A Secure Gateway Localization and Communication System for Vehicular Ad Hoc Networks

By

Yan Wang

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

In Partial Fulfillment of the Requirements

for the M.A.Sc Degree in

Electrical and Computer Engineering

Ottawa-Carleton Institution for Electrical and Computer Engineering

School of Electrical Engineering and Computer Science

Faculty of Engineering

University of Ottawa

Ottawa, Ontario, Canada

The following publication by the author is relevant to this thesis:

Conference

K. Abrougui, A. Boukerche, and Y. Wang. “Secure Gateway localization and communication system for Vehicular Ad Hoc Networks”, Accepted to IEEE Global Communications Conference.

Abstract

Intelligent Transport System (ITS) has become a hot research topic over the past decades. ITS is a system that applies the following technologies to the whole transportation management system efficiently, including information technique, wireless communication, sensor networks, control technique, and computer engineering. ITS provides an accurate, real time and synthetically efficient transportation management system. Obviously, Vehicular Ad Hoc NETworks (VANETs) attract growing attention from both the research community and industry all over the world. This is because a large amount of applications are enabled by VANETs, such as safety related applications, traffic management, commercial applications and general applications. When connecting to the internet or communicating with different networks in order to access a variety of services using VANETs, drivers and passengers in different cars need to be able to exchange messages with gateways from their vehicles. A secure gateway discovery process is therefore critical, because vehicles should not be subject to security attacks while they are communicating; however, currently there is no existing protocol focusing on secure gateway discovery.

In this thesis, we first analyze and compare current existing secure service discovery protocols and then we propose a Secure Gateway Localization and Communication System for Vehicular Ad Hoc Networks (SEGAL), which concentrates on the security issue in gateway discovery. We focus on the authentication aspect by proposing secure cluster based VANETs, that can ensure the gateway discovery messages exchanged through secure clusters. We present the principle and specific process of our SEGAL protocol and analyze its performance to guarantee its outstanding practical applicability.

Acknowledgements

Foremost, I would like to offer my deep and sincere gratitude to my supervisor Professor Dr. Azzedine Boukerche for his patience and immense knowledge in addition to the motivation and financial support he provided me with. The guidance and suggestions he gave me helped me to avoid and overcome several failures, so that I could smoothly complete my master's thesis. I could not imagine having a better supervisor and mentor for my graduate studies.

In addition, I would like to give my special thanks to Dr. Abrougui Kaouther for her interest, attention and patience. Her guidance helped me throughout my research, without which this thesis would have taken a longer time. Dr. Kaouther is a great friend of mine. Her warmth and guidance kept me motivated and strong in a foreign country.

I would like to thank Dr. Richard Pazzi and Dr. Robson De Grande, the managers of PARADISE Research Laboratory at the University of Ottawa, for their continuous help during this work.

I would like to express my gratitude to my fellow lab mates in the PARADISE Research Laboratory: Rui Liu, David Zhao, Yunfeng Gu, and Xin Fei for their companionship and help.

Last but not least, I would like to give my thanks to my family: my parents Guojuan Wang and Fengyun Liu for giving birth to me and encouraging and supporting me throughout my life.

Contents

List of Figures

List of Tables

Glossary of Terms

ASs	Application Servers
DoS	Denial-of-Service
DSRC	Dedicated Short Range Communication
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
ELP	Electronic License Plate
GID	Group Identification Certificate
GSA	Group-based Secure Source Authentication protocol
HAggt	Hash Aggregation
ID	Identity
ITS	Intelligent Transport System
KEKs	Key Encrypting Keys
MAC	Message Authentication Code
MAPWPP	A Secure and Efficient Message Authentication Protocol for Vehicular Ad Hoc Networks With Privacy Preservation
PK	Public Key
PPGCV	Privacy Preserving Group Communications Protocol for VANETs
RAISE	Roadside Unit (RSU)-Aided Message Authentication Scheme
RFID	Radio-Frequency Identification
RSU	Road Side Units
SRAAC	Secure Revocable Anonymous Authenticated Inter-Vehicle Communication
TA	Trusted Authorities
TESLA	Timed Efficient Stream Loss-Tolerant Authentication Broadcast Authentication Protocol
TPD	Tamper-Proof Device

VANETs Vehicular Ad Hoc NETworks

VAST VANETs Authentication Using Signatures and TESLA++

VIDs Vehicle’s Identities

Chapter 1

Introduction

1.1 Background

1.1.1 Advantages of Vehicular Ad Hoc Networks

Vehicular Ad Hoc NETworks (VANETs) have becomes a hot research topic [1]within research communities and industry. In the near future, all vehicles built with VANETs [2]will be equipped with the ability to use Dedicated Short Range Communication [3](DSRC), making it possible for vehicles to communicate with each other and with road side units (RSUs)[4].The creation of VANETs provides a great deal of services which will be advantageous to traffic management and safety [5]. For example, if a traffic accident occurred, by using message broadcasting [6, 7] in VANETs, authorities and medical assistance can be notified of the exact location and severity of the accident in a timely and effective manner [8]. Other vehicles around the accident site can be notified to possibly avoid the traffic jam [9]. In addition, some entertainment services [10], such as online videos [11] and games [12]are accessible through VANETs to offer passengers a more comfortable trip-experience. What's more, VANETs enable the application of smart parking [13, 14], which allows drivers to find parking more easily as well as being environmentally friendly. Therefore, VANETs offer drivers a safe and enjoyable driving experience [15] and the transportation authorities an effective and efficient way to manage traffic [16],

while providing society with a way to reduce pollution [17] and ultimately improve the environment of our planet. This is why VANETs are regarded as promising future transportation solutions [18] which have gained increasing attention from both research communities and industries.

1.1.2 Security Weaknesses of VANETs

VANETs, like other kinds of networks, inherit some security weaknesses [19]much as traditional networks do. Any vehicles with malicious behavior, such as those that modify and send reply messages, may produce fatal consequences for other vehicles, even for the whole VENETs. In addition, conditional privacy is another important security requirement that should be taken into consideration [20].That means the users' identities and information such as speed, acceleration and position should be kept secrets from their peer users, while ensuring their this information is available to authorities. What we should emphasize is that compared to other application areas with in VANETs, managing a safety road is the most important aspect and the main purpose of VANETs [21].

1.1.3 Current Studies for Security Issues

Recently, several secure service discovery protocols dealing with foundational issues of security have been investigated by laboratories, companies and traffic authorities. To overcome security weaknesses and to achieve the secure service discovery goals, such as integrity, authentication, and non-repudiation, several protocols were proposed. All the secure service discovery protocols can be categorized into two groups, namely, an infrastructure-based group and a group without an infrastructure. Protocols in each group can be divided into two

categories based on the techniques they adopt. One of the categories makes use of the technique of public keys and the other one adopts the technique of symmetric keys. Each protocol has their own characteristics used to achieve different aspects of security requirements, making itself perform better than others in specific scenarios.

1.2 Motivation

Even though there are several existing service discovery protocols that focus on security issues, to the best of our knowledge, there is still no existing secure gateway discovery protocol. Secure gateway discovery is extremely important; in the situation where drivers and passengers need to connect to the internet and communicate with vehicles or RSU in other networks, they should exchange information through gateways and vehicles in the VANETs that are not subject to security attacks during this communicating.

1.3 Objective

In this thesis, I would like to find a solution to the secure gateway discovery problem that consists of the vehicles' authentication mechanism, the secure gateway discovery process and the secure discovery message exchange. I plan to assign several consecutive overlapping secure clusters in VANETs, so that gateway discovery messages can only be exchanged through the authenticated clusterheads and cluster members. The security requirements that need to be achieved are listed in the following points: 1. Clusters need to be created in a secure way, which means malicious vehicles cannot join as a clusterhead or a cluster-member; 2. Malicious vehicle should not be allowed to participate in the gateway discovery process; 3. The authentication of the senders and the integrity of the messages must be guaranteed; 4. All the gateway discovery messages need

to be traceable.

1.4 Thesis Outline

The remainder of the thesis is organized as follows:

Chapter 2 presents a literature review of the security issues in VANETs; in this chapter we show the challenges and problems VANETs face and conquer, we propose which security requirements should be achieved in VANETs and describe the operating principles of several existing secure service discovery protocols, while evaluating the security requirements they achieved. At the end of this section, a comparison of all the mentioned protocols is presented. Chapter 3 presents the principle of our SEGAL protocol, describing in detail the cluster creation process, the gateway advertisement process and the gateway discovery messages exchange process. Chapter 4 shows the performance evaluation of our proposed SEGAL protocol. Chapter 5 presents the conclusion and the future work of this thesis.

Chapter 2

Literature Review of the Security Issues in VANETs

2.1 Introduction

Vehicular ad hoc networks enable short to medium-range communications between vehicles and roadside units, as well as those among vehicles themselves. These kinds of characteristics support opportunities that enable the blossoming of a large amount of service oriented applications, which can provide drivers and passengers with a great deal of convenience and enjoyment in their trips. Therefore, research groups are determined to find a reliable and secure way to discover and access these services that ensures users' advantages.

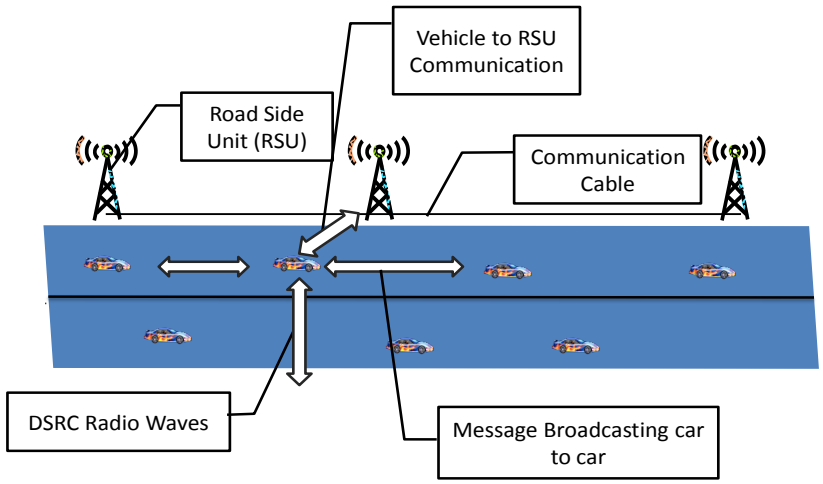

Figure 2.1 The structure of vehicular ad hoc network [30]

Some research communities focus on addressing the connection issue of service discovery in VANETs. However, VANETs, as a special type of network systems, has its own characteristics [22], which are used to distinguish it from other kinds of networks, giving rise to challenges on several levels. The characteristics are shown as follows [23]:

- Nodes in VANETs have comparatively high mobility, which could lead to a unstable link topology.
- The nodes' density could be very low or very high, which may cause an frequent disconnection.
- The features of wireless equipment used in VANETs are specific and limited, in terms of bandwidth.

As described above, it is necessary to find a secure service discovery mechanism to overcome the weakness of VANETs, enable the safe exchange of service discovery messages and guarantee the service requester is resilience against malicious attacks [24].

In this chapter, we present several secure service discovery mechanisms. We show their main ideas, the process steps, and the security goals achieved separately, and make a comparison of them in terms between the security requirements they achieve.

The remainder of this chapter is divided as follows: Section 2.2 describes the challenges of secure service discovery in VANETs. Section 2.3 presents the security requirements that should be achieved. Section 2.4 introduces several existing secure service discovery protocols. Section 2.5 is written to draw comparisons between current mechanisms. Section 2.6 contains a discussion of the different secure service discovery protocols.

2.2 Challenges

2.2.1 Attacks

Denial of Service Attacks

When one vehicle's resources are controlled by the attacker, massive useless or incorrect information can be sent out to jam the communication channel used by the vehicular network [25], which results in the arriving failure of critical information. This attack causes the failure of the vehicle's applications and, even worse, it could increase the danger in regards to the driver who has to depend on the application's information. For example, if a malicious vehicle wants to create a massive pile-up on the highway, it could make an accident and use the DoS attack [26] to prevent the accident warnings from reaching other approaching vehicles.

Message Suppression Attacks

Unlike the DoS attack which leads to dropped packets by jamming the communication channel of a vehicular network, this attack selectively drops these packets that may hold critical information for other vehicles [27]. For instance, an attack could be launched to suppress congestion alerts and to use them at another time, so that other vehicles without acknowledge of the congestion are forced to wait in the traffic.

Fabrication Attacks

A fabrication attack [28] can be initiated by a vehicle broadcasting false

information into the vehicular network. An attacker may choose to fabricate his own information, including his identity, location, speed, or other specific parameters used by the application. For example, the attacker could claim to be a police vehicle in order to acquire some traffic privilege. It's quite a big challenge to fend off this kind of attack in a vehicular network, since traditionally the use of strong identities along with cryptographic authentication would enable the preserving of drives' privacy in the network [4].

Alteration Attacks

An attacker could alter some existing data to launch an alteration attack [27]. The data includes delaying the transmission of the information on purpose, replaying earlier transmission, or altering its own entry within a transmission. For example, the attacker can alter a message alerting other vehicles that the current road is congested while the road is actually clear. Obviously, to defend against this kind of attack, authentication of both the source of the data and the data itself has to be validated in the applications in vehicular networks [4].

Replay Attacks

An attacker could replay the transmission of earlier information, in order to take advantage of the situation of the message at the time of sending [4] . Because it's possible for malicious vehicles to replay previous received messages [29] without detecting of the input of untrue messages since the keys can be reused. Thus, encryption alone is not enough; it's also necessary to authenticate the packets. The purpose of this attack can be to confuse the authorities and to prevent identification of the attacker itself [30].

2.2.2 Adversaries

Greedy Vehicles

In most VANETs, all the vehicles are initially trusted to follow the current protocols. However, some vehicles will violate protocol illegally to get their maximal profits, which may lead to a large amount of cost for their neighbors and the whole system [4]. For example, a greedy vehicle that wants a clear pass to a destination may send out a message falsely informing other vehicles of a congestion ahead. Therefore, the vehicle's neighbors, who trust it, will choose an alternate route, instead of the one which the greedy vehicle will use; as a result, the greedy vehicle gains its clear pass.

Pranksters

These kind of attackers could be bored people seeking out vulnerabilities and hackers who need to catch and utilize the vulnerabilities to be famous [30] . For example, pranksters may lead to a traffic collision by informing one vehicle to slow down while telling the one behind it to speed up. Besides, messages sent by pranksters could give rise to an information congestion.

2.3 Security Requirement

Authentication

To ensure a message's origin and to control authorization levels of vehicles, every message in a vehicular network has to be authenticated [27, 30]. Vehicles would assign all the messages their private key and the messages' certificate.

Once the receiver vehicle receives the message, the key and certificate is checked and the message can then be verified. A certain overhead will be produced in this authentication process and the ECC approach (Elliptic Curve Cryptography) can be used as an efficient public key cryptosystem to reduce this overhead.

Availability

It's necessary to make vehicular networks available all the time. The real-time capabilities are required by many applications in vehicular networks and these applications require a faster response than traditional sensor networks and even ad hoc networks. For some applications a delay in seconds will make the message useless [4].

Attempts to meet real-time demands will make the network vulnerable to the DoS attack. For some message, even a delay of a millisecond is intolerable; the problem is furthermore troublesome because of the unreliable application layer, since one potential way to deal with unreliable transmission is to store partial messages in the hope that the next transmission will complete the message.

Privacy

Private information of the drivers such as real identity, trip path, speed, should be protected from unauthorized observers [31]. Temporary (anonymous) keys can be used to protect the privacy. To change the keys frequently, each temporary key is used just for one time and expires after usage. All the keys can be stored in the TPD(Tamper-Proof Device) and reloaded when the vehicle gets an official

checkup.

To preserve the real identity of the vehicles, an ELP (Electronic License Plate), which is initiated in the factory for every new vehicle, can be used to provide an identification number for every vehicle. By holding the ELP with the RFID technology, the vehicle can be identified anywhere.

Non-Repudiation

To prevent cheaters from denying their crimes, non-repudiation is quite important for the system's ability to identify the attack vehicles even after the attack has happened [32]. Any official side holding authorization can retrieve the data related to the vehicles, such as the trip-rout, speed and violation record. Once any violation happens, it is stored in the TPD together with other information, which can be obtained by authorized observers.

2.4 Encryption Systems

To avoid attacks and to achieve security goals, messages should be encrypted while exchanged through the networks. Presently, there are two kinds of encryption systems widely used in secure service discovery protocols, namely, the public key encryption system and the symmetric key cryptography.

Public Key Encryption System

On the contrary, in the public key encryption system, each node has its own public and private key pair. The public key can be known by all the nodes in the

networks, while the private key is kept only by the respective owner. Once a message is encrypted with a public key, it can only be decrypted by the corresponding private key, and vice versa.

A major advantage to the public key encryption system is increased security; only the respective owner knows the private key. What is more, unlike the symmetric system, public key encryption achieves the integrity and authentication of the message, as well as privacy. Moreover, the public key system guarantees repudiation by providing a digital signature algorithm.

The most significant disadvantage is that the speed of the public key system is slower than that of the symmetric key system. Furthermore, its computing complexity is higher than the symmetric algorithm's; the public key system uses more computer resources.

Symmetric Key Cryptography

In symmetric cryptography, messages are encrypted and decrypted with the same key.

The mainly advantages of using symmetric key cryptography are that it achieves a higher speed and it is much easier to implement than the public key encryption system. This is because it uses the same key in both encryption and decryption steps. Furthermore, symmetric key encryption use less computer resources than the public key encryption system.

However, the symmetric key system has its disadvantages. For example, it needs a secure channel to propagate the secret key. In addition, it cannot guarantee the

authentication, since all the nodes use the same secret key to encrypt and decrypt messages.

2.5 Secure service discovery protocols

The current protocols for secure service discovery in VANETs can be categorized into infrastructure-based protocols and infrastructure-less protocols based on whether they use road components or not. In each category, we classify the protocols by encryption methods they adopt: for example, public key encryption and symmetric key encryption.

2.5.1 Infrastructure-Based

Security requirements can be achieved by infrastructure-based secure service discovery protocols through adopting a public key encryption [33, 34, 35,38] or a symmetric key encryption[36, 37,41] .

2.5.1.1 Public Key Encryption System

The Peer-to-Peer Anonymous Authentication (PPAA) protocol [34], which is presented by Tsang et al., focuses on the authentication and privacy of both clients and servers. An authentication scheme proposed by Calandriello et al. [33]adopts the group key concept to generate the key pairs of public and private keys. Lin et al. presented a security and privacy preserving protocol [35] which adopts the techniques of Group Signature and Identity-based Signature (GSIS). Here we set MAPWPP (A Secure and Efficient Message Authentication Protocol for Vehicular Ad Hoc Networks with Privacy Preservation) which was proposed by Subhashree et al. [36]as an example that describes this type of

infrastructure-based secure service discovery protocol.

(1). **MAPWPP** [38]

Subhashree et al. presented MAPWPP (A Secure and Efficient Message Authentication Protocol for Vehicular Ad Hoc Networks with Privacy Preservation). MAPWPP is based on road side units（RSUs）. A RSU stores the private and public key pairs of vehicles in its communication range, which are generated with ECDSA[39]; A RSU furthermore arranges each vehicle with pseudo and temporary IDs. A sender broadcasts the message signed with its private key and attaches a pseudo ID to it[40] . Any receiver who receives this message, queries the RSU for the corresponding public key by presenting the sender's temporary ID. The RSU looks up the pseudo ID and discovers the corresponding actual ID and then broadcasts the needed public key of the sender. So any receiver can use the public key to verify the sender's signature to achieve authentication without knowing the actual ID of the sender.

System assumption

- Each vehicle has a particular set of parameters to specify their elliptic curve, which is used to generate the pool of public/private key pairs of vehicles.
- The communication range of RSUs are larger than that of other vehicles.

Process

step 1: Vehicle Registration

Each vehicle registers itself with trusted authorities by providing its identity and address before VANETs setup. After verification, vehicles generatetheir public/private key pool by using ECDSA; vehicles then register the public keys in their key pool with trusted authorities as their pseudo VIDs against their real IDs. Each key pool has its own life period. After the expiry time, the key pool will be regenerated and registered.

step 2: RSU Installation

RSUs are deployed at each road section after the vehicles' registration and they can get information on all vehicles from the transport authorities. Similarly， RSUs register themselves and their public keys with trusted authorities.

step 3: Temporary Identity Acquisition

When a vehicle enters into the communication range of a RSU, it sends its public keys and identity to the RSU. After validating the received information from the vehicle, a RSU uses the VID of the vehicle and its own private key to calculate a temporary identity and sends it to the vehicle as a reply.

step 4: Message Transfer

- Broadcast of message

The sender chooses its private key from its key pool to sign the message, which includes the real ID of the sender, and sends it to the RSU.

Upon receiving this message, the vehicle queries the nearby RSU for the public key by sending it the temporary ID. The RSU uses the temporary ID to calculate the corresponding public key and to broadcasts it.

The receiver uses ECDSA method to verify the signature with the corresponding public key.

- Personalized message transfer

Firstly, the sender checks whether the destination vehicle is in the range of the RSU. The sender sends out its temporary ID and the temporary ID of the destination vehicle to the RSU, the RSU then verified this on its database and sends back the information to the sender.

Secondly, if the destination is in the range of the RSU, the process of communication starts.

If the destination vehicle is in the range of both the RSU and the source vehicle, it uses ECDSA protocol to achieve authentication by verifying whether the elliptic parameter calculated by the destination vehicle is equal to the one used by the sender. After authentication, message transfer starts with the encryption of ECDSA.

If the destination vehicle is out of the range of the source vehicle, the authentication and message transfer should be done with the help of an intermediate vehicle. The method of information exchange between the source and the intermediate vehicle and likewise between the intermediate vehicle and the destination is the same as that one directly between the source and the destination.

Security Evaluation

Authentication, Integrity and Non-repudiation

All vehicles have been registered with the TA to ensure the authentication of communication. And the use of the public/ private key pair generated by using ECDSA prevents the impersonation attack. The real IDs of vehicles are stored in RSUs to ensure Non-repudiation.

Conditional privacy preservation

The use of temporary IDs ensure that the real IDs of vehicles are kept secret from each other; however, the real IDs are stored in RSUs and can be traced by TAs.

Prevention from black and grey hole attack

All vehicles are monitored by their one-hop neighbors. Any vehicles which are misbehaving can be detected and added to the blacklist of RSUs.

2.5.1.2 Symmetric Key Cryptography

In this category, Zhang et al. presented the authentication scheme called RAISE [37]. Their scheme is based on RSUs. In their protocol, message authentication verification is performed through the RSUs. They used k-anonymity in order to guarantee the privacy of users. They proposed a variant that works with the absence of RSUs. Their proposed scheme has a low computation and communication overhead. You Lu et al. proposed GSA [41] protocol, which is based on the method of TESLA [42] and use the method by which the group relieves authentication delay caused by TESLA. Different schemes are used by GSA protocol when dealing respectively with inter-group and intra-group communication based on their own characteristics. Wasef et al. presented a protocol called a Privacy Preserving Group Communications Protocol for Vehicular Ad Hoc Networks (PPGCV) [36]. The proposed protocol uses a probabilistic key distribution concept and relies on a security threshold mechanism. It is based on group communications and guarantees the confidentiality of users' information. The proposed PPGCV has the property of being stateless while computing a new key and updating the compromised ones

under the condition that the number of revoked nodes does not bypass a certain value.

(1). **RAISE** [37]

Zhang et al. proposed a Novel Roadside Unit (RSU)-Aided Message Authentication Scheme) (RAISE). The main idea of RAISE is that the RSUs and the vehicles in their communication range proceed first with mutual authentication and key verification. When there is a need to communicate with other vehicles, the sender sends its safety message attached with a MAC [42] tag to the RSU; the RSU then verifies the MAC tag and sends the authentication result to all vehicles in its range. In this way, the speed of verification will be much faster than the traditional PKI-based scheme.

System Assumption

(1). The VANETs have been divided into two hierarchical layers. The upper layers consists of Application Servers (ASs) and RSUs. Services are supported by ASs; moreover, RSUs act as gateways for information delivery to lower layer vehicles. Messages can be exchanged securely between ASs and RSUs that obey the transport layer security.

(2). RSUs and vehicles are time synchronized, and each message has a time stamp.

(3). Compared to over vehicles, RSUs have a larger communication range, in addition to a higher computation capability, and are trusted.

Process

Step 1: Symmetric Key Establishment

When a vehicle (*v)* detects a RSU (*R),* a mutual authentication process is undergone while the vehicle in question shares its secret key with *R.* The process

is described as follows.

(1)The vehicle first sends its certificate encrypted with the PK (public key) of *R* to *R;*

(2)When *R* receives this message, *R* sends back a message encrypted with its SK (private key) including the pseudo ID of *v* and the corresponding information needed in order to work out the shared secret key *K*;

(3)At the reception of the message sent by *R*, *v* verifies the signature and then sends back a signature message of that includes the ID of *R* encrypted with *v*'s own SK.

R maintains an ID-Key table and a Trace evidence table to keep the information received from a vehicle including a vehicle's pseudo ID, secret key, certificate and receive time. If the interval between the current time and the received time of *R* is more than the threshold, the information should be stored in the trace table. Otherwise, messages should be stored in the ID-Key table.

Once a vehicle goes outside the radio range of R, a vehicle updates its anonymous certificate.

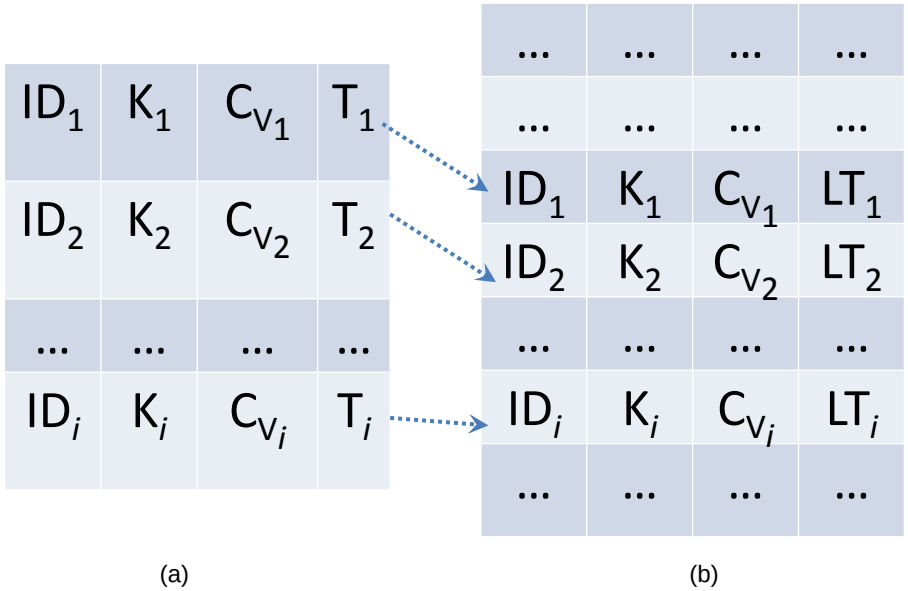

Figure 2.2ID-Key table. And correspondence Trace evidence table [37].

Step 2: Hash Aggregation

The vehicle $V_i$generatesK_i, using it to compute the MAC of its message. Then it one-hop broadcasts the MAC-attached message. At this time, only $V_i$and R can verify this message. So R has the responsibility to aggregate multiple authenticated messages by using hash function, packeting it as hash aggregation and then one-hop broadcasting it so that other vehicles can verify the message. The formation of hash aggregation is as follows:

$$\text{HAggt} = H(ID_1||M_1||TS_1)||H(ID_2||M_2||TS_2)||\ldots||H(ID_n||M_n||TS_n)$$

And the message is signed with *R*'s SK.

Step 3: Verification

Upon receiving a message from other vehicles, a vehicle first buffers the message until it receives the HAggt. It then verifies the buffered messages by comparing them to the consecutive two disaggregated messages. If the buffered message matches one of the disaggregated messages, it will be consumed.

Step 4: Conditional Privacy

RSUs assign a pseudo ID to *k* vehicles in its radio range. That is to say these *k* vehicles have the same pseudo ID, so an adversary cannot trace any one of them. A pseudo ID corresponds to *k* symmetric keys shared between each vehicle and the RSU stored in the ID-Key Table. When it receives a message from a vehicle, a RSU checks the MAC tags made by corresponding *k* symmetric keys and compares them to the MAC tag it received in the packet. If there is a match, the packet is considered valid. Otherwise, the packet is dropped.

Security Evaluation

Message Integrity and Source Authentication

A sender attaches a MAC tag to each message, calculated by the assigned RSU, so that the RSU can verify the MAC tag and know the corresponding ID of the sender for authentication. However, malicious vehicles cannot get the key assigned by a RSU, which saves to prevent their attacks.

Prevention of Internal Attack

The use of a pseudo ID can prevents internal attacks. Even if an adversary gets a vehicle's secret key, it cannot trace the compromised vehicle's movement, since all group members have the same pseudo ID.

Replay Attack Resistance

Messages in RAISE protocol are set with time stamps and all the vehicles are time synchronized, so the replay attacks are not accepted.

Conditional Privacy Preservation

A pseudo ID makes it so that a vehicle cannot be traced by other vehicles. On the

other hand, RSUs can get the corresponding anonymous certificate of a pseudo ID, which can be used by TA to trace the real ID of a vehicle.

(2). **GSA** [41]

You Lu et al. proposed a Group-based Source Authentication Protocol (GSA), which is based on both the method of TESLA and the attribute of group (Figure 2.3); this protocol is proposed to relieve the authentication delay caused by TESLA. Different schemes are used by GSA protocol when dealing respectively with inter-group and intra-group communication in a manner sensitive to their own characteristics.

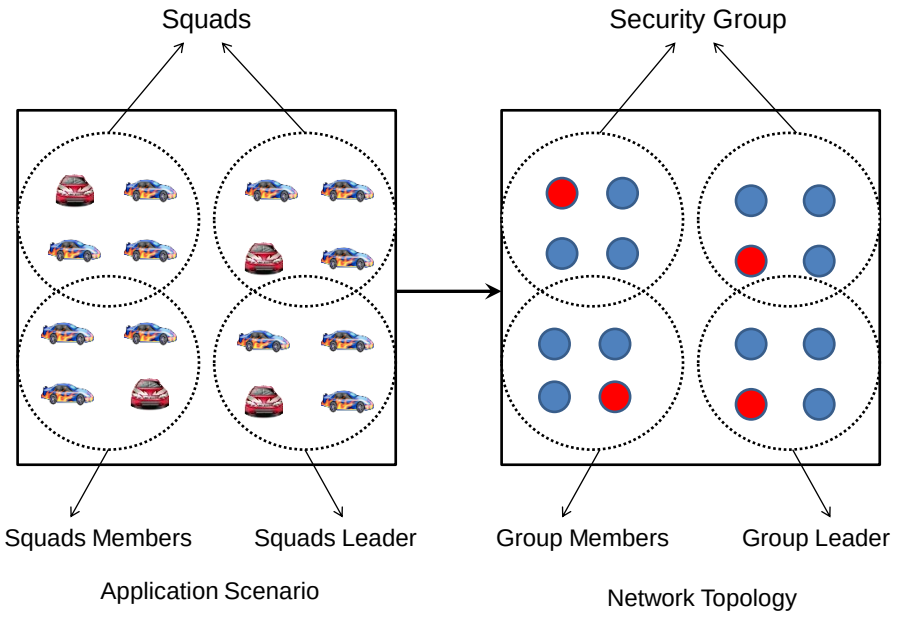

Figure 2.3 Groups in Military Troops [41]

System Assumption

- In GSA protocol, all the vehicles are divided into several squads. Each squad is regarded as a mobility group. Each squad has its leader which is responsible for the delivery of secure messages to its squad members and the

exchanging of information with other squads.

- Some easy-to-compute characters are picked up as some of the group attributes, since this is easy to update periodically.
- The authors managed the scenario in a military vehicle troop for the purpose of presenting the performance of GSA.
- Initial group members are not any attackers.

Process

Step 1: Initialization (Sender setup)

The original TESLA [42] scheme is used to authenticate a group member.

The sender generates a one-way hash chain, whose element value is the key to calculating the MAC of P(*i*). The key is sent simultaneously with the current package through intra-group communication, while it is sent after a time interval *d* with the current package through inter-group communication.

Group Membership authentication

To gather the GID (Group Identification Certificate) and the mobility information of the receiver group, characteristics selected to describe the attributes of the group, the sender first broadcasts declaration. Then each receiver replies as the TESLA scheme requires. MAC values are calculated with the key for the initial GID and the initial mobility of the receiver group. After disclosure delay, the server can then authenticate whether the receiver is one of its group members by using the disclosed key to check whether the calculated MAC equals that of the received MAC.

Bootstrapping Parameter Transmission

Bootstrapping packets will then be sent to receivers by the server; these include the data of the current time interval (I_i), the life time of the encryption key (T_{int}) , the beginning time of the current time interval (T_{i_begain}), the pseudo random function (*F*)- used to calculate the key- and its inverse function *F'*- used to commit the key- as well as the disclosure interval (d) and the key sent in time interval $i - d$.

Step 2: Intra-group Communication

Sender Operation

Sent data packets are assigned the key of the same time interval. The message is encrypted with MAC, while the key is encrypted with the attributes of the group. Therefore, only the group members can decrypt the key with their attributes.

Receiver Operation

Upon reception of the data packets from the sender, each receiver uses its attributes to decrypt the key. Then it uses the key and its key to calculate MAC and to check whether the calculated MAC equals to the received MAC. The equation presents the data packet that can be immediately delivered to the application; otherwise, the data packet is dropped.

Group Membership Update

The group membership authentication phase should be repeated periodically. And every time following the group membership authentication process, the bootstrapping parameter transmission phase is implemented again.

Step 3: Inter-group Communication

Sender Operation

The data packets are multicast by the sender including the message and the key

of the time interval $i - d$, which is $K(i - d)$, and the MAC of the current packet.

Receiver Operation

Upon receiving a data packet, the receiver implements the following security condition: first, the disclosure delay is required to be longer than the time required to multicast this message to all recipients in the network. Only messages which satisfy the security condition are accepted; the receiver then verifies $K(i - d)$'s ability by using it to calculate the older $K(j)$ ($j < i - d$) and checking whether the new K's ability equals the previously received one; finally, $K(i - d)$ is used to calculate the MAC value of $M(i - d)$;this is compared with the previously received MAC of $M(i - d)$; the packet with an equal MAC value is accepted.

Security Evaluation

Secure group creation

Attributes are only used to calculate the MAC value without being transmitted. So even if a malicious vehicle gets the group to create a reply message that is sent by the receiver, it cannot masquerade as a group member without the group attributes.

Message authentication

By using TESLA scheme and the group attributes, GSA protocol can authenticate whether the received message is sent by the expected sender with less overhead and less time consumption. Replay attacks can be prevented since the attributes of vehicles change dynamically and attributes provide important data to calculate MAC.

(3). PPGCV [36]

A Privacy Preserving Group Communications Protocol for Vehicular Ad Hoc Networks (PPGCV) is based on a probabilistic key distribution approach and a security threshold scheme. PPGCV provides an efficient and scalable group communications; and, at the same time it preserves the privacy of users. In addition, PPGCV provides conditional and full stateless property; this allows a node to calculate the new group key and to update its compromised keys. This is even possible if the node misses the group rekeying process provided that number of the revoked nodes does not exceed a certain number.

System Assumption

- There exists a key server acting as the group manager, which distributes the keys to all nodes in the network.
- A revoked node is instantly detected by the key server.
- The group size may range from a select number of nodes to all nodes in the network.

Process

Step 1: Initial Key Bootstrapping

The key server has a key pool P which consists of l keys, from which each node will randomly pick a set of keys R; these consist of m distinct keys. Those keys will be used as the Key Encrypting Keys (KEKs). An initial group key $k_g$is loaded in each node. Each node is then loaded with the first key in the key chain that is used for authentication using TESLA.

Threshold scheme (t, n):

$$F_{thresh}(x) = a_0 + a_1x + a_2x^2 + \cdots + a_{t-1}x^{t-1} \bmod p_{ca}$$

Where $(a_0, a_1, a_2, \ldots, a_{t-1}) \in Z^*_{p_{ca}}$,*n* is the total number of participants, and *t* is the minimum number of participants that can collude to reveal the shared secret. The key server will also select a set of deterministic functions $(g_1, g_2, \ldots, g_{t-1})$, which will be used to generate thecoefficients $(a_0, a_1, a_2, \ldots, a_{t-1})$. Finally, each vehicle will be loaded with *t*, p_{ca} and the set$(g_1, g_2, \ldots, g_{t-1})$.

Step 2: Group Rekeying

Group rekeying happens after a node u revoked.

The key server broadcasts a node revocation message that contains M, the ID of the revoked node (i.e., *u*) and $f_{k'_g}(0).k'_g$ is the new group key, while f_k is a family of pseudo-random functions.When receiving and verifying the revocation message, each node will check if it possesses the k_M or not. If it does, the node then computes the intermediate keyk_{im} independently. If it does not, this vehicle V randomly selects r keys out of the m keys in R_v; and, vehicle V will broadcast the IDs of the selected keys to the neighboring vehicles. If any one of the neighboring vehicles have the shared keys, it will encrypt the k_{im} with the shared key and send it to vehicle V. Each node computes the new group key $k'_g = f_{k_{im}}(0)$, and checks whether the one calculated out is equal to the one in the revocation message. No vehicle will forward k_{im} to the revoked node, since the ID of the revoked node is contained in the revocation message. Each node updates their key set:$k'_i = f_{k_i}(0)$and$\forall k_i \in R_v$ .If a node has a key that belongs to the revoked node,k_{im}must be used to calculate the $f_{thresh}(r_1)$, in which random $r_1 \in Z^*_{p_{ca}}$. Additionally, the shadow $(f_{thresh}(r_1), r_1)$ and the timestamp

corresponding to the start of the rekeying process must be saved. Finally, every node erasesk_{im}, $(a_1, a_2, \ldots, a_{t-1})$, and the original k'_i s.

Step 3: Intermediate Key Regeneration

If a vehicle *y* misses a rekeying process, it may get the corresponding intermediate key to generate the new group key and to update any compromised keys it may have.

The vehicle *y* selects *r* keys from its key set and broadcasts their IDs, vehicle *y* additionally selects an intermediate key request, and the timestamp of the last performed rekeying process. Any vehicles receiving the request will verify the IDs of those *r* keys not revoked and select the shadows to be sent by using the received timestamp. They will then encrypt the shadows with k'_g and broadcast them. Any vehicles that receive different *t-1* shadows can use the received shadows and its own shadow to generate intermediate keys, and to send them to the requesting node *y*, which is encrypted with a key shared between them. The vehicle *y* updates its non-compromised keys. *y* broadcasts a confirmation message encrypted with the new group key k'_g. Upon receiving the confirmation message, any vehicle which has received shadows from others must erase those shadows and the regenerated k_{im}.

Security Evaluation

Forward and backward secrecy

That is achieved because the group key will change after a node revocation or addition.

Authentication

This occurs because a message needs to be checked to see if it is encrypted with the correct group key. Another reason is that before regenerating k_{im}, each node needs to check that the IDs of the keys sent by the requesting vehicle have not been revoked.

Protection against collusion

The above mentioned is necessary because following the rekeying process, each node possesses only a shadow instead of k_{im}.

Privacy

To establish a secure connection, a node selects random *r* keys from its key set and broadcasts the IDs of those keys. Each time the selected keys are different. Therefore, the real ID of the node is hidden.

2.5.2 Infrastructure-Less

The authentication of infrastructure-less secure service discovery can rely on public key [43, 44, 45] or symmetric key [42, 46, 47]as described below.

2.5.1.3 Public Key Encryption System

Studer et al. presented VAST [43], which is a hybrid authentication mechanism combing TESLA++ (a modified version of TESLA) and ECDSA [39] signature. VAST protocol provides advantages for both these two schemes. In the Secure Revocable Anonymous Authenticated Inter-Vehicle Communication (SRAAC) mechanism, proposed by Fisher et al. [44],the exchange of certificates is blinded and based on a quorum. Kamat et al. [45] proposed a framework that provides security to VANETs. To achieve the security goals of authentication, integrity,

confidentiality and non repudiation, they used identity based cryptography. Their scheme is flexible and can be tuned to adapt to different levels of trust and privacy. Here in our survey, we describe VAST protocol as a typical infrastructure-less protocol with asymmetric keys.

(1). **VAST**[43]

(VANET Authentication using Signatures and TESLA++)

Main idea

Ahren et al. proposed VAST protocol, which stands for VANETs' Authentication using Signatures and TESLA++ [43]. It draws upon advantages elements in both TESLA++ and ECDSA signature; since TESLA++ protocol provides broadcast authentication, and DoS resilience, while ECDSA [39] signature provides necessary non-repudiation and multi-hop communication. What is more, the proposed VAST protocol is a flexible solution; it enables developers to tune parameters, which are used in ECDSA signature, to satisfy different properties.

Processes

Step 1: TESLA++

TESLA ++ is based on TESLA (Timed Efficient Stream Loss-Tolerant Authentication) with compressed MAC of the message; this reduces the memory requirement when compared with that of TESLA. The steps of TESLA++ in comparison with those of TESLA are shown as follows.

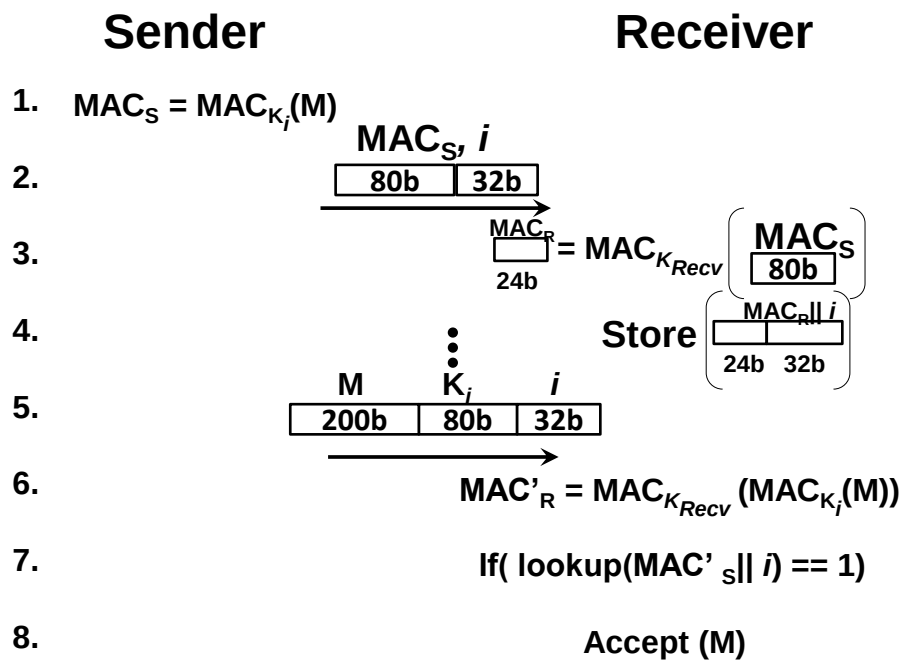

Figure 2.4The process of TESLA++ [43]

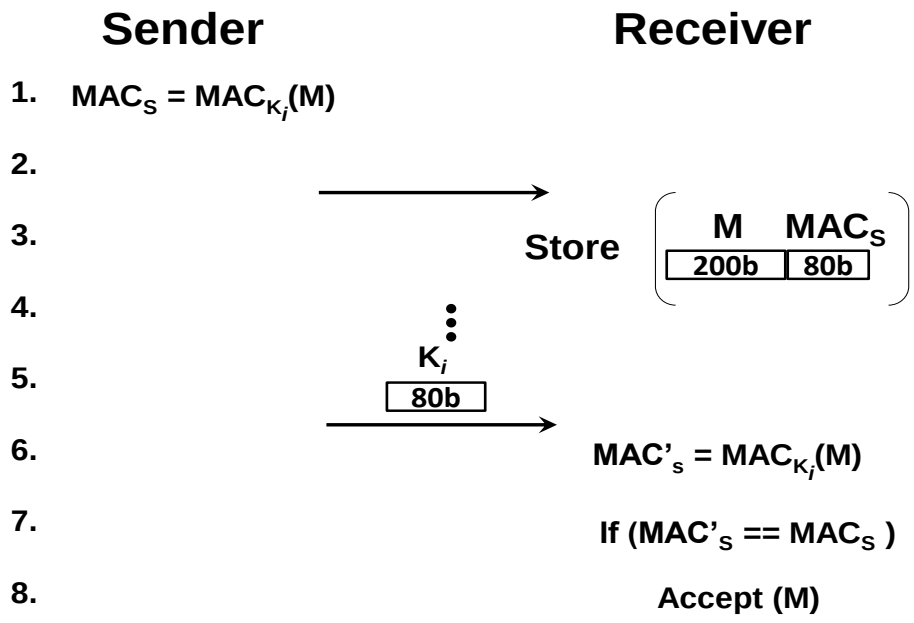

Figure 2.5The process of TESLA [43]

Step 2: ECDSA Signature

(1). The sender applies the ECDSA protocol to calculate the signature σ_{S_i} using the ECDSA keys.

(2). The sender calculates the MAC_S by encoding the message and its signature σ_{s_i} using the key of the current time interval K_i.

(3). The sender only broadcasts the calculated MAC_S and key index *i* to receivers.

(4). At the reception of the MAC_S and key index *i*, the receiver re-MACs the received MAC_S with its key K_{Recv} to reduce the memory requirement for MAC from 80b to 24b. The result is MAC_R.

(5). The receiver stores $MAC_R \| i$.

(6). When the key K_i is able to be disclosed, it is broadcast with all messages M and their signatures σ_{s_i}.

(7). To verify the validity of the message and signature, upon obtaining the information includes in Step 6, the receiver checks first the hash chain to ensure the received key is used.

(8). After the verification of validity, the receiver calculates the MAC with the disclosed key and message and then re-MACs it to get the compressed MAC, MAC'_S.

(9). The receiver compares MAC'_S with the MACs stored in its memory to discover if there is a corresponding MAC/key index pair in its memory.

(10). If there is a corresponding MAC/key index pair and non-repudiation is necessary, ECDSA keys are used to verify the signature of the message and to decide whether the message can be accepted or not.

(11). However, if there is no matching MAC/key index pair in the receiver's memory, the receiver as certain whether the utilization of CPU and the number of messages in the processing queue are, or can take place in the threshold. If this is the case, ECDSA keys are used to verify the signature of the message and to decide whether the message can be accepted or not.

Security Evaluation

Authentication

Both TESLA++ and the ECDSA signature enable VAST to satisfy authentication requirements. TESLA++ adopts the technique of symmetric keys while the ECDSA signature adopts the technique of digital signatures to verify the senders of messages.

Denial of Service (DoS) Resistance

VAST is a flexible protocol which can mold the parameters to satisfy the specific property requirement. When DoS resistance is a vital issue, the adopting of TESLA++, which applies the technique of symmetric keys, provides a faster computation speed than that of the digital signature technique; this gives less of a chance to DoS attackers.

Non-repudiation

In VAST protocol, no matter whether a vehicle sends or forwards a message, it will attach its own signature to the message. Therefore, any sender can be traced back by law enforcement agencies.

Multi-hop Authentication

When the received message needs to be forwarded to other vehicles, the original sender's MAC of TESLA++ may have been missed. To solve this challenge, the ECDSA signature is used for authentication.

2.4.2.1. Symmetric Key

Lin et al. presented research in a Timed Efficient and Secure Vehicular

Communication (TSVC) scheme [47] that preserves privacy for users. The TSVC minimizes packet overhead due to signature overhead and reduces the latency incurred from the verification of a signature. Riley et al. proposed in [46] a secure scheme for cooperative collision warning applications in VANETs. They proposed a delay efficient authentication protocol, named GHAP; this protocol relays through a group based behavior. Their protocol guarantees individual privacy and permits the non-repudiation of messages. Perrig et al. proposed a Timed Efficient Stream Loss-Tolerant Authentication (TESLA) protocol [42]. This uses symmetric keys, which utilizes a Message Authentication Code (MAC), to achieve asymmetric security properties, authenticating broadcasts. TESLA protocol holds a relatively low computational overhead because of its use of symmetric cryptography. However, it does not take the preservation of users' privacy into consideration. In addition, it needs memory to buffer messages. Besides, message authentication is delayed.

In the following sections, TSVC protocols will be introduced to present infrastructure-less protocol with symmetric keys.

(1). **TSVC** [47]

Xiaodong et al. proposed new TESLA (Timed Efficient Stream Loss-Tolerant Authentication) based Secure Vehicular Communication protocol with privacy preserving protocol (TSVC) based on TESLA protocol and public key cryptography. The sender generates a hash chain and uses this to compute the MAC in advance. The data message is sent encrypted with the MAC; this message can be verified by the receivers after a delay, by using the disclosed key released in the key release packet. In the first key release packet, in addition to

the released key, the sender also attaches a signature to the traditional public key technique. On the other hand, the following key release packets only deliver the key used to verify the MAC. Receivers use the released key to verify the authentication of the message.

System assumption

- Each vehicle generates a one-way hash chain with the one-way function H ($h_i = \mathrm{H}^{\mathrm{j-i}}(h_j),\ j > i$) and uses the values to compute the MAC, consequently encrypting the message.
- The clock at senders and receivers in TSVC are settled and loosely synchronized.
- Groups are formatted in TSVC, which contains all the vehicles in the communication range of each centered vehicle.
- The sender can estimate the message transmission delay with the information of traffic density.
- Each packet is equipped with a timestamp.

Processes

Step 1: Vehicle Setup

All vehicles need to be installed with a series of short life time public/private key pairs as well as the anonymous certificates$Cert_i$. Each vehicle generatesits own hash chain by randomly picking up a seed S and making $h_n = S$and then using the pseudo function$h_i = \mathrm{H}^{\mathrm{j-i}}(h_j),\ j > i$. Each MAC is computed with one chain value and attached to one message.

Step 2: Broadcasting Authenticated Messages

The packets are divided into two categories; these are named data packet and key release packet, which are sent after a fixed delay.

Data Packet

The format of the data packet is shown in the following:

$$P_j = < PVID, M_j, MAC_{hj}\ (M_j||T_j),\ T_j, \text{index} >, \text{j} \geqslant 1$$

where $PVID$ is the pseudo ID of the sender, which is related to the current public key certificate; M_1 is the first safety message; T_1 is the time when the sender sends this packet and *index* is the hash order value.

Key Release Packet

The first key release packet is sent after a disclosure delay of the first data packet; it has the following format:

$$kr\_P_1 = < PVID, Sig_{SK_O}(h_1, index, T_1'),\ h_1, \text{index},\ T_1'\ , Cert_O >$$

where SK_O is the current private key corresponding to the anonymous certificate$Cert_O$ of the sender; $T_1'$is the time when the first key release packet is sent;$Sig_{SK_O}(h_1, index, T_1')$ relays that the commitment of the hash chain h_1 is signed with the traditional public key.

On the other hand, the following key release packet is sent in the following format:

$$kr\_P_1 = < PVID, h_j, \text{index} = \text{j}\ ,\ T_j' >, \text{j} > 1$$

Step 3: Authentication at Receiver

Each time the receiver buffers the data packet first. Upon receiving the first key release packet, the receiver can verify the packet by first using the public key of the sender to verify the sender and then using the released key to verify the MAC. When the receiver receives the following key release packet, the receiver just checks the MAC tag to verify the authentication of the message.

Step 4: Setup of the Key Disclosure Delay

The key disclosure delay should be set as longer than the travel time of a message from sender to all the receivers.

Security Evaluation

Data source privacy

This occurs when the vehicle randomly picks up a public/private key pair as it sends the first key release packet, which cannot indicate its real id. Besides, each one-way chain has a short life time, so it is difficult to track the vehicle with the anonymous public key certificate.

Traceability

The authorities could trace back the senders by matching the sender's pseudo ID to the real identities in their databases.

Data source authentication

With the delayed disclosed key, receivers could verify the source by computing the MAC and comparing it with the previously received one. Messages sent by malicious vehicles will contradict the hash function.

Resilient to the replay attack

Due to the use of timestamp, the replay attack is prevented.

2.6 Comparison

Table 1 shows a summary of the comparison regarding the achievement of security requirements among all infrastructure-based secure service discovery protocols mentioned above. Table 2 presents a summary of the comparison about the secure goals achieved by all the infrastructure-less secure service discovery protocols mentioned above.

Table 1 Infrastructure-Based Secure Service Discovery

[Author, year]	Protocol Description	Authentication	Integrity	Confidentiality	Non-repudiation
Public Key Cryptography					
[Tsang, 2008]	PPAA	X			
[Calandriello, 2007]	Efficient and robust pseudonymous authentication in VANTEs	X			
[Lin, 2007]	GSIS	X		X	X
[Behera, 2011]	MAPWPP	X	X	X	X
Symmetric Key Cryptography					
[Zhang, 2008]	RAISE	X	X	X	X
[Lu, 2010]	GSA	X			
[Wasef, 2008]	PPGCV	X		X	

Table 2 Infrastructure-Less Secure Service Discovery

[Author, year]	Protocol Description	Authentication	Integrity	Confidentiality	Non-repudiation
Public Key Cryptography					
[Studer, 2009]	VAST	X			X
[Fischer, 2006]	SRAAC	X			X
[Kamat, 2006]	An identity-based security framework for VANETs	X	X	X	X
Symmetric Key Cryptography					
[Lin, 2008]	TSVC	X			
[Perrig, 2002]	TESLA	X			

2.7 Discussion

The services provided by Vehicular Ad Hoc Networks make VANETs promising technology concerned with consumers, automobile manufacturers and government. However, VANETs are still a new and developing field which is vulnerable to malicious attacks. To prevent malicious attacks, security service discovery protocols were proposed with different ideas. Each protocol has its own property to be used in a given situation, satisfying the specific security requirements. In this survey, we present some current typical security service discovery protocols using the aspects of their main ideas, processes and the security goals they achieved; we make a summary of comparison. So people who need to pick up a security service discovery protocol to achieve needed security

requirements could have their choice among the protocols we introduced in this paper.

Even though there are several discovery protocols focusing on secure service discovery, there is no existing discovery protocol focusing on secure gateway discovery. It is obvious that the security of gateway discovery is quite important; there is a pressing need to have a secure gateway discovery protocol, since gateways plays a crucial role when drivers and passengers need to connect to the internet or to be offered services.

Chapter 3

Secure Gateway Localization and Communication System for Vehicular Ad Hoc Networks (SEGAL protocol)

3.1. Introduction

Drivers and passengers need to be able to connect to the Internet or communicate to different networks through gateways from their vehicles, and have access to a plethora of services. This is why gateway discovery in VANETs is a very promising research subject. However, existing discovery protocols in VANETs did not focus on secure gateway discovery in particular, but they considered the discovery of secure services in general. Secure gateway discovery is very important, because drivers and passengers should not be subject to security attacks while they are communicating through the gateway.

In this chapter, we discuss the importance of secure gateway discovery and communication by presenting the possible attacks related to gateway discovery in the VANETs. Then, we propose the secure gateway discovery and communication system for VANETs (SEGAL). In our proposed SEGAL protocol, we focus on the authentication aspect because message authentication is very important in ensuring secure gateway discovery and communication. Our proposed SEGAL builds a secure clustered VANET and permits the exchange of gateway discovery messages through authenticated clusterheads and cluster

members.

The remainder of this chapter is organized as follows: Section 3.2presents the possible attacks that can happen during the gateway discovery process in VANETs and the security goals that need to be achieved. Section 3.3 describes our proposed SEGAL protocol. Section 3.4 discusses how the required security goals for gateway discovery are achieved through our proposed protocol.

3.2. System Model

In our system model, we consider a vehicular ad hoc network that comprises gateways and vehicles. Gateways are road components that permit the connection of vehicles to the Internet or to other networks using different access technologies or access modes. Vehicles are circulating along the roads, and they are characterized by their high density and mobility. We consider a Manhattan model of straight lane roads and perpendicular lanes. Gateways are located on some intersections of the straight and the perpendicular lanes. We assume that a certification server exists in our model that permits the registration of vehicles for security purposes. The purpose from our protocol is to form authenticated cluster groups from the circulating vehicles, and exchange gateway advertisement and discovery messages in a secure way.

3.3. Gateway Discovery Possible Attacks and Security Goals

In the following, we will describe the possible attacks that can occur during the gateway discovery processes and how they attack the networks. Simultaneously,

we will present the security goals that should be achieved in my secure gateway discovery protocol, which is aimed at to overcome the possible attacks.

Possible attack 1: A malicious vehicle not registered with a certification server tries to join a cluster group. That is because in the initial process, a vehicle broadcasts a message to inform all vehicles in its communication range to create a cluster. Consequently, upon receiving this message, any malicious vehicle is identified as a cluster member. This unreliable cluster creation is extremely dangerous. A cluster-based protocol focusing on security issues will become totally meaningless, if it could not offer a reliable cluster creation.

Security goal 1: The new protocol should prevent untrusted malicious vehicles from joining cluster groups. It is the most basic feature for a cluster-based protocol focusing on security issues to offer a secure cluster creation.

Possible attack 2: A malicious vehicle tries to acquire a cluster group key by sending unwanted valid messages that can incur the congestion in the VANETs. Even though with a secure cluster creation, a protocol prevent malicious vehicles from joining cluster groups and revealing cluster group keys, attacks still can occur. An attacker can attack a VANET by sending out massive useless or incorrect messages to cause a message jam in the communication channel used by the VANETs. The congestions can lead to a serious delay and even loss of package for important useful messages.

Security goal 2: The new protocol should prevent from malicious vehicles that want to participate in the discovery process.

Possible attack 3: A malicious vehicle pretending to be a gateway or a gateway

requester, and sends advertisement messages, or gateway request messages by intercepting, and replaying previously received gateway advertisement messages or request messages. Malicious vehicles can also modify the intercepted discovery messages. That is because these malicious vehicles can attacks the cluster group from inside, That is to say, they are the group members, or once were group member. So that they have the abilities to receive gateway advertisement messages or gateway request messages. Then they can reuse the keys, which are held by them or were once obtained by them, to decrypt previously received messages, change the contents and then encrypt the messages again. Consequently, upon receiving the messages, attackers can store the previously received messages, and alter the information to launch alteration attacks and replay the transmission of the earlier information.

Security goal 3: The new protocol should authenticate the discovery messages to ensure their integrity and that they were sent by authenticated vehicles inside a cluster or gateways.

Possible attack 4: A vehicle sends valid gateway discovery messages that cannot be traced back by authorized law enforcement agencies. This is because an attacker can broadcast false information of itself into the VANETs. It may choose to forge its own information, including its identity, location, speed or other specific information to prevent itself from being discovered out while preforming attacks, such as DoS attacks and replaying attacks. These kind of attacks can continuously occur and offer help to other kinds of attacks, therefore become a serious problem for the VANETs systems.

Security goal 4: The new protocol should prevent non repudiation attack, and allow law enforcement agencies to trace back valid gateway discovery messages.

3.4. The Proposed Secure Gateway Localization and Communication System: SEGAL

This section describes our proposed secure gateway localization and communication protocol (SEGAL) for Vehicular Ad Hoc Networks (VANETs). First, the illustrative examples of SEGAL are shown in 3.3.1. Then, we describe the process of cluster formation, and we explain the vehicle authentication mechanism. After that, we focus on the gateway discovery process and the secure discovery messages exchange. In order to prevent from replay attacks and other malicious attacks, we assume that all the cluster formation messages and the gateway advertisement and discovery messages are timestamped.

3.4.1. Illustrative Example

To describe and explain the main features of our SEGAL protocol briefly, in this session, we will present the illustrative examples in Figure 3.1—3.1

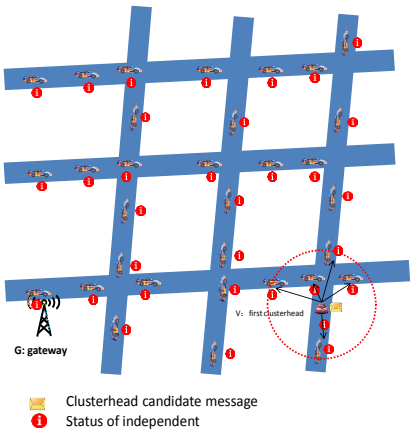

Figure 3.1 Protocol bootstrapping—Vehicle V asks to be a clusterhead

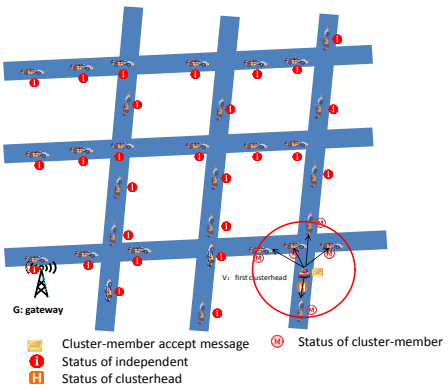

Figure 3.2 Protocol bootstrapping—V accepts those vehicles in its communication range to be its cluster members

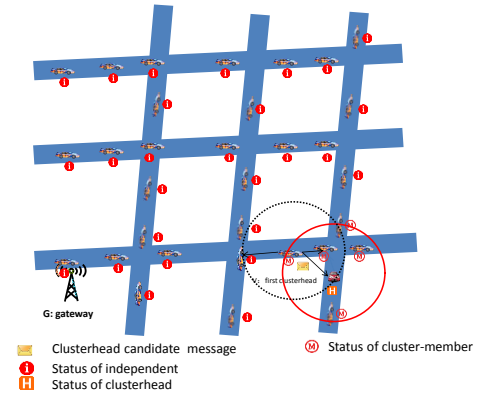

Figure 3.3 Cluster formation—The potential clusterhead sends a clusterhead candidate messages

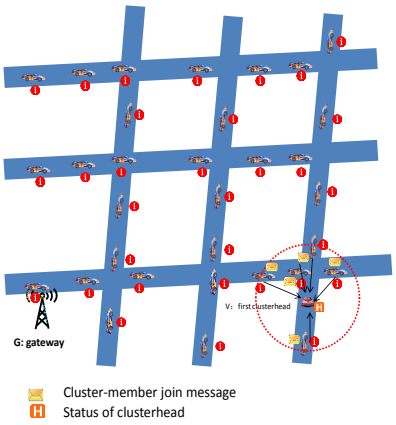

Figure 3.4 Protocol bootstrapping—Vehicles in V's communication range agree to be a cluster member of V

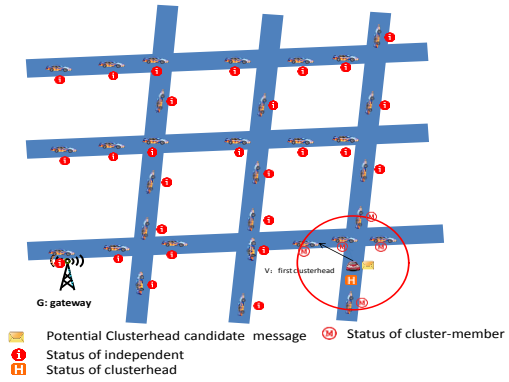

Figure 3.5 Cluster formation — V asks the farthest cluster members to be its neighbor clusterheads

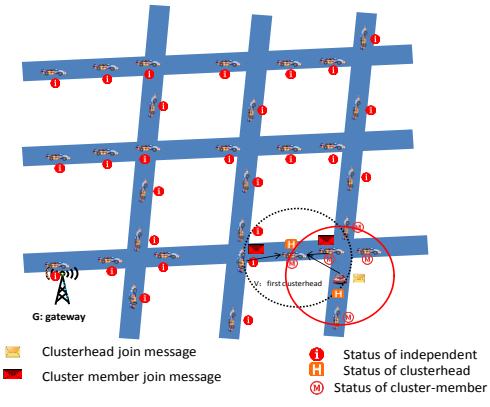

Figure 3.6 Cluster formation—Vehicles request to be cluster members even the neighbor clusterhead

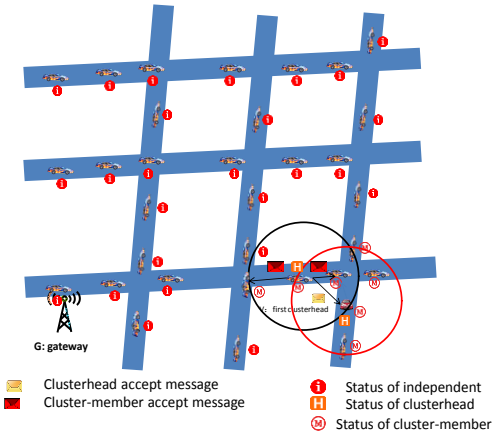

Figure 3.7 Cluster formation—Accept cluster members even neighbor clusterhead

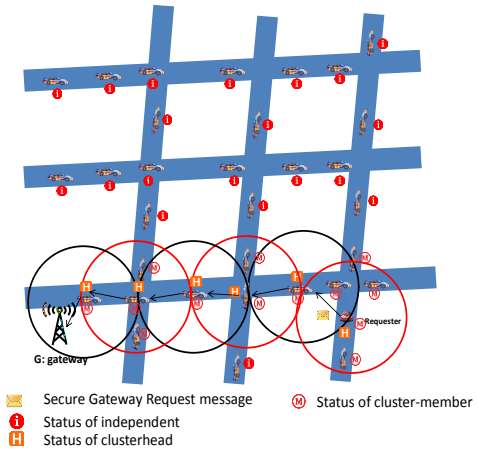

Figure 3.8 Propagation of the Gateway request message

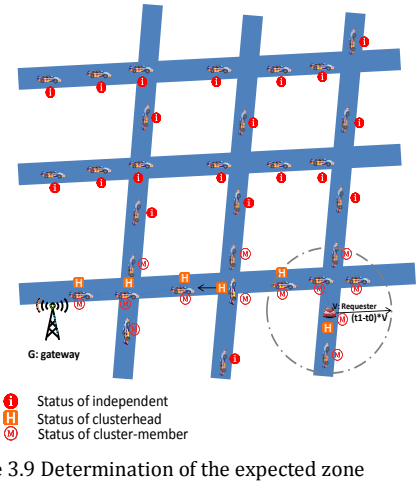

Figure 3.9 Determination of the expected zone of the vehicle V

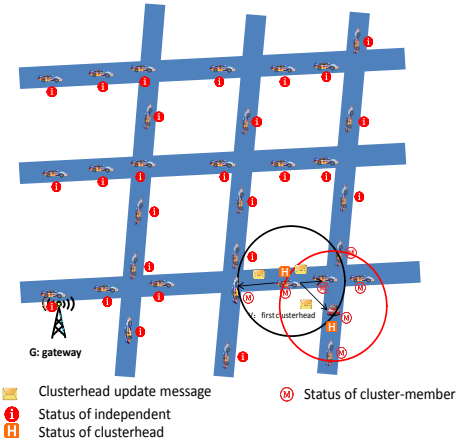

Figure 3.10 Cluster formation—Clusterheads send the clusterhead update messages

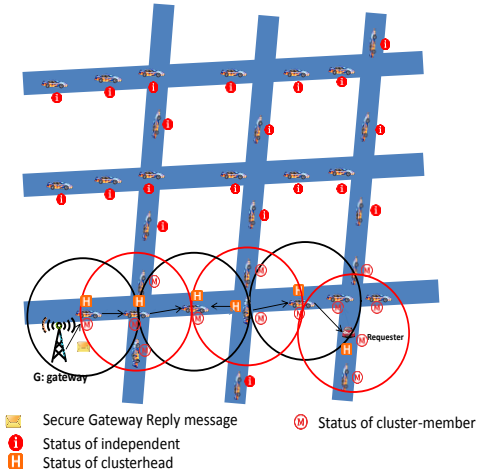

Figure 3.11 Sending the gateway reply message to requesters

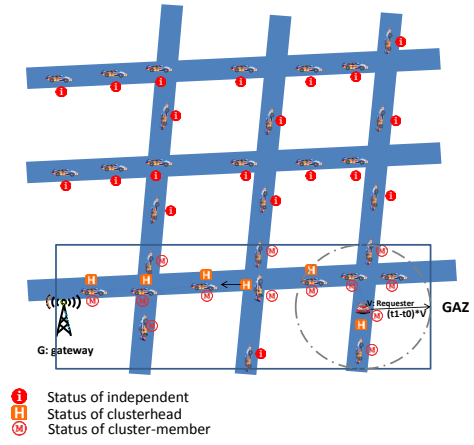

Figure 3.12 Determination of the GAZ of the

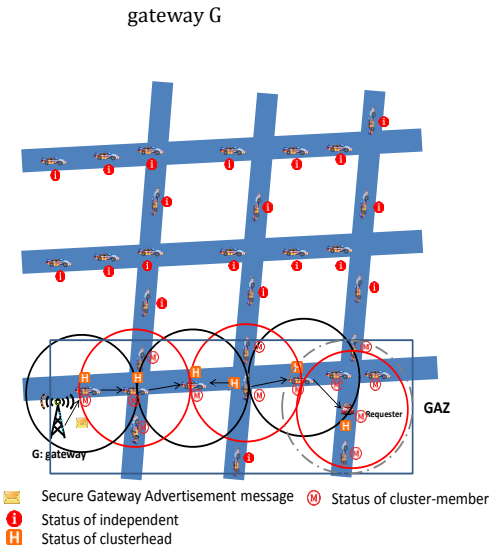

Figure 3.13 Propagation of secure gateway advertisement message

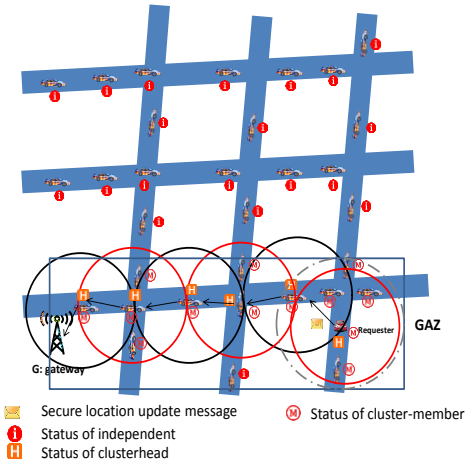

Figure 3.14 Propagation of secure location update message

Execution sequence:

[Seq1 :] Bootstrapping: A independent vehicle V_{ch} asks to be a clusterhead

[Seq2 :] Vehicles in on hop range of V_{ch} agree to join V_{ch}'s cluster.

[Seq3 :] V_{ch} accepts its cluster members and change its status to a clusterhead.

[Seq4 :] V_{ch} invites iis farthest cluster members to be its neighbor clusterhead.

[Seq5 :] The potential clusterhead V_{f1} accepts the invitation of V_{ch} and asks to be a clusterhead.

[Seq6 :] Vehicles in on hop range of V_{f1} including V_{ch} agree to join V_{f1}'s cluster.

[Seq7 :] V_{f1} accepts its cluster members and change its status to a clusterhead. And then V_{f1} invites its farthest cluster members to be its neighbor clusterhead.

[Seq4 :] to [Seq7 :] will be repeated until overlapping clusters are formed in all the VANETs.

[Seq8 :] All the clusterheads in VANETs send clusterhead update messages periodically to maintain their clusters.

[Seq9 :] When a vehicle V needs to query the gateway, it generates a gateway request message and sends it to the gateway. Only vehicles inside the GAZ and nearer the gateway propagate the message. During propagation, only cluster

members of the currently sending clusterhead can decrypt the messages and only the neighbor clusterhead encrypts the message again and sends it to its cluster members. Processions like this repeat until the message reaches the gateway.

[Seq10 :] When the gateway receives a gateway request message, it generates a gateway reply message and sends it to the requester. Only vehicles inside the GAZ and nearer the requester V propagate this message. During the propagation, only cluster members of the currently sending clusterhead could decrypt the messages and only the neighbor clusterhead encrypts the message again and sends it to its cluster members. Processions like this repeat until the message reaches the requester.

[Seq11 :] In the gateway advertisement period, the gateway collects the requester's previous location and speed information, and uses it to determine the expected zone.

[Seq12 :] Gateway determines its GAZ as the minimum rectangular zone contains the expected zone of the requester and the gateway itself.

[Seq13 :] The gateway sends the secure gateway advertisement message. The message is only allowed to propagate inside the GAZ.

[Seq14 :] The gateway collects the location and speed information of the requester at each time interval and uses it to update the determination of its GAZ.

3.4.2. Cluster Formation and Road Components Authentication

In our proposed SEGAL protocol, consecutively overlapping clusters are formed along the VANETs as illustrated in Figure 3.15. Clusterheads are selected so that two consecutive clusterheads are in the communication range of each other. At any time, a road component can be in one of the following statuses: (i)

independent; (ii) *clusterhead*; or (iii) *cluster member*. A clusterhead CH_i of the cluster C_i has to be both a cluster member of the left side and the right side overlapping clusters C_l and C_r. During the cluster formation, cluster group keys are exchanged in a secure manner. The purpose is to have a clusterhead's group public key sent to all its members in a secure manner, which will permit secure gateway advertisement and discovery messages' propagation in the VANETs. We suppose that every vehicle has its pair of private/public keys. We suppose that each vehicle has its identity and public key signed and certified by a certification server. In the following, we describe the behavior of road components under different statuses.

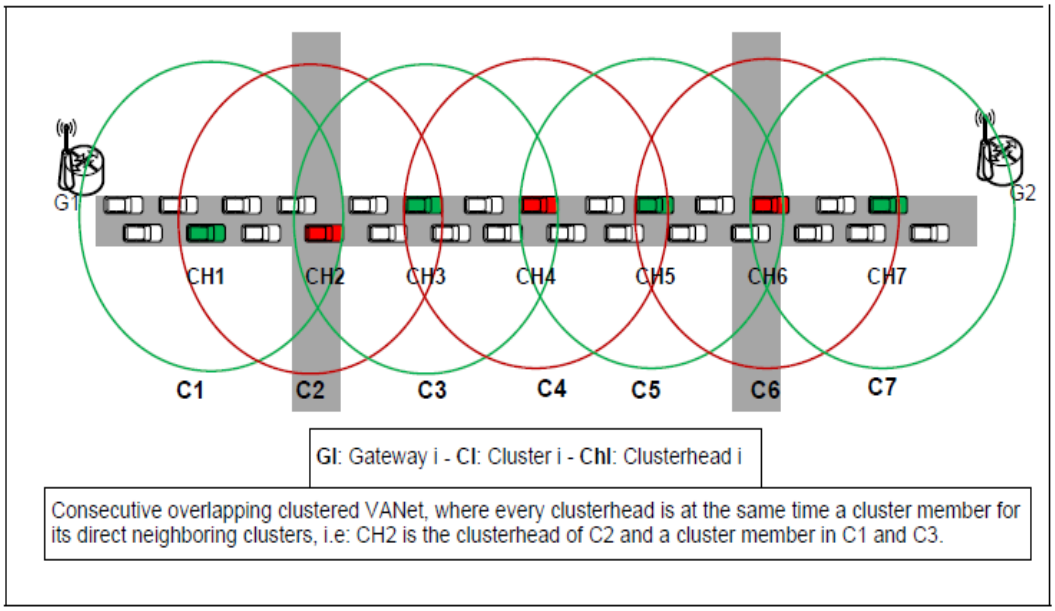

Figure 3.15The clustering process inSEGAL

3.4.2.1 Independent Road Component

At the protocol bootstrapping phase, a vehicle $V_1$with an independent status, starts the clustering process.V_1 sends out a clusterhead candidate message (*CH_Cand_msg*) in its range. The *CH_Cand_msg* is timestamped and contains the location, the speed, and the signed certificate of V_1.

Table 3 Message Types and Descriptions

Message Type	Message Description
CH_Cand_msg	clusterhead candidate message
CM_join_msg	cluster member join message
CM_accept_msg	cluster member accepted message
PCH_Cand_msg	potential clusterhead candidate message
CH_join_msg	clusterhead join message
CH_accept_msg	clusterhead accepted message
Neigh_CH_S	neighboring clusterhead message
CH_up_msg	clusterhead update message
CGKP	Cluster Group private/public Key Pair
CGPuK	Cluster Group Public Key
CGPrK	Cluster Group Private Key
SAdv_msg	secure advertisement message
SGReq_msg	secure gateway request message
SGRep_msg	secure gateway reply message
SLocUp_msg	secure location update message

Any independent vehicle V_i that receives the *CH_Cand_msg*, decrypts the certificate of V_1 with the public key of the certification server and retrieves the sender ID and public key. The V_i stores the ID of V_1, its location, its speed and its public key in its clusterhead information Table as illustrated in Table 5. V_i then generates a cluster member join message (*CM_join_msg*). The *CM_join_msg* is timestamped and contains the location, the speed, and the signed certificate of V_i. V_i sends the *CM_join_msg* to the vehicle V_1.

Table 4 Cluster Member Table

ID	ID_1	ID_i
Location information	(x_1, y_1)	(x_i, y_i)
Velocity	Vel_1	Vel_i
Public key	PK_1	PK_i
Neighboring clusterhead status	Yes/No	Yes/No
Neighboring clusterhead Cluster Group Public Key	$CGPuK_1$	$CGPuK_i$

At the reception of a *CM_join_msg*, V_1 decrypts the certificate with the public key of the certification server to retrieve the ID and the public key of V_i. V_1 stores the ID, the location, the speed, and the public key of V_i in its cluster member table as illustrated in Table 4.V_1then changes its status to clusterhead, and generates a Cluster Group Key Pair (CGKP). The CGKP contains a Cluster Group Public Key (CGPuK), and a Cluster Group Private Key (CGPrK). The V_1generates a cluster member accepted message (*CM_accept_msg*) that contains its CGPuK. The *CM_accept_msg* is signed withV_1 private key and encrypted with V_i public key for security and authentication purposes. V_1then sends the (*CM_accept_msg*) to V_i.

An independent vehicle V_i that receives a *CH_accept_msg* from a vehicle V_1 after sending a *CM_join_msg* to V_1, decrypts the received message, changes its status to cluster member and storesV_1's CGPuK in its clusterhead information table.

Table 5 **Algorithm 3.4.2.1** Protocol at an Independent Road Component V_{in} in Cluster Formation Processes

1: **Case A.** Bootstrap:

2: Step A0. V_{in} broadcasts a CH_Cand_msg containing V_{in} 's signed certificate in one hop.

3: **Case B.**V_{in} receives a CH_Cand_msg from V_i:

4: **if** Step B0. V_i is registered with a certification server **then**

5: Step B0.0. V_{in} decrypt the CH_Cand_msg by the PuK of certification server.

6: Step B0.1. V_{in} stores V_i's PuK into its clusterhead table.

7: Step B0.2. V_{in} generates a CM_join_msg containing V_{in} 's signed certificate.

8: Step B0.3. V_{in} sends a CM_join_msg to V_i

9: **end if**

10: **Case C.** V_{in} receives a message CM_join_msg from V_i:

11: **if** Step C0. V_i is registered with a certification server **then**

12: Step C0.0. V_{in} decrypts the CM_join_msg by the PuK of certification server.

13: Step C0.1. V_{in} changes its status to a clusterhead.

14: Step C0.2. V_{in} stores V_i's PuK into its cluster member table.

15: Step C0.3. V_{in} generates its CGKP.

16: Step C0.4. V_{in} generates a CM_accept_msg containing its CGPuK.

17: Step C0.5. V_{in} signs the CM_accept_msg with V_{in}'s PrK.

18: Step C0.6. V_{in} encrypts the CM_accept_msg with V_i's PuK.

19: Step C0.7. V_{in} sends the CM_accept_msg to V_i.

20: **end if**

21: **Case D.** V_{in} receives a CM_accept_msg from V_{ch} containing the CGPuK of V_{ch}:

22: Step D0. V_{in} decrypts the CM_accept_msg with V_{ch}'s PuK.

23: Step D1. V_{in} changes its status to a cluster member of V_{ch}.

24: Step D2. V_{in} stores V_{ch}'s CGPuK into its clusterhead information table.

3.4.2.2 Clusterhead Road Component

A vehicle V_1 that receives a *CM_join_msg* from V_i, decrypts the certificate with the public key of the certification server to retrieve the ID and the public key of V_i.V_1 stores the ID, the location, the speed, and the public key of V_i in its cluster member table. V_1then sends the (*CM_accept_msg*) to V_i containing the CGPuK of V_1. The *CM_accept_msg* is signed with V_1's private key and encrypted with V_i's public key. V_1 waits a period of time to receive further *CM_join_msg*s from potential cluster members. Then, supposing that the VANET is modeled in an orthogonal system, V_1 sends a potential clusterhead candidate message (*PCH_Cand_msg*) to the most distant 2 vehicles V_{f1} andV_{f2} among its cluster members in its direction and in both senses. The *PCH_Cand_msg* sent to V_{f1} is signed with the private key of V_1 and encrypted with the public key of V_{f1}. Similarly, the *PCH_Cand_msg* sent to V_{f2} is signed with the private key of $V_1$and encrypted with the public key of V_{f2}. These 2 vehicles,V_{f1} andV_{f2} would be the potential clusterheads of the consecutive right and left sides clusters of C_1. V_{f1} and V_{f2} will still be cluster members in C_1, even if they become clusterheads of the side clusters of V_1.Moreover, V_1 would also be a cluster member in the clusters ofV_{f1} and V_{f2}, respectively. For this purpose, when V_1 hears the *CH_Cand_msg* from V_{f1} and V_{f2} respectively, V_1 sends a clusterhead join message (*CH_join_msg*) to its neighboring clusterhead V_{f1} signed with V_1's private key and encrypted withV_{f1}'s public key;V_1sends another *CH_join_msg* to V_{f2} signed with V_1's private key and encrypted with V_{f2}'s public key. This permits V_1 to be added to the cluster member table of V_{f1} andV_{f2} respectively, but it is added with the following special neighboring clusterhead status: *Neigh_CH_S*.

Upon receiving of the *CH_join_msg* from V_1, V_{f1} decrypts the received message

and adds V_1 to its cluster members table with the status *Neigh_CH_S*; moreover, it sends a signed and encrypted clusterhead accepted message (*CH_accept_msg*) to V_1 containing the CGPuK of V_{f1}. V_1 is at the same time a clusterhead of its cluster and a cluster member in the cluster of V_{f1}.

At the reception of the clusterhead accepted message (*CH_accept_msg*) from V_{f1}, V_1 decrypts the received message, and adds V_{f1} to its cluster member table with the status *Neigh_CH_S*.V_1 adds as well the CGPuK of V_{f1} to its cluster members table. Clusterheads periodically send clusterhead update messages (*CH_up_msg*) to their cluster members, encrypted with their appropriate CGPrK.

Table 6 **Algorithm 3.4.2.2** Protocol at the Clusterhead V_{ch} in Cluster Formation Processes

1:	**Case A**. V_{ch} receives a message CM_join_msg from V_i:
2:	**if** Step A0. V_i is registered with a certification server **then**
3:	Step A0.0. V_{ch} decrypts the CM_join_msg by the PuK of certification server.
4:	Step A0.1. V_{ch} changes its status to a clusterhead.
5:	Step A0.2. V_{ch} stores V_i's PuK into its cluster member table.
6:	Step A0.3. V_{ch} adds V_i to its cluster member.
7:	Step A0.4. V_{ch} generates its CGKP.
8:	Step A0.5. V_{ch} generates a CM_accept_msg containing its CGPuK.
9:	Step A0.6. V_{ch} signs the CM_accept_msg with V_{ch}'sPrK.
10:	Step A0.7. V_{ch} encrypts the CM_accept_msg with V_i'sPuK.
11:	Step A0.8. V_{in} sends the CM_accept_msg to V_i.
12:	**end if**
13:	Step A1. Waits a period of time to receive further CM_join_msg from potential cluster members.

14: Step A2. V_{ch} identifies its farthest cluster members V_{fr} and V_{fl}.

15: Step A3.V_{ch} generates two PCH_Cand_msgs.

16: Step A4. V_{ch} signs the PCH_Cand_msgs with V_{ch}'sPrK.

17: Step A5. V_{ch} encrypts the PCH_Cand_msg with V_{fr}'s and V_{fl}'sPuK respectively.

18: Step A6. V_{ch} sends PCH_Cand_msgs to V_{fr} and V_{fl} respectively.

19: **Case B**. V_{ch} receives messages CH_Cand_msg from V_{fr} orV_{fl}:

20: Step B0. V_{ch} generates two CH_join_msgs.

21: Step B1. V_{ch} signs the CH_join_msgs with V_{ch}'s PrK.

22: Step B2. V_{ch} encrypts the CH_join_msgs with V_{fr}'s and V_{fl}'s PuK respectively.

23: Step B3. V_{ch} sends CH_join_msgs to V_{fr} and V_{fl} respectively.

24: **Case C.** V_{ch} receives a CH_join_msg from V_{fr}or V_{fl}

25: Step C0. V_{ch} changes its status to a clusterhead.

26: Step C1. V_{ch} adds V_{fr} orV_{fl} to its cluster member in status of Neigh_CH_S.

27: Step C2. V_{ch} generates its CGKP.

28: Step C3. V_{ch} sends a CH_accept_msg to V_{fr} or V_{fl} containing its CGPuK.

29: Step C4. V_{ch} sends a CM_accept_msg in one hop containing its CGPuK.

30: **Case D**. V_{ch} receives a CH_accept_msg from V_{fr} or V_{fl}:

31: Step D0. V_{ch} decrypts the CH_accept_msg by its own PrK.

32: Step D1. V_{ch} stores *sender's (V_{fr}* or *V_{fl})* CGPuK into its cluster member table.

33: Step D2. V_{ch} adds *sender (V_{fr}* or *V_{fl})* to its cluster member with the status Neigh_CH_S.

34: **Case E.** V_{ch} maintains its cluster group.

35: **while** Step E0.*time* equals to the *TIME INTERVAL* **do**

36: Step E0.0.V_{ch} generates a CH_up_msg containing its CGPuK.

37: Step E0.1.V_{ch} encrypts the CH_up_msg with its CGPrK.

38: Step E0.2.V_{ch} sends the CH_up_msg to all its cluster members.

39: Step E0.3.Set time as zero.

40: Step E0.4.Start timer.

41: **end while**

42: **Case G**. V_{ch} receives messages CH_up_msg from V_{fr} or V_{fl}:

43: Step G0.V_{ch} decrypts the CH_up_msg with the sender's (V_{fr}'s or V_{fl}'s) CGPuK.

44: Step G1.V_{ch} stores the sender's (V_{fr}'s or V_{fl}'s) CGPuK in its clusterhead table.

3.4.2.3 Cluster Member Road Component

A cluster member CM_1that receives a *PCH_Cand_msg* from the clusterhead CH_1 decrypts it and sends out a *CH_Cand_msg* that contain its certificate in its range. Upon receiving of a *CM_join_msg* from a vehicle V_j, the vehicleCM_1 decrypts the received message, retrieves the ID, the location, the speed, and the public key of V_j, and then stores them in its cluster members table. CM_1 changes its status to clusterhead, generates a CGKP and sends a cluster member accepted message (*CM_accept_msg*)to V_j containing its CGPuK. The *CM_accept_msg* is signed withCM1's private key and encrypted with V_j's public key for security and authentication purposes.

A cluster member that does not receive a clusterhead update message (*CH_up_msg*) from its CH for 2 consecutive update periods and is in the *Neigh_CH_S* status, sends a signed and encrypted *PCH_Cand_msg* to its farthest cluster member in the direction of the previous clusterhead.

Table 7 Clusterhead Information Table

Clusterhead ID	ID_j
Clusterhead Location information	(x_j, y_j)
Clusterhead Velocity	Vel_j
Clusterhead Public key	PK_j
Clusterhead Cluster Group Public Key	$CGPuK_j$

Table 8 **Algorithm 3.4.2.3** Protocol at the Cluster Member V_{cm} in Cluster Formation Processes

1: **Case A.** V_{cm} receives a PCH_Cand_msg:

2: Step A0. V_{cm} decrypts the PCH_Cand_msg with its own PrK.

3: Step A0. V_{cm} sends a CH_Cand_msg containing its certificate in one hop.

4: **Case B.** V_{cm} receives a message CM_join_msg from V_i:

5: **if** Step B0. V_i is registered with a certification server **then**

6: Step B0.0. V_{cm} decrypts the CM_join_msg by the PuK of certification server.

7: Step B0.1. V_{cm} storesV_i's PuK into its cluster member table.

8: Step B0.2. V_{cm} changes its status to a clusterhead.

9: Step B0.3. V_{cm} generates its CGKP.

10: Step B0.4. V_{cm} generates a CM_accept_msg containing its CGPuK.

11: Step B0.5. V_{cm} signs the CM_accept_msg with V_{cm}'s PrK.

12: Step B0.6. V_{cm} encrypts the CM_accept_msg with V_i's PuK.

13: Step C0.7. V_{cm} sends the CM_accept_msg to V_i.

14: **end if**

15: **Case C.** Neigh_CH_S maintenance

16: **if** Step C0. (V_{cm} is in status of Neigh_CH_S and does not receive CH_up_msg for 2 consecutive *TIME INTERVAL* **then**

17: Step C0.0. V_{cm} identifies its farthest cluster members V_{fr} and V_{fl}.

18: Step C0.1.V_{cm} generates two PCH_Cand_msgs.

19: Step C0.2.V_{cm} signs the PCH_Cand_msgs with V_{cm}'s PrK.

20: Step C0.3.V_{cm} encrypts the PCH_Cand_msg with V_{fr}'s and V_{fl}'s PuK respectively.

21: Step C0.4.V_{ch} sends PCH_Cand_msgs to V_{fr} and V_{fl} respectively.

22: **end if**

3.4.3. Gateway Discovery

Our proposed SEGAL is a secure hybrid adaptive gateway localization and communication system. It is considered as a hybrid because gateways advertise themselves in a specific area, and requesters send their gateway requests out until they reach the gateway advertisement zone. SEGAL is adaptive because the advertisement zone of the gateway adapts to the changing number and location of gateway requesters.

In the following, we present our proposed secure gateway localization and communication protocol. First, we explain the secure gateway advertisement process. Second, we describe the secure gateway request propagation schema. Third, we discuss the secure gateway reply propagation mechanism.

3.4.3.1 Secure hybrid Adaptive Gateway Advertisement

Initially, the clustering process is performed. We suppose that the gateways are trusted and that they are deployed with the same pair of public/private keys. We suppose also that vehicles have retrieved the common gateways' public key from a certification server. Gateways advertise themselves by sending secure timestamped gateway advertisement messages (*SAdv_msg*) out in their respective ranges. The *SAdv_msg* is encrypted with the gateway's private key. Vehicles in the communication range of the gateway receiving the *SAdv_msg* decrypt it with the gateway's public key. Vehicles then store the gateway information in their gateway information table.

Upon receiving of a secure gateway request message (*SGReq_msg*), the gateway retrieves the requesting vehicle's location information$L_0(x_0, y_0)$ and velocity vel_0 at the instant t_0. The retrieved information permits the gateway to adapt its

advertisement zone to the expected location of the requester in the next advertisement period at the instant t_1. The gateway determines the expected location of the requesting vehicle as the disc centered at$L_0(x_0, y_0)$ and having the radius $radius_1$computed as:

$$radius_1 = (t_1 - t_0).vel_0(1)$$

Then, the gateway determines its advertisement zone as the minimal rectangular zone that includes the gateway and the predetermined expected zone of the requesting vehicle. The gateway generates its secure location based advertisement message *SAdv_msg* comprising the coordinates of the advertisement zone and encrypts it with its private key. Then, it sends out it towards the expected zone of their questing vehicle.

Table 9. **Algorithm 3.4.3.1.1** Protocol at the Gateway G_i in Secure Gateway Advertisement Processes

1:	**Case A.** Bootstrap:
2:	Step A0. G_i generates messages *Sadv_ msg.*
3:	Step A1. G_i encrypts with its PrK.
4:	Step A2. G_i broadcasts messages *Sadv_ msg* periodically in on hop.
5:	**Case B.** G_i receives a gateway request message *SGReq_ msg* from V_{req}:
6:	Step B0. G_i retrieves V_{req}'s information.
7:	Step B1. G_i determines *its GAZ.*
8:	Step B2. G_i generates SAdv_msg containing the GAZ information.
9:	Step B3. G_i encrypts SAdv_msg with its PrK.
10:	Step B4. G_i sends SAdv_msg towards the expected zone of V_{req}.
11:	**Case C.** G_i receives a message *Sloc_up_msg* from V_{req} containing V_{req}'s updated location information:
12:	Step C0. G_i retrieves V_{req}'s updated information.
13:	Step C1. G_i adapts *its GAZ.*

Vehicles in the communication range of the gateway receive the *SAdv_msg* and decrypt it with the gateway's public key and they store the gateway information in their respective gateway tables. A clusterhead CH_i that receives the *SAdv_msg* determines whether or not it is inside the gateway advertisement zone (*GAZ*). If CH_i is inside the *GAZ*, it encrypts the *SAdv_msg* with its *CGPrK* and sends it.

Table 10. **Algorithm 3.4.3.1.2** Protocol at the Clusterhead V_{ch} in Secure Gateway Advertisement Processes

1:	**Case A.** V_{ch} receives a SAdv_msg:
2:	**if** Step A0. SAdv_msg is sent by gateway G_i **then**
3:	Step A0.0. V_{ch} decrypts the message with G_i*'s PuK*.
4:	Step A0.1. V_{ch} stores the gateway information in its gateway information table.
5:	**If** Step A0.2. V_{ch} is inside the *GAZ* **then**
6:	Step A0.2.0. V_{ch} encrypts the SAdv_msg with its *CGPrK*.
7:	Step A0.2.1. V_{ch} sends SAdv_msg in one hop.
8:	**end if**
9:	**else if** Step A1. SAdv_msg is sent by its neighbor clusterhead V_{f1} **then**
10:	Step A1.0. V_{ch} decrypts the message with *CGPuK* of V_{f1}.
11:	Step A1.1. V_{ch} stores the gateway information in its gateway information table.
12:	**if** Step A1.2. V_{ch} is inside the GAZ **then**
13:	Step A1.2.0. V_{ch} encrypts the SAdv_msg with its *CGPrK*.
14:	Step A1.2.1. V_{ch} sends SAdv_msg in one hop.
15:	**end if**
16:	**end if**
17:	**Case B**. V_{ch} receives a SLocUp_msg:
18:	**if** Step B0. the SLocUp_msg is sent by a *cluster member* **then**

19: Step B0.0. V_{ch} decrypts the message with its *CGPrK*.

20: Step B0.1. V_{ch} encrypts the message with *CGPrK*.

21: Step B0.2. V_{ch} sends SAdv_msg in one hop towords the G_i

22: **else if** Step B1. the SLocUp_msg is sent by a *clusterhead* **then**

23: **if** Step B1.0 V_{ch} is closer to gateway than the sender **then**

24: Step B1.0.0 V_{ch} decrypts the message with its *CGPuK*.

25: Step B1.0.1. V_{ch} encrypts the message with *CGPrK*.

26: Step B1.0.2. V_{ch} sends SAdv_msg in one hop towords the G_i .

27: end if

28: end if

All the cluster members of CH_i, including the left side clusterhead CH_l and the right side clusterhead CH_r of CH_i, decrypt the received *SAdv_msg* with the *CGPuK* of CH_i. The cluster members store the gateway information in their gateway tables. The clusterheads CH_l and CH_r that receive the gateway advertisement message for the first time determine whether or not they are inside the *GAZ*. The clusterhead that is inside the *GAZ* encrypts the *SAdv_msg* with its *CGPrK* and sends it.

At the reception of the *SAdv_msg* by the requesting vehicle (V_{Req}), the *SAdv_msg* is decrypted with the *CGPuK* of the clusterhead of V_{Req}. Then, V_{Req} updates its gateway information table, and sends out a secure location update message (*SLocUp_msg*) to its clusterhead CH encrypted with the CH's CGPuK. The clusterhead CH decrypts the received *SLocUp_msg*, then encrypts it with its *CGPrK*. The *SLocUp_msg* is propagated towards the gateway. Only clusterheads that are closer to the gateway than the sending one, continue the propagation of the *SLocUp_msg*. Each time the *SLocUp_msg* is received by a clusterhead CH, it is decrypted with the sending clusterhead's *CGPuK* and encrypted with the

current clusterhead's *CGPrK*. The *SLocUp_msg* is propagated until it reaches the gateway. It would permit to the gateway to adapt its *GAZ*.

Table 11. **Algorithm 3.4.3.1.3** Protocol at the Cluster Member V_{cm} in Secure Gateway Advertisement Processes

1:	**Case A.** V_{cm} receives a SAdv_msg:
2:	**if** Step A0. SAdv_msg is sent by gateway G_i **then**
3:	Step A0.0. V_{cm} decrypts the message with G_i*'s PuK*.
4:	Step A0.1. V_{cm} stores the gateway information in its gateway information table.
5:	**else if** Step A1. SAdv_msg is sent by neighbor clusterhead V_{f1} **then**
6:	Step A1.0. V_{cm} decrypts the message with *CGPuK* of V_{f1}.
7:	Step A1.1. V_{cm} stores the gateway information in its gateway information table.
8:	**end if**
9:	**if** Step A2. V_{cm} is the V_{Req} **then**
10:	Step A2.0. V_{cm} generates a SLocUp_msg containing its new information.
11:	Step A2.1. V_{cm} encrypts the SLocUp_msg with its *clusterhead's CGPuK*
12:	Step A2.2. V_{cm} sends a SLocUp_msg to its *clusterhead.*
13:	**end if**

3.4.3.2 Secure Gateway Request and Reply Propagation

A gateway requester vehicle V_{Req} that needs to find a gateway, generates a secure timestamped gateway request message (*SGReq_msg*), and sends it encrypted to its clusterhead CH_i with CH_i's public key. The clusterhead CH_i decrypts the *SGReq_msg* with its private key and encrypts it with its *CGPrK*, then it sends it. All the cluster members of CH_i can decrypt the message with the *CGPuK* of CH_i. Each neighboring clusterhead CH_n, encrypts the *SGReq_msg* with its correspondent *CGPrK* and sends it. The *SGReq_msg* continue its secure

propagation by the clusterheads until it reaches the advertisement zone of the gateway.

Table 12. **Algorithm 3.4.3.2.1** Protocol at the Clusterhead V_{ch} in Secure Gateway Request and Reply propagation

1:	**Case A.** V_{ch} receives a SGReq_msg:
2:	**if** Step A0. The *sender* is the *requester* itself **then**
3:	Step A0.0. V_{ch} decrypts the message with its *PrK*.
4:	Step A0.1. V_{ch} encrypts the message with *CGPrK*.
5:	Step A0.2. V_{ch} sends SGReq_msg in one hop towords the G_i.
6:	**else if** Step A1. The sender is neighbor clusterhead V_{f1} **then**
7:	**if** StepA1.0. V_{ch} is not inside the *GAZ* **then**
8:	Step A1.0.0. V_{ch} decrypts the message with its *CGPuK*.
9:	Step A1.0.1. V_{ch} encrypts the message with *CGPrK*.
10:	Step A1.0.2. V_{ch} sends SGReq_msg in one hop towords G_i.
11:	**end if**
12:	**end if**
13:	**Case B.** V_{ch} receives a SGReq_msg:
14:	**if** Step B0. The SGReq_msg is sent by a *vehicle* inside *GAZ* **then**
15:	Step B0.0. V_{ch} decrypts the SGReq_msg with its *PrK*.
16:	Step B0.1. V_{ch} encrypts the SGReq_msg with its *CGPrK*.
17:	Step B0.2. V_{ch} sends the SGReq_msg toward the requesting vehicle.
18:	**else if** Step B1. The SGReq_msg is sent by a *clusterhead* outside GAZ **then**
19:	**if** Step B1.0. V_{ch} is closer to the V_{Req} than the *sender* **then**
20:	Step B1.0.0. V_{ch} decrypts the SGReq_msg with its *CGPuK*.
21:	Step B1.0.1. V_{ch} encrypts the SGReq_msg with its *CGPrK*.

22: Step B1.0.2. V_{ch} sends the SGReq_msg toward the *requesting vehicle*.

23: **end if**

24: **end if**

At the reception of a *SGReq_msg*, the gateway or a vehicle inside the *GAZ* generates a secure gateway reply message (*SGRep_msg*). The *SGRep_msg* contains the expected location of the gateway requester. Then, the gateway or the replying vehicle encrypts the *SGRep_msg* with the public key of the clusterhead CH and sends it to CH. The clusterhead CH, decrypts the *SGRep_msg* with its private key. After this, CH encrypts the *SGRep_msg* with its *CGPrK* and sends it toward the requesting vehicle. The *SGRep_msg* is received by the cluster members, as well as the clusterheads of the neighboring clusters. It is decrypted with the *CGPuK* of the sending clusterhead. Then, only clusterheads that are closer to the expected location of the requesting vehicle can forward the *SGRep_msg* after it is encrypted with the current clusterhead's *CGPrK*.

Table 13. **Algorithm 3.4.3.2.2** Protocol at the Clusterhead V_{cm} in Secure Gateway Request and Reply propagation

1: **Case A.** V_{cm} needs to request G_i:

2: Step A0. V_{cm} generates a SGReq_msg.

3: Step A1. V_{cm} encrypts the SGReq_msg with its *clusterhead's PuK*.

4: Step A2. V_{cm} sends the SGReq_msg to its *clusterhead*.

5: **Case B.** V_{cm} receives a SGReq_msg.

6: **If** Step B0. V_{cm} is inside the *GAZ* **then**

7: Step B0.0. V_{cm} generates a SGRep_msg containing the location of G_i.

8: Step B0.1. V_{cm} encrypts the SGRep_msg with the *PuK* of its *clusterhead*.

9: Step B0.2. V_{cm} sends out the SGRep_msg to its *clusterhead*.

10: **end if**

11: **Case C.** V_{cm} receives the SGRep_msg:

12: **If** Step C0. V_{cm} is the requester **then**

13: Step C0.0. V_{cm} decrypts the SGReq_msg with its *CGPuK*

14: Step C0.1. V_{cm} stores the G_i's information in its gateway information table

15: Step C0.2. V_{cm} starts its communication with G_i

16: **end if**

At the reception of the *SGRep_msg* by the requesting vehicle V_{Req}, the *SGRep_msg* is decrypted and the gateway information is retrieved and stored in the gateway information table of S_{Req}. Then, the vehicle S_{Req} starts its communication with the gateway.

3.5. SEGAL Security Analysis

In the following, we will provide the analysis of security of our SEGAL protocol. The security analysis of our proposed SEGAL protocol will be proved in 4 steps: First, we will prove that our SEGAL protocol prevents malicious vehicles from joining the cluster group. Second, we will prove that our protocol prevents attackers from modifying and replaying the previously received gateway advertisement messages or request messages. Third, we will prove that our proposed protocol prevents malicious vehicles from pretending to be a gateway or gateway requester. In the end, we will prove that Our SEGAL protocol prevents from non-repudiation attacks, and let law enforcement agencies able to trace back valid gateway discovery messages.

Lemma 1.(Secure Cluster Creation)*Our proposed SEGAL protocol prevents a malicious vehicle not registered with a certification server from joining a cluster group.*

Proof:

Every vehicle is assumed to have its identity verified by a certification server before it is issued a valid private/public key pair. For example, if a node *X* is a trusted vehicle registered with a certification server, it sends an *CH_Cand_msg* in one hop. And there is a malicious vehicle *Y* who is not registered with a certification server but wants to become a cluster member of *X*. Then *Y* sends *CM_join_msg* to *X*. In our proposed SEGAL, vehicles exchange their certified public keys before the creation of the clusters. Since malicious vehicle *Y* cannot be issued valid public and private key pair from a certification server, it cannot exchange the key pair with *X* and cannot be accepted by *X* as its cluster member. Consequently, a malicious vehicle not registered with a certification server cannot participate in the cluster formation process.

Moreover, in our SEGAL protocol the clusterhead exchanges its *CGPuK* with its cluster members in a secure way, using its members' respective public keys. For example, the malicious vehicle *Y* mentioned above wants to reveal the *CGPuK* of a secure clusterhead *X*. However, when *X* exchanges its *CGPuK* with its cluster member by sending the messages *CM_accept_msg*, the messages are encrypted with the receivers' respective *PuK*. Because *Y* cannot be assigned *PuK* and *PrK* pair, it does not have abilities to reveal the *CM_accept_msg* and get the *CGPuK* of *X*. Thus, only verified cluster members can acquire the clusterhead's *CGPuK* and be members in the formed cluster.

Thus, our SEGAL protocol permits clusters formed in a secure way.

Lemma 2. (Discovery Messages Authentication and Integrity) *Our proposed SEGAL protocol prevents malicious vehicles from modifying and replaying the discovery messages.*

Proof:

The exchange of gateway discovery messages in SEGAL is performed through the clusterheads. Every clusterhead encrypts the discovery message with its *CGPrK*. For example, a malicious vehicle *Y* receives a discovery message sent by a clusterhead *X* to its cluster member *Z*. Because *X* encrypted the message by its *CGPrK* and *Y* does not have the *CGPuK* of *X's* cluster group, *Y* cannot decrypt the message and change its content. Therefore, *Y* cannot make attacks to the authentication of the discovery message. As a consequence, only aclusterhead's cluster members including its neighboring clusterheads can decrypts the discovery message with the sending clusterhead's *CGPuK*.

Moreover, in our SEGAL protocol, every discovery message has its own timestamp, which makes the discovery message effective only in a certain and short time interval. For example, an attacker *Y* stored the previous received discovery message. After a period of time, *Y* tries to replay this message. However, this message is expired and loss effectiveness now because of its timestamp. So *Y* cannot make replay attacks to VANETs. As a consequence, our SEGAL can prevent from replay attacks.

Thus, our SEGAL protocol prevents attackers from modifying and replaying the previously received gateway advertisement messages or request messages.

Lemma 3. (Prevention from Compromised Vehicles) *Our SEGAL protocol prevents attackers from pretending to be a gateway or gateway requester.*

Proof:

In our proposed SEGAL, we assumed that gateways have their own common certified pair of private/public key assigned by certification servers. Since malicious vehicles are not registered with certification servers, they cannot acquire the certified key pairs from certification servers. Consequently, without the certified key pair, an attacker cannot pretend to be a gateway.

Moreover, when a requester in SEGAL system sends a *SGReq_msg* to its clusterhead, it encrypts the message with the clusterhead's public key. Upon receiving this message and before sending it out, its clusterhead decrypts the message with its private key. Since attackers are not be assigned public/private key pairs, the gateway request messages sent by them will not be accepted by any clusterhead. Thus, any malicious vehicle cannot initiate or participate in a discovery process.

Consequently, a compromised vehicle can pretend to be a gateway or a gateway requester and start sending gateway advertisement messages or gateway request messages.

Lemma 4. (Non-Repudiation) *Our SEGAL protocol prevents non-repudiation attacks, and allows law enforcement agencies to trace back valid gateway discovery messages.*

Proof:

In our proposed SEGAL, vehicles' have their own public keys which are certified by a certification server. These public keys are related to their real identities respectively in their certificates. When law enforcement agencies need to trace back a valid gateway discovery message, they can get the real identity of the sender by checking the signed certificate from certification servers. For

example, a law enforcement agency *A* wants to trace a gateway discovery message *SGReq_msg*. Because this message *SGReq_msg* is signed with its *sender's PrK*, *A* can ask certification servers to reveal *sender's* real *ID* to *A*. Certification servers then check their certification table and search the corresponding *sender's PuK* to get the relative real *ID* by using the signed *PrK*. Finally, the *sender's* real *ID* will be sent to *A*.

Thus, law enforcement agencies can trace back any sender of gateway discovery messages.

Chapter 4

Performance Evaluation of SEGAL Protocol

In this section, the performance evaluation of our SEGAL protocol is presented. In the first setting, we compare SEGAL protocol with the current gateway discovery scheme, which is the Location Aided Hybrid Adaptive Gateway Advertisement and Discovery LAGAD protocol for VANETs [48] . In the second setting, we evaluate the performance of SEGAL in different scenarios while varying different metrics.

4.1. Compare with LAGAD Scheme

LAGAD, is a gateway discovery scheme based on an adaptive advertisement zone, but does not consider the security aspects. In our proposed SEGAL, we modified the gateway discovery design to support security aspects and prevent from malicious attacks. To the best of our knowledge, our proposed protocol is the first gateway discovery protocol for VANETs that considers securing gateways discovery and communication. For this reason, we compare SEGAL to LAGAD in order to prove how it is important to prevent requests from malicious attacks during the discovery process. We evaluate also the performance of SEGAL and prove its scalability and efficiency.

4.1.1. Experiments Setup

We use the network simulation NS2 to conduct our experiments. A Manhattan traffic model is considered to represent the VANET. Weuse realistic mobility traces to simulate the movement of vehicles inside the VANET. In our simulation, we supposed that there are malicious vehicles that try to replay advertisement messages and gateway request messages.Table 14 shows the parameter values that have been used in our simulations.

In this experiment, we varied the number of gateway requests from 10 to 100, we considered that the average vehicle speed is 50 km/h, we assumed that the average per vehicle density is 40, and we used the following performance metrics in order to compare SEGAL to LAGAD:

- Success rate: indicates the average fraction of successful gateway discovery requests over the total number of gateway requests initiated during the simulation time.
- Total bandwidth usage: measures the total number of bits exchanged in the VANETs for all the gateway discovery queries initiated by gateway requesters during the simulation time.
- Average gateway request response time: measures the average time required for a gateway requester to receive a valid gateway reply after the initiation of the gateway discovery process.

In addition, we fixed the number of gateway requests to 100, then, we varied the per vehicle density from10 to 150, and the average per vehicle speed from 20 Km/h to 90Km/h to measure the following metric:

- Average message dropping ratio: calculates the percentage of dropped gateway discovery messages that were not able to reach their

correspondent destinations.

Table 14 Simulation Parameters in Comparison Experiments

Parameter Name	Parameter Value
Wireless medium	802.11
Data transmission rate	11Mbps
Transmission range(meters)	200
Received signal strength threshold (meters)	200
Vehicle's Speed (meter/second)	[0..20]
Gateway advertisement's life time (seconds)	50
Simulation Time (seconds)	1500
Number of gateways	4
Simulation area ($meter^2$)	4,203,848
Average vehicle's density	200
Number of gateway clients'	40
Average vehicles' speed	50km/h
Average per vehicle density	40

In our simulation, we supposed that there are malicious vehicles that try to replay advertisement messages and gateway request messages. We run our simulations many times, and the interval of confidence for the reported results is [90% -95%].

4.1.2. Experiment Results

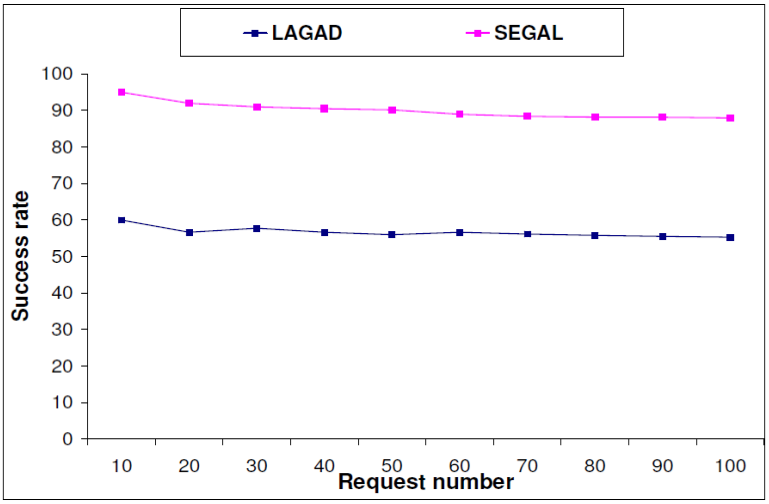

Figure 4.1Success rate comparison of the SEGAL to the LAGAD

Figure 4.1 plots the graphs related to the success rate comparison of our proposed SEGAL to the LAGAD protocol in a Manhattan traffic model when varying the number of requests from 10 to 100. Our proposed SEGAL achieves more than 90%for successful gateway discovery transactions. However, the LAGAD protocol achieves less than 60% in terms of success rate. The reason behind the low success rate for LAGAD is mainly related to the fact that the malicious vehicles in the VANETs dropped or modified some of the received gateway discovery messages. Thus, gateway requesters in LAGAD were not able to have their requests propagated till the gateway. In other scenarios, the gateway reply messages were not propagated toward the requesting vehicles in LAGAD. In opposite, in our proposed SEGAL, all the gateway discovery messages exchanged is authenticated and not altered. In SEGAL, gateway advertisement and gateway request and reply messages are propagated in a secure way.

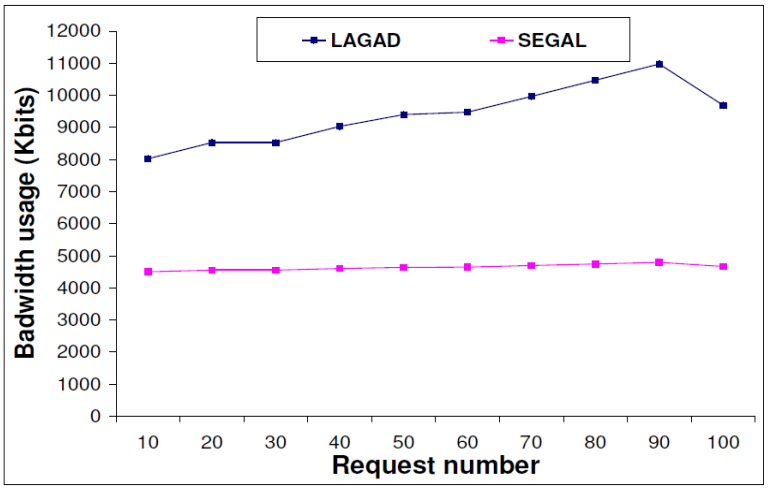

Figure 4.2Bandwidth consumption comparison of the SEGAL to the LAGAD

Figure 4.2 portrays the total bandwidth usage comparison of SEGAL to the LAGAD when the number of requests ranges between10 and 100. The total bandwidth usage of SEGAL is almost halved compared to the total bandwidth usage of the LAGAD protocol. This is mainly due to the fact that gateway advertisement and discovery messages in SEGAL are exchanged through the clusterheads. This saved the bandwidth used for SEGAL and proved its scalability, even if the SEGAL needs to exchange extra messages during the clustering process.

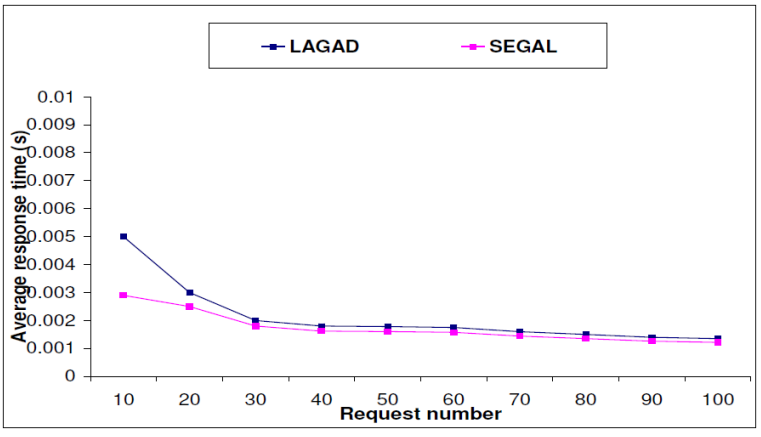

Figure 4.3Average gateway discovery delay comparison of the SEGAL to the LAGAD

Figure 4.3 shows the curves related to the average gateway discovery delay comparison of SEGAL to the LAGAD for a number of requests ranging between

10 and 100. Both SEGAL and LAGAD achieve an average low response time for the gateway discovery, with even a bit lower performance of SEGAL compared to LAGAD. This is mainly due to the fact that in the SEGAL, the discovery messages are exchanged through the clusterheads in the VANETs. Thus, even if SEGAL requires more processing time to decrypts and encrypts each time the gateway discovery messages, it is lower than the LAGAD protocol in terms of average response time because the communication in SEGAL is performed through the clusterheads only.

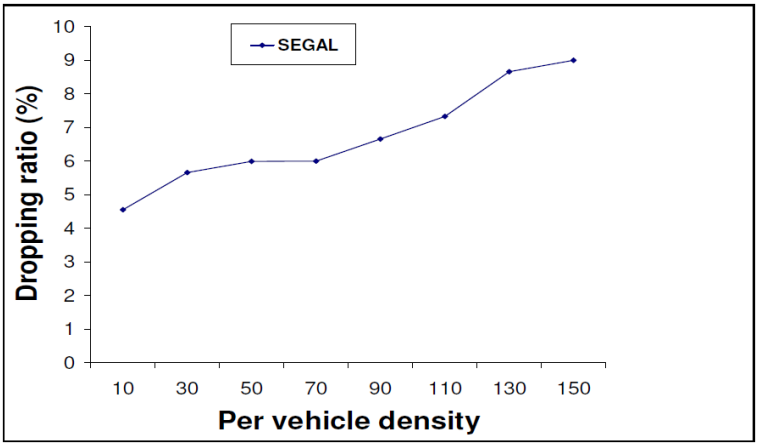

Figure 4.4Average gateway requests dropping rate of SEGAL for different per vehicle densities

Figure 4.4 shows the average gateway requests dropping rate of SEGAL for different per vehicle densities ranging from 10 to150. Per vehicle density refers to the number of vehicles in the communication range of a vehicle. As shown is Figure 4.4, with the increase of the per vehicle density, the dropping ratio grows up slowly to reach 10% as maximum, which proves the scalability of SEGAL. The main reason is that the heavy traffic may lead to a congestion of the communication packets between vehicles, making the gateway discovery messages lost on the way. Another reason is that the congestion-made delay may cause the serious trouble to the connection of the gateway advertisement packets.

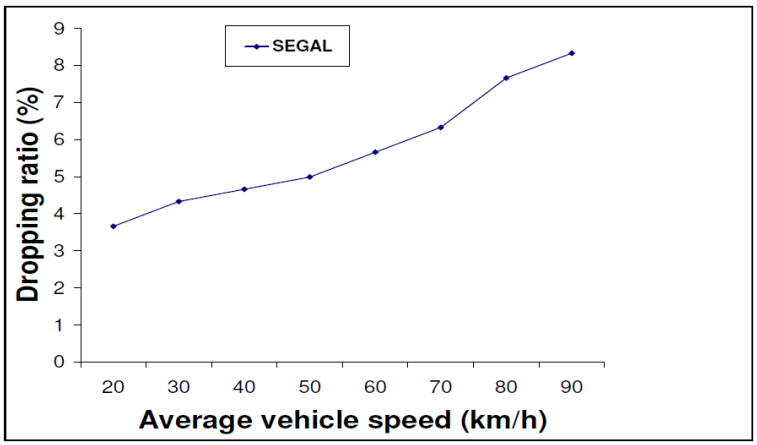

Figure 4.5Average gateway requests dropping rate of SEGAL for different vehicles' speed

Figure 4.5 plots the average gateway requests dropping rate of SEGAL for different vehicles speed ranging from 20 km/h to 90km/h. The average gateway dropping rate rises gradually as the vehicles speed increases. However, it does not exceed 9%, which proves again the scalability of SEGAL. This is mainly due to the fact that the higher is the speed, the weaker is the stability of clusters. Thus, gateway discovery messages could be dropped in their way to their respective destinations.

4.2. SEGAL: Performance Evaluation Study

In this part, we simulate the movement of vehicles in VANETs in different scenarios, namely, the city scenario and the highway scenario. We evaluate the performance of the SEGAL protocols in the following metrics: success ratio, average response delay and total bandwidth usage. In each scenario, we conduct one of the three parameters, namely, density, speed and the request number, as a variable and keep the other two the same. In this way, we want to show the performance of our SEGAL protocol objectively.

4.2.1. Experiments Setup

We used the NS2 simulator to simulate a Manhattan city model and a highway traffic model with realistic mobility traces, ensuring the validity and accuracy of the results obtained in our experiments. The parameters adopted in these experiments are shown in Table 15. In the simulation of this part, we set the simulation area as 1000,000m^2. In these set of experiments, there were 4 roads in both the highway scenario and the city scenario. Two roads were straight lane roads and the other two roads were perpendicular lanes. The length of each road is 1000 meters. We supposed that there were 10 malicious vehicles in the simulation area trying to drop advertisement messages, gateway requests and reply messages received without forwarding them.

Table 15 Parameters in City and Highway Scenarios

Parameter Name	Parameter Value
Wireless medium	802.11
Data transmission rate	11Mbps
Transmission range(meters)	200
Received signal strength threshold (meters)	200
Gateway advertisement's life time (seconds)	50
Simulation Time (seconds)	900
Number of gateways	4
Simulation area ($meter^2$)	1000,000
Number of gateway clients'	40
Number of malicious nodes	10

In both city and highway scenario, three metrics were chosen to reflect the performance of SEGAL, which are the metrics of success rate, total bandwidth

and average gateway request response time (average latency),in the experiments with the request number variable, the speed variable and the density variable respectively.

City scenario: In the experiments using the request number as a variable, we fixed the average speed of vehicles at40km/h and fixed the density of the traffic model to30/km, which means there were 30 vehicles in each 1000 meters road lane, and varied the number of requests from 100 to 800.In the experiments using the speed as a variable, we set the number of requests to 100 and set the density to30/km and varied the speed from 30 km/h to 50 km/h. In experiments using vehicles' density as a variable, we set the number of requests to 100 and set the average speed of the vehicles to 40km/h and varied the traffic density from 10/km to 50/km.

Highway scenario: In the experiments using the request number as a variable, we set the speed of vehiclesat80km/h and we set the density of the traffic model as 10/km and varied the number of requests from 100 to 800. In the experiments using the speed as a variable, we set the number of requests as 100 and set the density as 10/km and varied the speed from 80 km/h to 120 km/h. In experiments using vehicles' density as a variable, we set the number of requests to 100 and set the vehicles' speed at80km/hand varied the traffic density from 2/km to 16/km.

In our simulation, we supposed that there are malicious vehicles that try to replay advertisement messages and gateway request messages. We run our simulations many times, and the interval of confidence for the reported results is [90% - 95%].

4.2.2. Experiment Results

Now let us turn to the results obtained in both the city scenario and the highway scenario, which are summarized in Figures[4.6-4.23].

Success Ratio: indicates the average fraction of successful gateway discovery requests over the total number of gateway requests initiated during the simulation time.

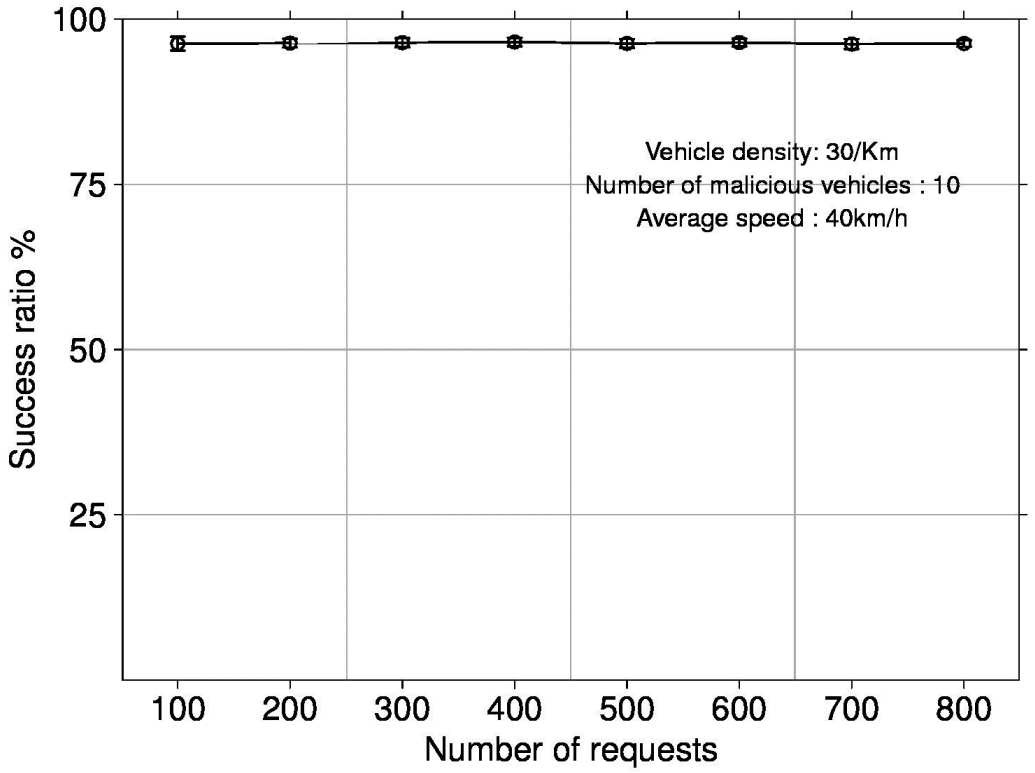

Figure 4.6Success ratio for different numbers of requests in the city scenario

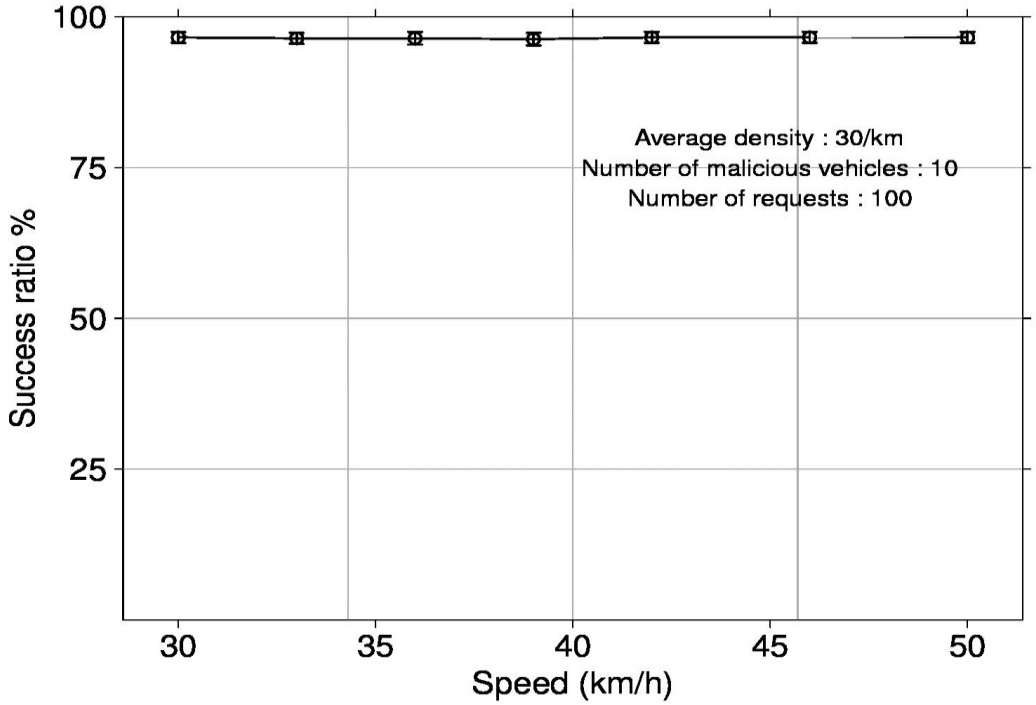

Figure 4.7 Success ratio for different average speeds in the city scenario

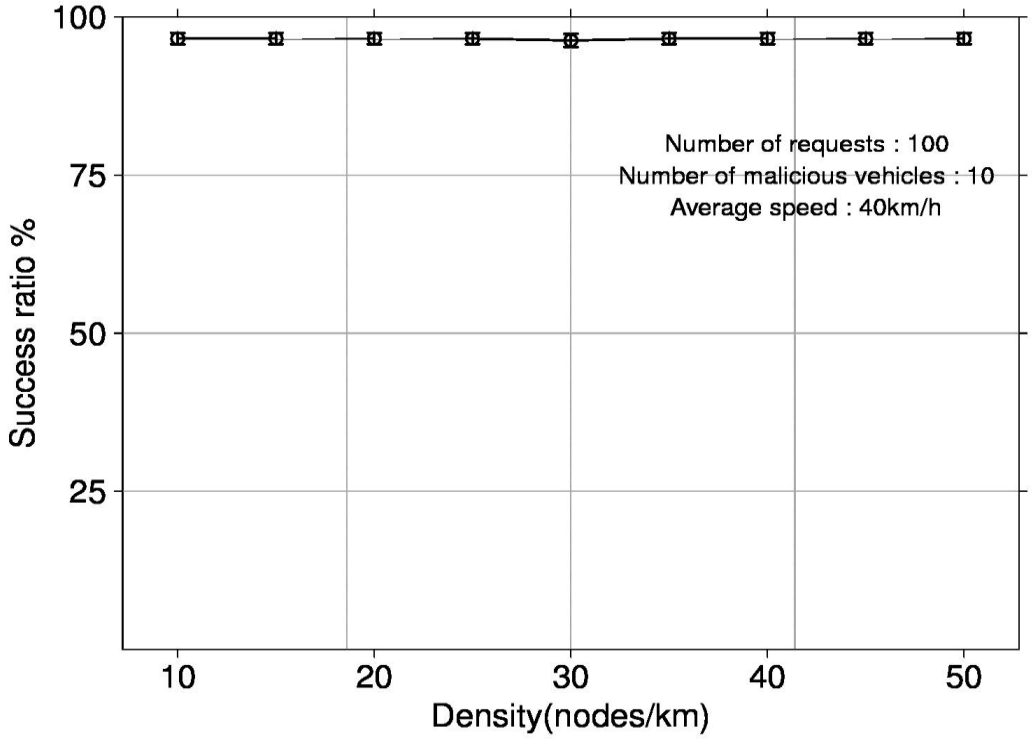

Figure 4.8 Success ratio for different levels of density in the city scenario

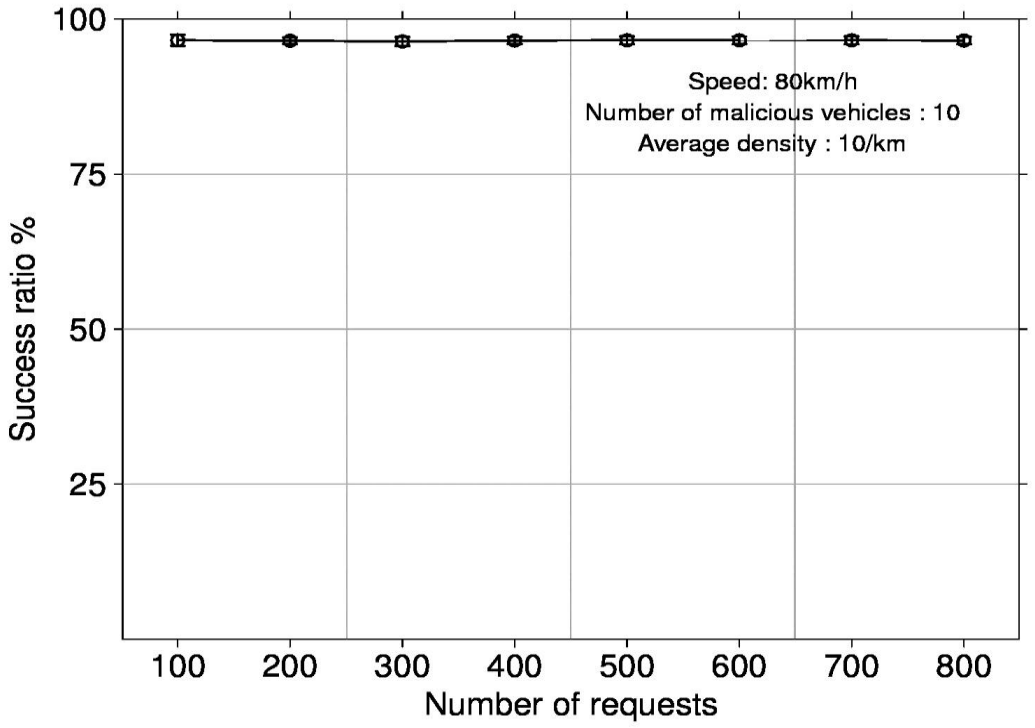

Figure 4.9 Success ratio for different numbers of requests in the highway scenario

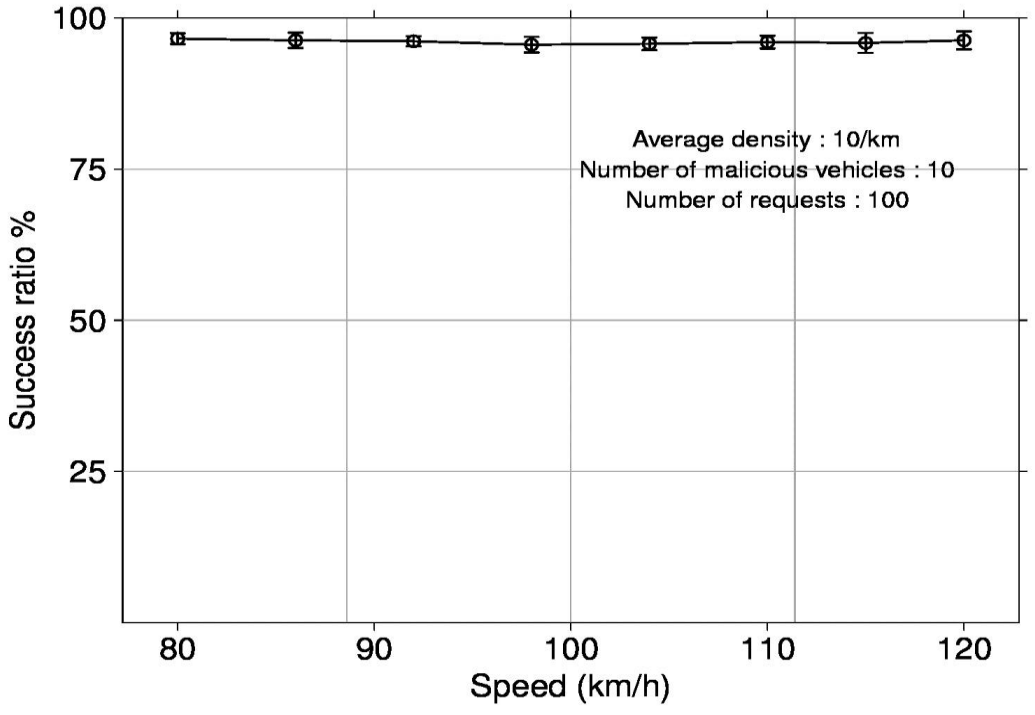

Figure 4.10 Success ratio for different average speeds in the highway scenario

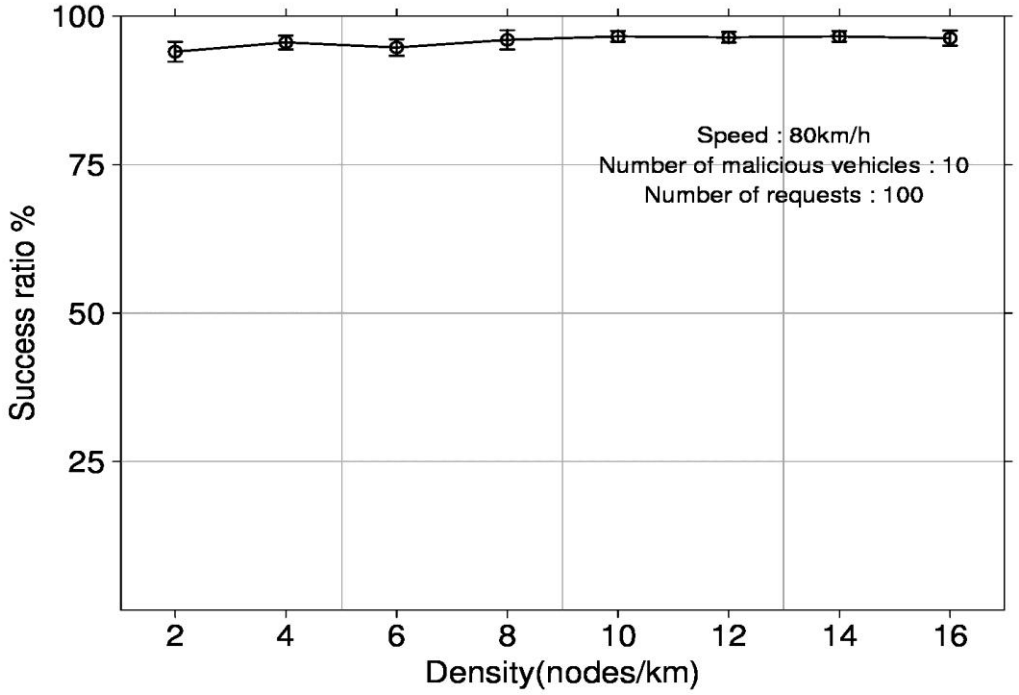

Figure 4.11 Success ratio for different levels of density in the highway scenario

Figures [4.6-4.8] show the results of success ratios obtained from experiments using the number of requests, the average speed and the level of density as variables respectively in the city scenario. Figures [4.9-4.11] show the results of success ratios obtained from experiments using the number of requests, the average speed and the level of density as variables respectively in the highway scenario. As Figure 4.6 to Figure 4.11 show, a high level in the success rate can be achieved by our SEGAL protocol; the success rate is more than 96 percent, even when there are 10 malicious nodes that try to drop messages they received without forwarding them. That is because our SEGAL protocol is an efficient and secure gateway discovery protocol. SEGAL protocol adopts the mechanism of an adaptive gateway advertisement zone. This mechanism can quickly adapt to the current network, as the same time determine and maintain the gateway advertisement zone, ensuring that the different number of requests, the different mobility of vehicles, the different vehicles' density and the network topology

cannot fail into the communication of gateway discovery messages. In addition, the SEGAL applies the cluster-based mechanism. The creation of clusters is undergone in a secure way and messages are only exchanged between secure clusterheads and their authenticated cluster members; therefore, malicious nodes cannot join the process of gateway discovery. The results shown in Figure 4.6 to Figure 4.11 prove the stability and reliability of the SEGAL protocol.

Total Bandwidth Usage: measures the total number of bits exchanged in the VANET for all the gateway discovery queries initiated by gateway requesters during the simulation time.

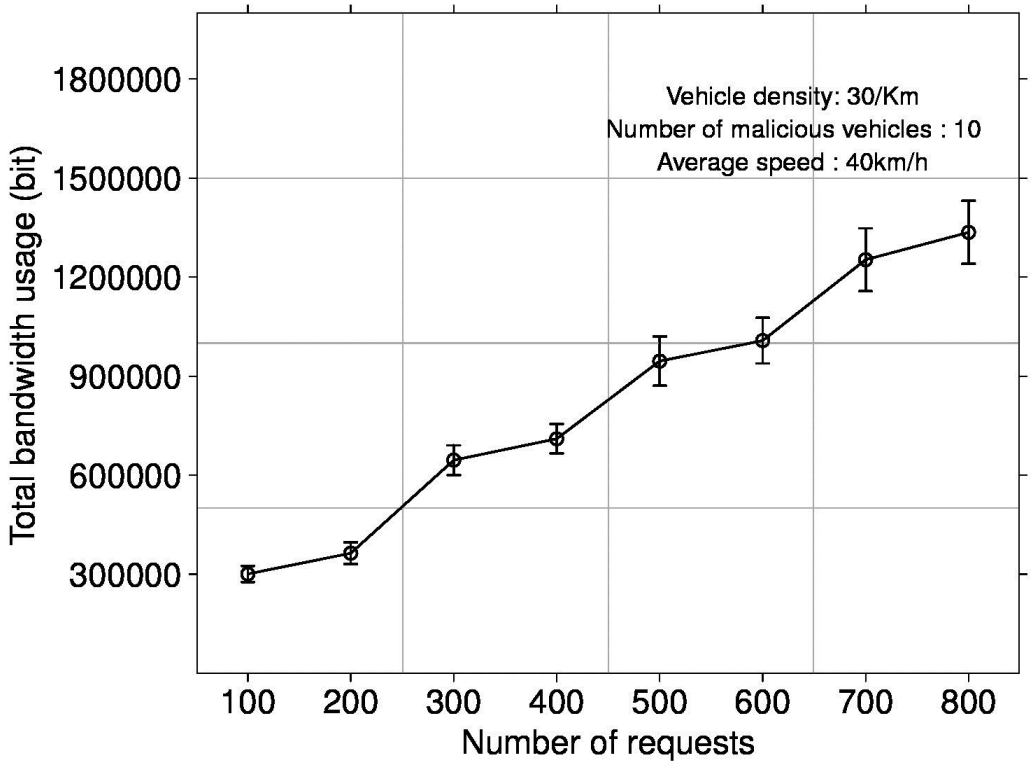

Figure 4.12The total bandwidth usage for different numbers of requests in the city scenario

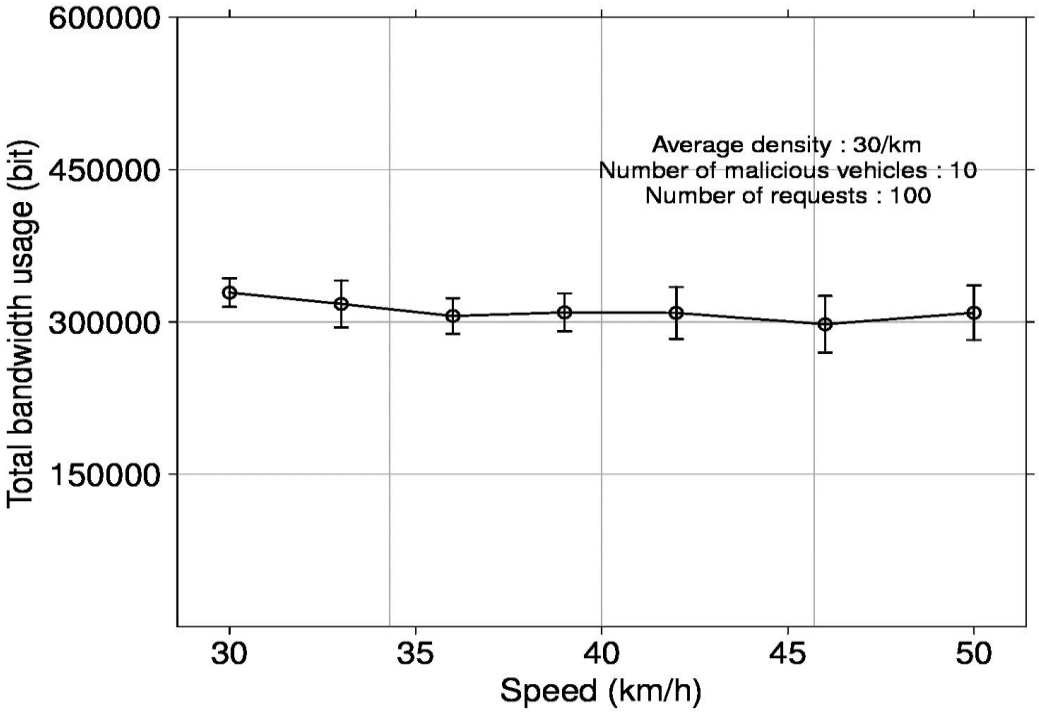

Figure 4.13The total bandwidth usage for different average speeds in the city scenario

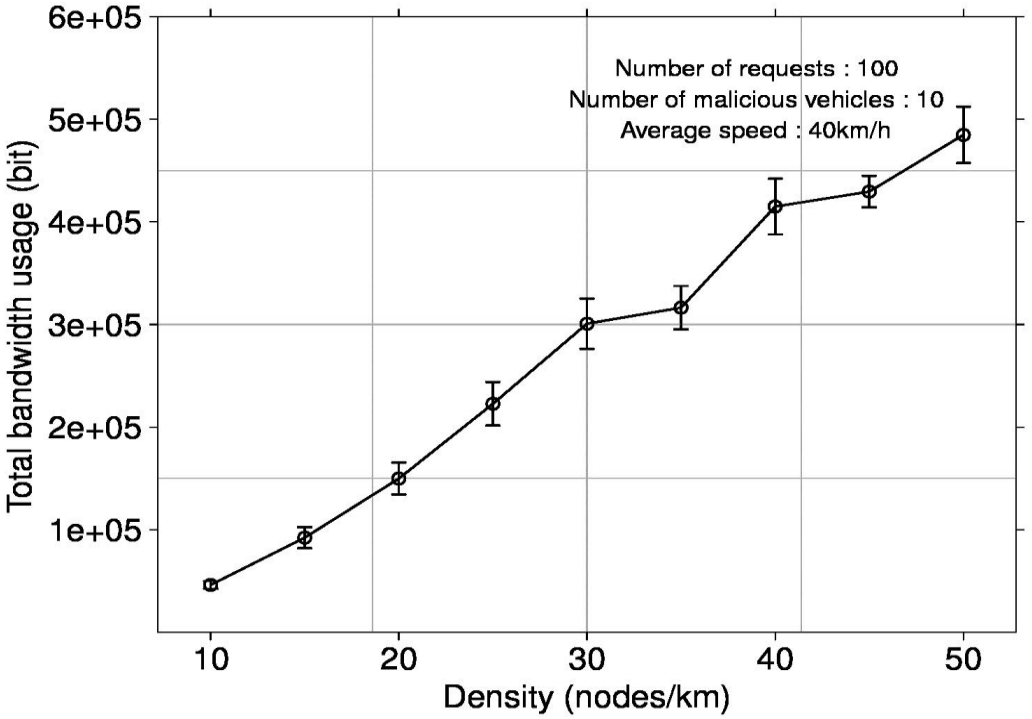

Figure 4.14The total bandwidth usage for different levels of density in the city scenario

Figure 4.12-Figure 4.14] plot the total bandwidth usage of gateway request process in SEGAL protocol. The results were obtained from experiments which use the number of requests, the average speed and the levels of density as variables in the city scenario. As we can see in Figure 4.12 and Figure 4.14, with the increase in the number of requests or density in the city scenario, the total bandwidth usage of a gateway request process climbs up. In Figure 4.12, when the number of gateway requests increases from 100 to 800, the total bandwidth usage rises up gradually from 300,000 bits to above 1,350,000 bits. In Figure 4.14, the total gateway discovery bandwidth usage increases from about 40,000 bits to a maximum of just below 500,000 bits as the density grows up from 10/km to 50/km. This phenomenon is mainly due to the fact that both the larger number of gateway requests and the higher traffic density lead much easily to message congestion. Therefore, message congestion will cause the message to be stuck and will require it to be re-sent. This leads to high bandwidth usage. Differently, Figure 4.13 shows the total gateway bandwidth usage stays stable at around 300,000 bits as the vehicles' speed increases from 30km/h to 50 m/s. This demonstrates the scalability of SEGAL. The stability presented here should be attributed to the adoption of the adaptive gateway advertisement zone method in the SEGAL protocol, which ensure that messages are only exchanged through clusterheads.

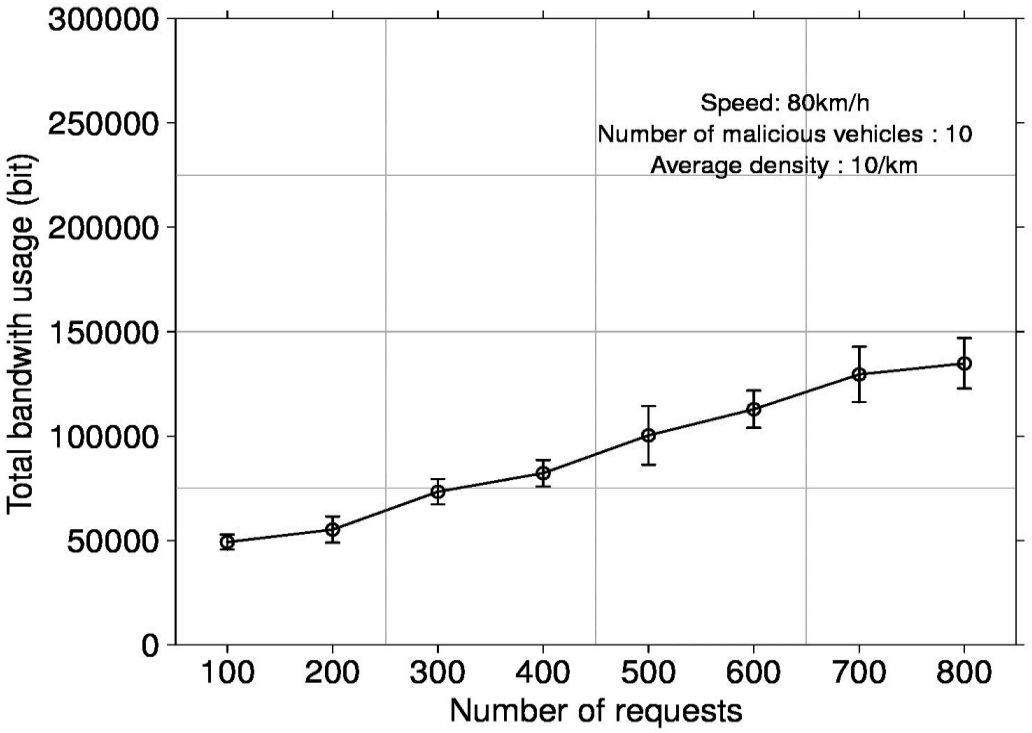

Figure 4.15The total bandwidth usage for different numbers of requests in the highway scenario

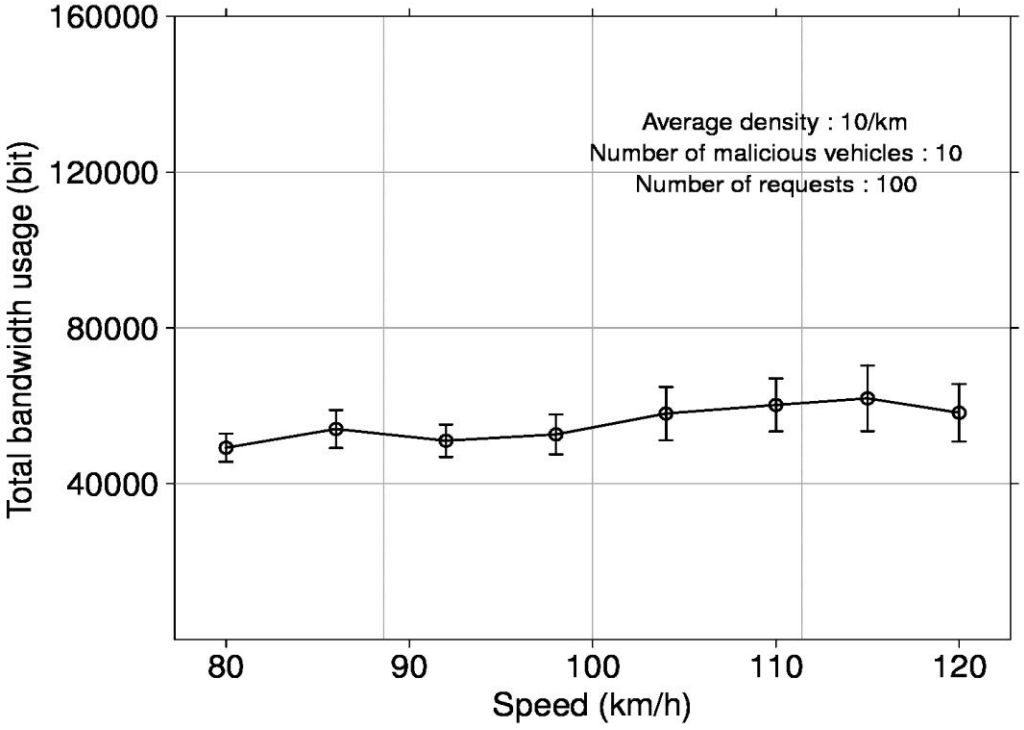

Figure 4.16The total bandwidth usage for different average speeds in the highway scenario

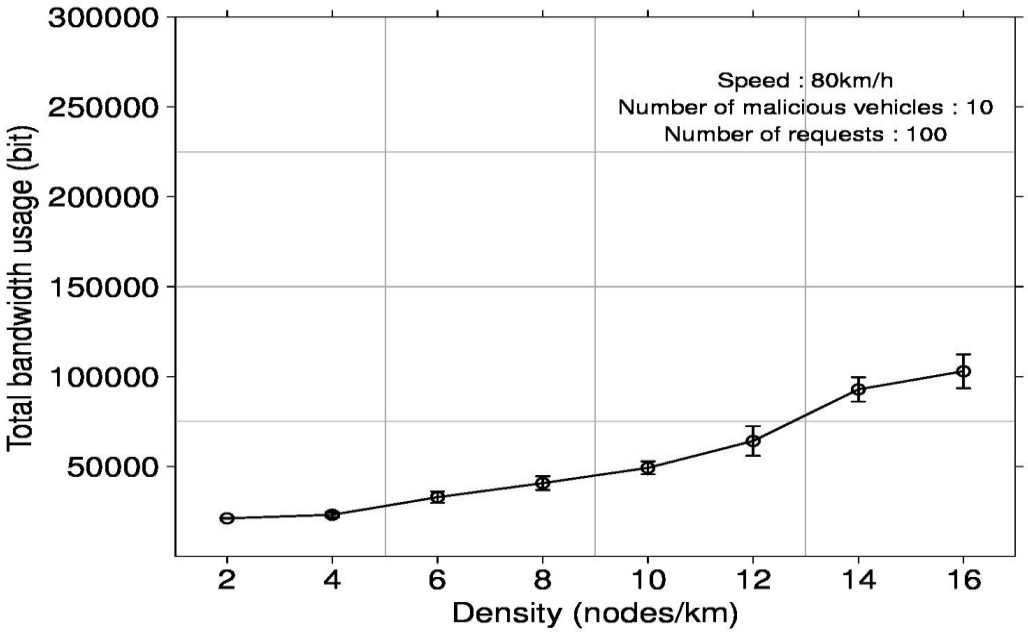

Figure 4.17The total bandwidth usage for different levels of density in the highway scenario

Figure 4.15-4.17] show the total bandwidth usage of gateway request processes in the SEGAL protocol for different numbers of requests, different average speeds, and for different levels of density in the highway scenario. Figure 4.15 and Figure 4.17 show a rise in total bandwidth usage with an increase in the number of requests or density in the highway scenario. In Figure 4.15, when the number of requests increases from 100 to 800, the total bandwidth usage increases rapidly from 50,000 bits to above 140,000 bits. However, in Figure 4.17, the total gateway discovery bandwidth usage grows significantly to a maximum of just above 100,000 bits; this occurs as the density increases from 2/km to 16/km. This is mainly because of the message congestion caused by the larger number of requests and the higher traffic density. However, Figure 4.16 shows the total bandwidth usage of SEGAL for different average speed ranging from 80km/h to 120km/h in highway scenario. The total gateway bandwidth usage grows slightly from 50,000 bits to 60,000 bits as vehicles' speed increases. The stable trend is thanks to the usage of the adaptive gateway advertisement zone mechanism in SEGAL protocol.

Average latency: measures the average time required for a gateway requester to receive a valid gateway reply after the initiation of the gateway discovery process.

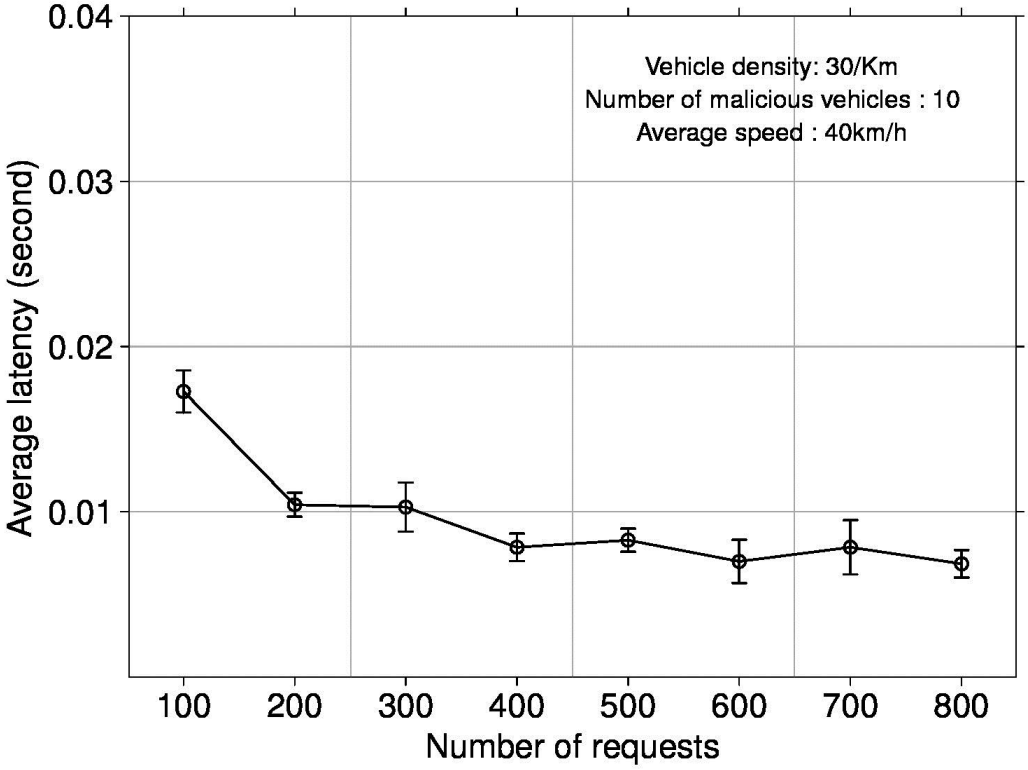

Figure 4.18The average latency for different numbers of requests in the city scenario

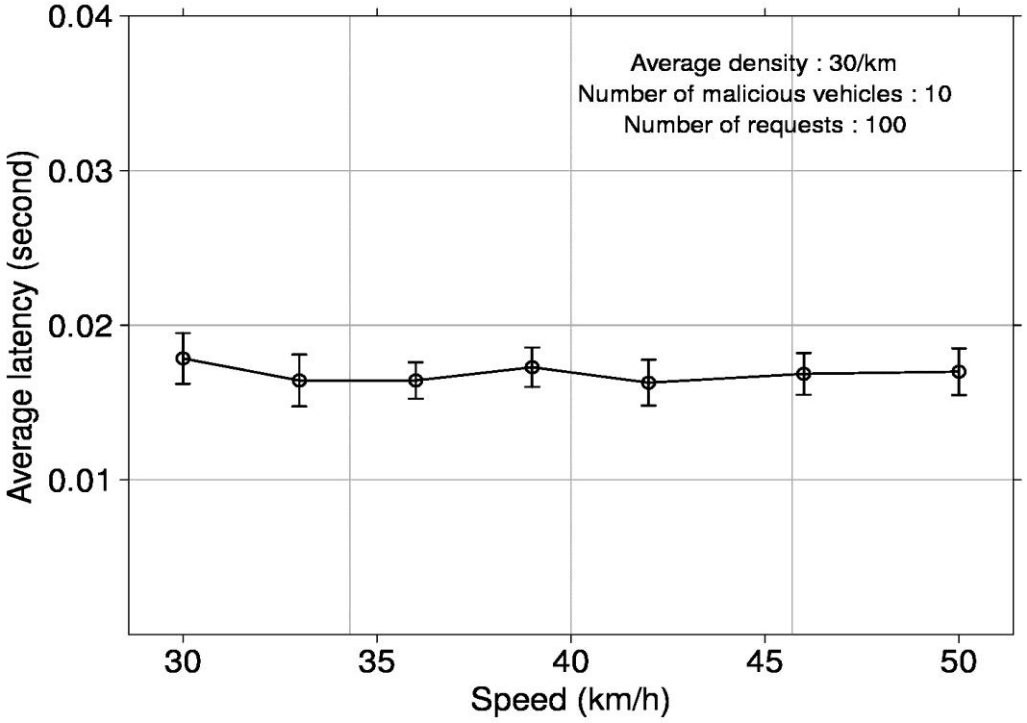

Figure 4.19The average latency for different average speeds in the city scenario

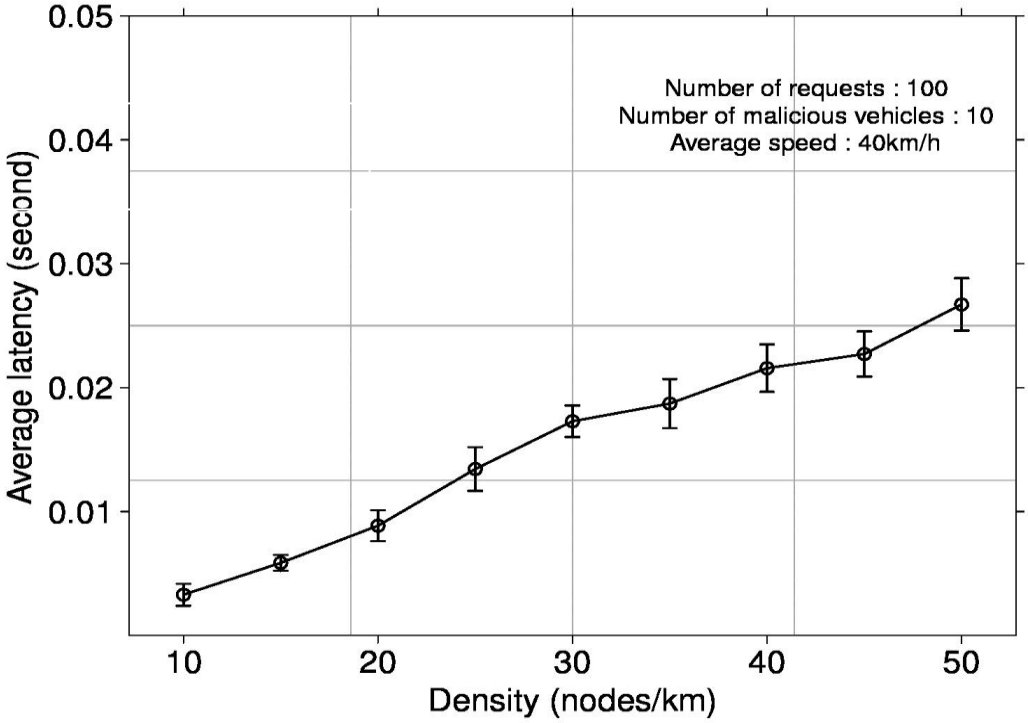

Figure 4.20The average latency for different levels of density in city scenario

Figure 4.18-Figure 4.20] show the average latency (average response time) of gateway request processes in SEGAL for different numbers of requests ranging from 100 to 800, for different average speeds ranging from 30km/h to 50km/h and for different levels of densities ranging from 10/km to 50/km in the city scenario. Figure 4.20 demonstrates how the average latency increases gradually from 0.002 to 0.28 of a second, when the density in the city scenario increases from 10/km to 50/km. The main reason for this is that a higher density may facilitate message congestion, which gives rise to a higher probability of congestion-made delay. As shown in Figure 4.20, the average latency does not surpass the maximum value of 0.028 of a second, which is still considered a low level of response delay. However, Figure 4.18 shows that when the number of gateway requests increases from 100 to 800, the average response time drops significant first from 0.018 of a second to about 0.01 of a second, then keeps stable at around 0.008 of a second. This is because with the increase of the requests number, gateways may maintain a bigger gateway advertisement zone (GAZ) by the use of adapt GAZ mechanism. And each vehicle inside GAZ has the ability to send a gateway reply message. As a result, the average latency is reduced. In Figure 4.19, when the speed of vehicles' increases from 30 km/h to 50 km/h, the average latency remains at a low level; this is around 0.017 of a second. This is mainly because in SEGAL, discovery messages are exchanged in VANETs by only clusterheads. Even though the processes of decryption and encryption cause time consumption, the overall delay is quite low because of the reduction of the exchange time. Both Figure 4.18 and Figure 4.19 prove the scalability of SEGAL

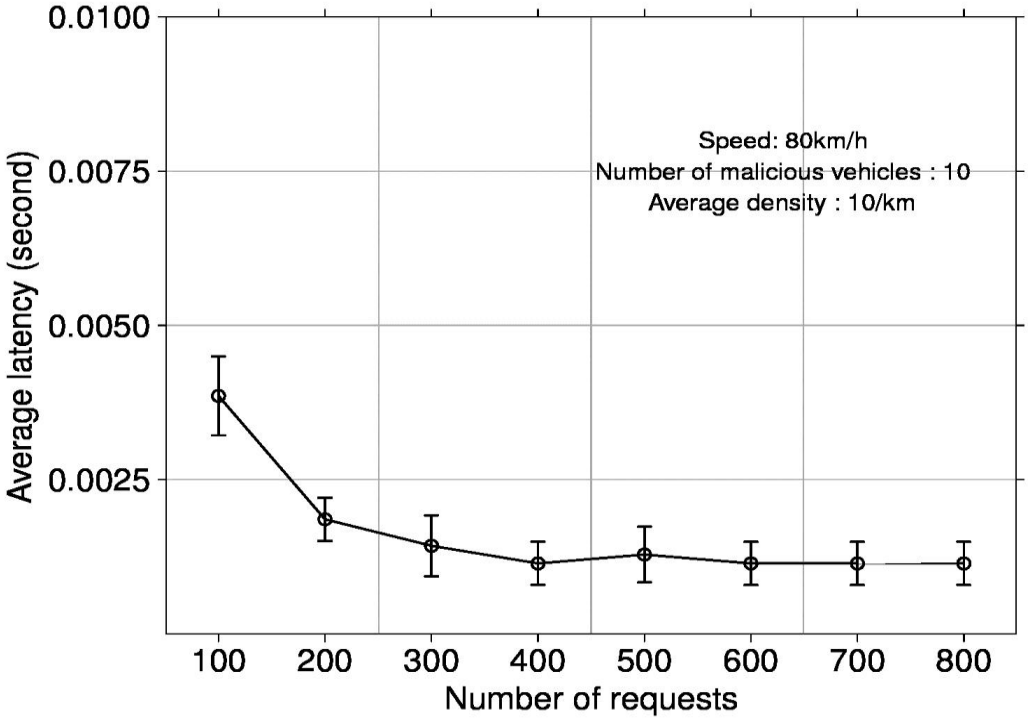

Figure 4.21The average latency for different numbers of requests in the highway scenario

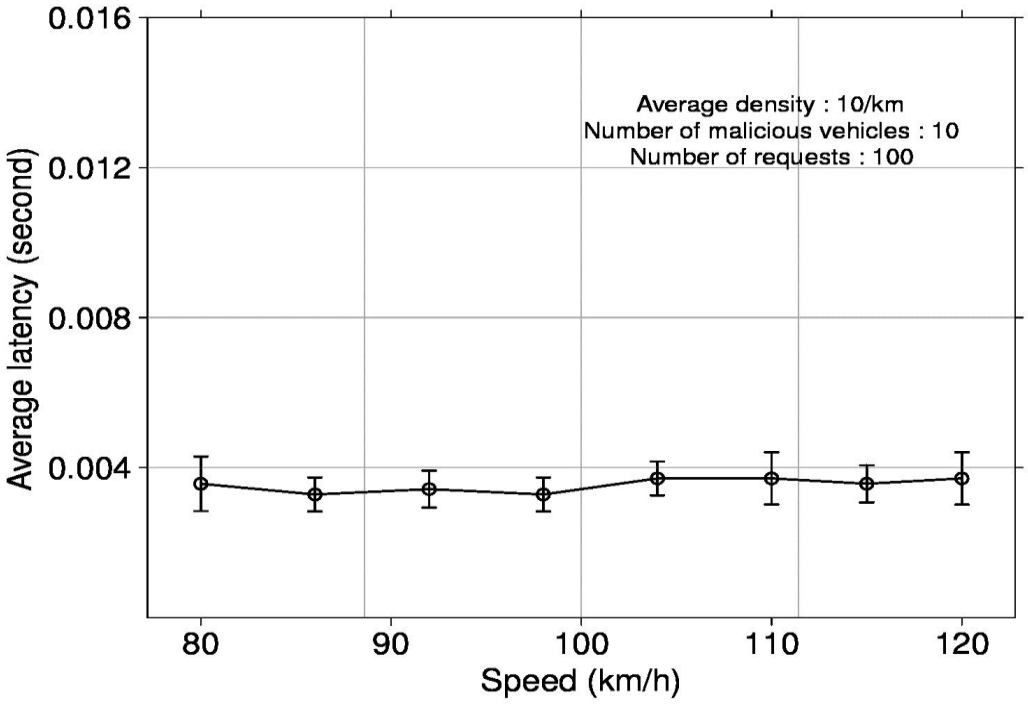

Figure 4.22The average latency for different average speeds in the highway scenario

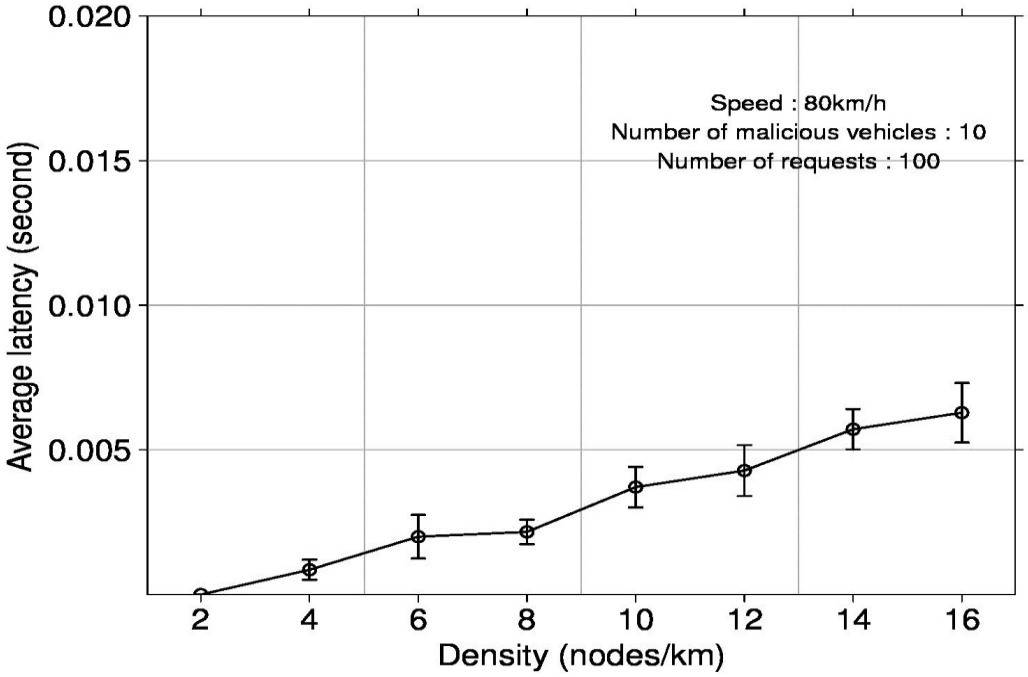

Figure 4.23The average latency for different levels of density in the highway scenario

Figure 4.21-Figure 4.23] plot the average latency of gateway request processes in SEGAL for different numbers of requests ranging from 100 to 800, for different average speeds ranging from 80km/h to 120km/h, and for different levels of density ranging from 2/km to 16/km in the highway scenario. Figure 4.23 show the average latency increases significantly with an increase in the density in the highway scenario. When the density increases from 2/km to 16/km, the average latency grows from just above 0.000 of a second to 0.006 of a second. The reason for this phenomenon is the congestion-made delay. However, similar to the results presented in Figure 4.18 and Figure 4.19, and Figure 4.21 and Figure 4.22 demonstrate that the average latency is kept stable at a very low level, Figure 4.21 shows that when the number of gateway requests increases from 100 to 800, the average response time drops significant first from 0.004 of a second to about 0.002 of a second, then keeps stable at around 0.001 of a second. This is again because with the increase of the requests number, gateways may maintain

a bigger gateway advertisement zone (GAZ) by the use of adapt GAZ mechanism. And each vehicle inside GAZ has the ability to send a gateway reply message. As a result, the average latency is reduced. Figure 4.22 shows that when the average speed increases, the average response time is between 0.003 and 0.004. The main reason is still that the gateway discovery messages are only exchanged through clusterheads in our SEGAL protocol, which reduces the exchange time to achieve a low response delay. It demonstrates again the good stability of SEGAL.

In addition, the values for the average response delay metrics and the value for the total bandwidth usage metrics obtained with all the three variables, namely, the number of request, traffic speed and density, in the highway scenario, are generally less than those obtained with the same variables in the city scenario. That is because the level of traffic density in city scenario is higher than that in highway scenario. Compared to the level of density in a highway scenario, the higher level of density in a city scenario leads to more cluster members in a cluster group. When a clusterhead propagates messages in a city scenario, it needs to send out messages to each cluster member. As a result, the total bandwidth usage in a city scenario is generally more than that in a highway scenario. On the other hand, the high level of density in a city scenario may cause congestions to messages transmission, so that there will be a congestion made delay. However, in a highway scenario, the level of density is comparatively low and causes less congestions to message transmission than that caused by the high level of density in city scenario. Therefore, the total bandwidth usage and the average latency in city scenarios are higher than those in highway scenarios.

Chapter 5

Conclusion

The research on communication systems in vehicular ad hoc networks has been growing rapidly in recent years. In fact, because of the increasing demands for safe message exchanges, much research has concentrated on security issues. In the previously proposed protocols on security issues, most of them focus on ensuring a secure service discovery process for vehicles; none of them, however, concentrate on providing a secure gateway discovery process. The lack of focus on security related improvements could lead to serious consequences, such as placing drivers and passengers in dangerous situations where they are vulnerable to attacks by malicious nodes, since messages may be dropped, changed or replayed on the way to the destination.

In this chapter, we will conclude with our contributions to the secure gateway localization and communication system in VANETs and outline possible directions for our future work.

5.1. Summary of Contributions

In this thesis, we presented a secure, hybrid and adaptive gateway discovery and communication protocol for VANETs. First, we presented a literature review, in which we described the challenges and requirements of security issues in VANETs. Moreover, we introduced several existing secure service discovery protocols and classified them into categories. Then, we proposed our secure gateway localization and communication system in VANETs and reported the performance of our algorithms.

The contributions of this thesis are as follows:

1. *A comprehensive classification and comparison of the secure service discovery protocols in VANETs.*

2. *A cluster-based secure gateway discovery protocol:* First, we explained the different attacks that can occur during the gateway discovery process and our security goals. Second, we described the secure clustering process in our proposed

SEGAL protocol and how the gateway discovery messages are exchanged in a secure manner. Third, we discussed how the security goals explained earlier are met when using the SEGAL protocol.

3. *The performance evaluation of the SEGAL protocol:* First, we compared SEGAL to the LAGAD protocol and we proved that our proposed SEGAL achieves a higher success rate, and a lower response time and dropping rate in the gateway discovery, while maintaining the scalability of the network. Then we showed the performance study of SEGAL protocol in both a city and a highway scenarios objectively.

5.2. Future Work

We can propose several directions for our future work:

- We are planning to apply the pseudo random mechanism to regenerate the cluster group key pair in each time interval. The implementation of this will secure the system access, even if a malicious node gets the cluster group public key, it will not work in the next time interval.

- We are planning to adapt a few existing secure service discovery protocols for mobile ad hoc networks to VANETs, and to conduct more comparison studies, since our proposed SEGAL protocol is the first gateway discovery protocol for VANETs that supports security issues.

Reference

[1] Yue Liu; Jun Bi; Ju Yang; , "Research on Vehicular Ad Hoc Networks," *Control and Decision Conference, 2009. CCDC '09. Chinese* , vol., no., pp.4430-4435,17-19 June 2009

[2] Xiaodong Lin; Rongxing Lu; Chenxi Zhang; Haojin Zhu; Pin-Han Ho; Xuemin Shen; , "Security in vehicular ad hoc networks," *Communications Magazine, IEEE* , vol.46, no.4, pp.88-95, April 2008

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

[4] B. Parno and A. Perrig, “Challenges in Securing Vehicular Networks”, Proc. of Hot Nets-IV, 2005.

[5] Barba, C.T.; Mateos, M.A.; Soto, P.R.; Mezher, A.M.; Igartua, M.A.; , "Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights," *Intelligent Vehicles Symposium (IV), 2012 IEEE* , vol., no., pp.902-907, 3-7 June 2012

[6] Ching-Yi Yang; Shou-Chih Lo; , "Street Broadcast with Smart Relay for Emergency Messages in VANET," *Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on* , vol., no., pp.323-328, 20-23 April 2010

[7] Buchenscheit, A.; Schaub, F.; Kargl, F.; Weber, M.; , "A VANET-based emergency vehicle warning system," Vehicular Networking Conference (VNC), 2009 IEEE , vol., no., pp.1-8, 28-30 Oct. 2009
doi: 10.1109/VNC.2009.5416384

[8] Qiong Yang; Lianfeng Shen; , "A Multi-Hop Broadcast scheme for propagation of emergency messages in VANET," Communication Technology (ICCT), 2010 12th IEEE International Conference on , vol., no., pp.1072-1075, 11-14 Nov. 2010

[9] Namritha, R.; Karuppanan, K.; , "Opportunistic dissemination of emergency messages using VANET on urban roads," Recent Trends in Information

Technology (ICRTIT), 2011 International Conference on , vol., no., pp.172-177, 3-5 June 2011

[10] Costa-Montenegro, E.; Quinoy-Garcia, F.; Gonzalez-castano, F.J.; Gil-Castineira, F.; , "Vehicular Entertainment Systems: Mobile Application Enhancement in Networked Infrastructures," Vehicular Technology Magazine, IEEE , vol.7, no.3, pp.73-79, Sept. 2012

[11] Razzaq, A.; Mehaoua, A.; , "Video transport over VANETs: Multi-stream coding with multi-path and network coding," *Local Computer Networks (LCN), 2010 IEEE 35th Conference on* , vol., no., pp.32-39, 10-14 Oct. 2010

[12] Alpcan, T.; Buchegger, S.; , "Security Games for Vehicular Networks," *Mobile Computing, IEEE Transactions on* , vol.10, no.2, pp.280-290, Feb. 2011

[13] Rongxing Lu; Xiaodong Lin; Haojin Zhu; Xuemin Shen; , "SPARK: A New VANET-Based Smart Parking Scheme for Large Parking Lots," *INFOCOM 2009, IEEE* , vol., no., pp.1413-1421, 19-25 April 2009

[14] Guey-Yun Chang; Jang-Ping Sheu; Cheng-Yu Chung; , "Zooming: A Zoom-Based Approach for Parking Space Availability in VANET," *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st* , vol., no., pp.1-5, 16-19 May 2010

[15] Amoroso, A.; Marfia, G.; Roccetti, M.; Palazzi, C.E.; , "A Simulative Evaluation of V2V Algorithms for Road Safety and In-Car Entertainment," *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on* , vol., no., pp.1-6, July 31 2011-Aug. 4 2011

[16] Rahman, Sumair Ur; Hengartner, Urs; , "Secure crash reporting in vehicular Ad hoc networks," *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on* , vol., no., pp.443-452, 17-21 Sept. 2007

[17] Po-Yu Chen; Je-Wei Liu; Wen-Tsuen Chen; , "A Fuel-Saving and Pollution-Reducing Dynamic Taxi-Sharing Protocol in VANETs," *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd* , vol., no.,

pp.1-5, 6-9 Sept. 2010

[18] Losilla, F.; Garcia-Sanchez, A.J.; Garcia-Sanchez, F.; Garcia-Haro, J.; , "On the role of wireless sensor networks in intelligent transportation systems," Transparent Optical Networks (ICTON), 2012 14th International Conference on , vol., no., pp.1-4, 2-5 July 2012

[19] Xiaodong Lin; Rongxing Lu; Chenxi Zhang; Haojin Zhu; Pin-Han Ho; Xuemin Shen; , "Security in vehicular ad hoc networks," *Communications Magazine, IEEE* , vol.46, no.4, pp.88-95, April 2008

[20] Naz, F.; Chowdhury, T.A.; Sabah, S.H.; Ferdous, H.S.; , "A study on the challenges and importance of vehicular network in the context of bangladesh," *Research and Development (SCOReD), 2011 IEEE Student Conference on* , vol., no., pp.265-270, 19-20 Dec. 2011

[21] Marfia, G.; Roccetti, M.; Amoroso, A.; Pau, G.; , "Safe Driving in LA: Report from the Greatest Intervehicular Accident Detection Test Ever," Vehicular Technology, IEEE Transactions on , vol.PP, no.99, pp.1, 0

[22] Ververidis, C.N.; Polyzos, G.C.; , "Service discovery for mobile Ad Hoc networks: a survey of issues and techniques," *Communications Surveys & Tutorials, IEEE* , vol.10, no.3, pp.30-45, Third Quarter 2008

[23] Abrougui, K.; Boukerche, A.; Pazzi, R.W.N.; , "Design and Evaluation of Context-Aware and Location-Based Service Discovery Protocols for Vehicular Networks," *Intelligent Transportation Systems, IEEE Transactions on* , vol.12, no.3, pp.717-735, Sept. 2011

[24] Leinmuller, T.; Schoch, E.; Maihofer, C.; , "Security requirements and solution concepts in vehicular ad hoc networks," *Wireless on Demand Network Systems and Services, 2007. WONS '07. Fourth Annual Conference on* , vol., no., pp.84-91, 24-26 Jan. 2007

[25] Na Ruan; Hori, Y.; , "DoS attack-tolerant TESLA-based broadcast authentication protocol in Internet of Things," *Mobile and Wireless Networking (iCOST), 2012 International Conference on Selected Topics in* , vol., no., pp.60-65, 2-4 July 2012

[26] Li He; Wen Tao Zhu; , "Mitigating DoS attacks against signature-based

authentication in VANETs," *Computer Science and Automation Engineering (CSAE), 2012 IEEE International Conference on* , vol.3, no., pp.261-265, 25-27 May 2012

[27] Hubaux, J.P.; Capkun, S.; Jun Luo; , "The security and privacy of smart vehicles," *Security & Privacy, IEEE* , vol.2, no.3, pp.49-55, May-June 2004

[28] Chhoeun, S.A.; Ayutaya, K.S.N.; Charnsripinyo, C.; Chamnongthai, K.; Kumhom, P.; , "A Novel Message Fabrication Detection for Beaconless Routing in VANETs," *Communication Software and Networks, 2009. ICCSN '09. International Conference on* , vol., no., pp.453-457, 27-28 Feb. 2009

[29] Ming-Chin Chuang; Jeng-Farn Lee; , "PPAS: A privacy preservation authentication scheme for vehicle-to-infrastructure communication networks," *Consumer Electronics, Communications and Networks (CECNet), 2011 International Conference on* , vol., no., pp.1509-1512, 16-18 April 2011

[30] Samara, G.; Al-Salihy, W.A.H.; Sures, R.; , "Security issues and challenges of Vehicular Ad Hoc Networks (VANET)," New Trends in Information Science and Service Science (NISS), 2010 4th International Conference on , vol., no., pp.393-398, 11-13 May 2010

[31] Bayrak, A.O.; Acarman, T.; , "S3P: A Secure and Privacy Protecting Protocol for VANET," Wireless and Mobile Communications (ICWMC), 2010 6th International Conference on , vol., no., pp.441-447, 20-25 Sept. 2010

[32] Raya, M.; Papadimitratos, P.; Hubaux, J.-P.; , "SECURING VEHICULAR COMMUNICATIONS," *Wireless Communications, IEEE* , vol.13, no.5, pp.8-15, October 2006

[33] Calandriello G, Papadimitratos P, Lloy A, and Hubaux J-P.Effcientand robust pseudonymous authentication in vanets.*In Proceedings of VANET'07*, 2007.

[34] Patrick Tsang and Sean Smith.Ppaa: Peer-to-peer anonymous authentication. 5037:55–74, 2008.

[35] Xiaodong Lin; Xiaoting Sun; Pin-Han Ho; Xuemin Shen; , "GSIS: A Secure and Privacy-Preserving Protocol for Vehicular Communications," *Vehicular Technology, IEEE Transactions on* , vol.56, no.6, pp.3442-3456, Nov. 2007

[36] Wasef, A.; Xuemin Shen; , "PPGCV: Privacy Preserving Group

Communications Protocol for Vehicular Ad Hoc Networks," *Communications, 2008. ICC '08. IEEE International Conference on* , vol., no., pp.1458-1463, 19-23 May 2008

[37] Chenxi Zhang; Xiaodong Lin; Rongxing Lu; Pin-Han Ho; Xuemin Shen; , "An Efficient Message Authentication Scheme for Vehicular Communications," *Vehicular Technology, IEEE Transactions on* , vol.57, no.6, pp.3357-3368, Nov. 2008

[38] Behera, S.; Mishra, B.; Nayak, P.; Jena, D.; , "A secure and efficient message authentication protocol for vehicular Ad hoc Networks with privacy preservation(MAPWPP)," *Internet Multimedia Systems Architecture and Application (IMSAA), 2011 IEEE 5th International Conference on* , vol., no., pp.1-6, 12-13 Dec. 2011

[39] Petit, J.; , "Analysis of ECDSA Authentication Processing in VANETs," *New Technologies, Mobility and Security (NTMS), 2009 3rd International Conference on* , vol., no., pp.1-5, 20-23 Dec. 2009

[40] Calandriello G, Papadimitratos P, Lloy A, and Hubaux J-P.Effcientand robust pseudonymous authentication in vanets. In Proceedings ofVANET'07, 2007.

[41] You Lu; Biao Zhou; FeiJia; Gerla, M.; , "Group-based secure source authentication protocol for VANETs," *GLOBECOM Workshops (GC Wkshps), 2010 IEEE* , vol., no., pp.202-206, 6-10 Dec. 2010

[42] Perrig A, Canneti R, Song D, and Tygar JD. The tesla broadcast authentication protocol. *RSA Cryptobytes*, 5(2):213, 2002.

[43] Studer A, Bai F, Bellur B, and Perrig A. Flexible, extensible, and efficient vanet authentication. *Journal of Communications and Networks(JCN), (Special Issue on Secure Wireless Networks)*, 11(6):574588, 2009.

[44] Fischer L, Aijaz A, Eckert C, and Vogt D. Secure revocable anonymous authenticated inter-vehicle communication (sraac).*In Proceedings of Workshop on Embedded Security in Cars (ESCAR)*, 2006.

[45] Kamat P, Baliga A, and Trappe W.An identity-based security frameworkfor vanets.*In VANET '06: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks*, page 9495, 2006.

[46] Marshall Riley, Kemal Akkaya, and Kenny Fong. Group-based hybrid authentication scheme for cooperative collision warnings in vanets. Security and Communication Networks, 2011.

[47] Xiaodong Lin; Xiaoting Sun; Xiaoyu Wang; Chenxi Zhang; Pin-Han Ho; Xuemin Shen; , "TSVC: timed efficient and secure vehicular communications with privacy preserving," *Wireless Communications, IEEE Transactions on* , vol.7, no.12, pp.4987-4998, December 2008

[48] Abrougui, K.; Boukerche, A.; Pazzi, R.W.N.; , "Location-Aided Gateway Advertisement and Discovery Protocol for VANets," *Vehicular Technology, IEEE Transactions on* , vol.59, no.8, pp.3843-3858, Oct. 2010