lin, X. Shen, and M. Shi, "Mutual authentication and key exchange protocols for roaming services in wireless mobile networks," *IEEE Transactions on Wireless Communications*, vol. 5, no. 9, pp. 2569–2577, 2006.

[11] K. P. Laberteaux, J. J. Haas, and Y. Hu, "Security certificate revocation list distribution for VANET," *Proc. 5th ACM international workshop on VehiculAr Inter-NETworking*, pp. 88–89, 2008.

[12] P. P. Papadimitratos, G. Mezzour, and J. Hubaux, "Certificate revocation list distribution in vehicular communication systems," *Proc. 5th ACM international workshop on VehiculAr Inter-NETworking*, pp. 86–87, 2008.

[13] M. Raya and J. Hubaux, "Security aspects of inter-vehicle communications," *Swiss Transport Research Conference (STRC)*, 2005.

[14] C. Laurendeau and M. Barbeau, "Threats to security in DSRC/WAVE," in *Ad-Hoc, Mobile, and Wireless Networks*, 2006, pp. 266–279.

[15] M. Raya, P. Papadimitratos, and J. P. Hubaux, "Securing vehicular communications," *IEEE WIRELESS COMMUNICATIONS MAGAZINE*, vol. 13, pp. 8—15, 2006.

[16] B. Parno and A. Perrig, "Challenges in securing vehicular networks," *Proc. HotNets-IV*, 2005.

[17] K. Plossl, T. Nowey, and C. Mletzko, "Towards a security architecture for vehicular ad hoc networks," *Proc. ARES 2006.*, pp. 374–381, 2006.

[18] P. Papadimitratos, V. Gligor, and J. P. Hubaux, "Securing vehicular communications - assumptions, requirements, and principles," *IN WORKSHOP ON EMBEDDED SECURITY IN CARS (ESCAR)*, 2006.

[19] D. Chaum and E. van Heyst, "Group signatures," *Proc. Eurocrypt*, vol. 547, pp. 257–265, 1991.

[20] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," *Proc. Crypto, LNCS*, vol. 3152, pp. 41–55, 2004.

[21] X. Lin, X. Sun, P.-H. Ho, and X. Shen, "GSIS: A secure and privacy-preserving protocol for vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 56, pp. 3442–3456, 2007.

[22] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing," *Proc. 21st Annual International Cryptology Conference on Advances in Cryptology*, pp. 213–229, 2001.

[23] J. Guo, J. Baugh, and S. Wang, "A group signature based secure and privacy-preserving vehicular communication framework," *Proc. Mobile Networking for Vehicular Environments*, pp. 103–108, 2007.

[24] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, "ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications," *Proc. INFOCOM 2008*, pp. 1229–1237, 2008.

[25] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," *Proc. 4th ACM international workshop on Vehicular ad hoc networks*, pp. 19–28, 2007.

[26] C. Zhang, R. Lu, X. Lin, P.-H. Ho, and X. Shen, "An efficient identity-based batch verification scheme for vehicular sensor networks," *Proc. IEEE INFOCOM 2008*, pp. 246–250, 2008.

[27] R. Anderson and M. Kuhn, "Tamper resistance - a cautionary note," *Proc. 2nd Usenix Workshop on Electronic Commerce*, pp. 1–11, 1996.

[28] C. Zhang, X. Lin, R. Lu, P. Ho, and X. Shen, "An efficient message authentication scheme for vehicular communications," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 6, pp. 3357–3368, 2008.

[29] A. Pfitzmann and M. KÄhntopp, "Anonymity, unobservability, and pseudonymity- a proposal for terminology," *Designing Privacy Enhancing Technologies*, pp. 1–9, 2001.

[30] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," *Proc. 3rd ACM workshop on Security of ad hoc and sensor networks*, pp. 11–21, 2005.

[31] E. Schoch, F. Kargl, T. Leinmüller, S. Schlott, and P. Papadimitratos, "Impact of pseudonym changes on geographic routing in VANETs," *Proc. 3rd European Workshop on Security and Privacy in Ad-Hoc and Sensor Networks (ESAS)*, pp. 43–57, 2006.

[32] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "AMOEBA: robust location privacy scheme for VANET," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1569–1589, 2007.

[33] J. Freudiger and M. Raya, "Mix-zones for location privacy in vehicular networks," *Proc. WiN-ITS*, August 2007.

[34] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, and J. P. Hubaux, "Certificate revocation in vehicular networks," *Technical Report LCA-Report-2006-006*, 2006.

[35] 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*, vol. 25, pp. 1557–1568, 2007.

[36] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[37] M. Scott, "Computing the Tate pairing," *Topics in Cryptology, Springer*, pp. 293–304, 2005.

[38] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-Reductions." *IEIC Technical Report (Institute of Electronics, Information and Communication Engineers)*, vol. 100, no. 323(ISEC2000 58-67), pp. 99–108, 2000.

[39] "Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL)." [Online]. Available: http://www.shamus.ie/

[40] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 19, no. 2, pp. 173–193, Mar. 2000.

[41] D. Boneh and R. Lipton, "Algorithms for black-box fields and their application to cryptography," *Proc. Advances in Cryptology - CRYPTO '96*, pp. 283–297, 1996.

[42] S. Al-Riyami and K. Paterson, "Certificateless public key cryptography," *Proc. Advances in Cryptology - ASIACRYPT 2003*, pp. 452–473, 2003.

[43] X. Huang, W. Susilo, Y. Mu, and F. Zhang, "On the security of certificateless signature schemes from asiacrypt 2003," *Proc. 4th International Conference on Cryptology and Network Security (CANS05), LNCS*, vol. 3810, Springer Verlag, pp. 13–25, 2005.

[44] Y. Jiang, M. Shi, X. Shen, and C. Lin, "BAT: a robust signature scheme for vehicular networks using binary authentication tree," *IEEE Transactions on Wireless Communications*, vol. 8, no. 4, pp. 1974–1983, 2009.

[45] R. Oppliger, "Protecting key exchange and management protocols against resource clogging attacks," *Proc. Joint Working Conference on Secure Information Networks: Communications and Multimedia Security*, pp. 163–175, 1999.

[46] A. Wasef and X. Shen, "REP: location privacy for VANETs using random encryption periods," *ACM Mobile Networks and Applications (MONET)*, vol. 15, no. 1, pp. 172–185, 2010.

[47] "The network simulator - ns-2." [Online]. Available: http://nsnam.isi.edu/nsnam/index.php/UserInformation

[48] "Traffic and network simulation environment - TraNS." [Online]. Available: http://trans.epfl.ch/

[49] D. Cottingham, I. Wassell, and R. Harle, "Performance of IEEE 802.11a in vehicular contexts," *IEEE 65th Vehicular Technology Conference*, pp. 854–858, 2007.

[50] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, "Performance evaluation of safety applications over DSRC vehicular ad hoc networks," *Proc. 1st ACM international workshop on Vehicular ad hoc networks*, pp. 1–9, 2004.

[51] D. Johnson, A. Menezes, and S. Vanstone, "The elliptic curve digital signature algorithm (ECDSA)," *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.

[52] K. Sadasivam and T. Yang, "Evaluation of certificate-based authentication in mobile ad hoc networks," *Proc. IASTED NCS 2005*, April 2005.

[53] "List of U. S. states by area." [Online]. Available: http://www.knowledgerush.com/kr/encyclopedia/List_of_U.S._states_by_area/

[54] "United states department of transportation - federal highway administration." [Online]. Available: http://www.fhwa.dot.gov/policyinformation/statistics/2007/hm20m.cfm

[55] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and verifiably encrypted signatures from bilinear maps," *Proc. Advances in Cryptology. EUROCRYPT 2003*, pp. 416–432, 2003.

[56] Z. Gong, Y. Long, X. Hong, and K. Chen, "Two certificateless aggregate signatures from bilinear maps," *Proc. 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. SNPD 2007*, vol. 3, pp. 188–193, 2007.

[57] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, "A novel anonymous mutual authentication protocol with provable link-ayer location privacy," *IEEE Transactions on Vehicular Technology,*, vol. 58, no. 3, pp. 1454–1466, 2009.

[58] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[59] S. Zhu, S. Setia, S. Xu, and S. Jajodia, "GKMPAN: An efficient group rekeying scheme for secure multicast in ad-hoc networks," *Journal of Computer Security*, vol. 14, pp. 301–325, 2006.

[60] "US bureau of transit statistics." [Online]. Available: http://en.wikipedia.org/wiki/Passenger_vehicles_in_the_United_States

[61] J. J. Haas, Y. Hu, and K. P. Laberteaux, "Design and analysis of a lightweight certificate revocation mechanism for VANET," *Proc. 6th ACM international workshop on VehiculAr InterNETworking*, pp. 89–98, 2009.

[62] F. Dötzer, "Privacy issues in vehicular ad hoc networks," *Proc. 2nd ACM Workshop on Vehicular Ad Hoc Networks*, September 2006.

[63] J. Y. Choi, M. Jakobsson, and S. Wetzel, "Balancing auditability and privacy in vehicular networks," *Proc. 1st ACM international workshop on Quality of service and security in wireless and mobile networks*, pp. 79–87, 2005.

[64] K. Sha, Y. Xi, W. Shi, L. Schwiebert, and T. Zhang, "Adaptive privacy-preserving authentication in vehicular networks," *Proc. ChinaCom '06*, pp. 1–8, 2006.

[65] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[66] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT Press, 2001.

[67] S. Frankel, R. Glenn, and S. Kelly, "The AES-CBC cipher algorithm and its use with IPsec," *RFC3602*, Sept. 2003.

[68] D. Eastlake and P. Jones, "US secure hash algorithm 1 (SHA1)," *RFC 3174*, Sept. 2001.

[69] "Crypto++ library 5.5.2." [Online]. Available: http://www.cryptopp.com/

[70] N. Lo and H. Tsai, "A reputation system for traffic safety event on vehicular ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, 2009.

[71] G. Yan, Y. Wang, M. C. Weigle, S. Olariu, and K. Ibrahim, "WEHealth: A secure and privacy preserving eHealth using NOTICE," *Proc. Intern. Conference on Wireless Access in Vehicular Environments (WAVE)*, Dec 2008.