The Islamic University of Gaza
Deanery of Graduate Studies
Faculty of Engineering
Computer Engineering Department

PRIVACY PRESERVING SECURE COMMUNICATION PROTOCOL FOR VEHICULAR AD HOC NETWORK (VANET)

By

Yousif Mostafa Mansour

Supervised by

Prof. Mohammad Mikki

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Computer Engineering

July, 2012

ABSTRACT

Vehicular Ad-hoc Networks (VANETs) have attracted much attention recently because of its applications and features. The main purpose of adopting VANET technology is to increase safety and efficiency on roads. In VANET, vehicles broadcast safety messages periodically. Considering security with preserving privacy of vehicles in VANET is very important. Unauthorized tracking of vehicles is a major problem which violates privacy requirement. Therefore, an anonymous message authentication scheme should be used.

In this thesis, a privacy preserving secure communication protocol (PPSCP) for VANET is proposed to address the issue on anonymous authentication for safety messages with authorized traceability. In PPSCP, vehicles authenticate safety messages with shared symmetric keys using Message Authentication Code (MAC) algorithm. The trusted authority generates and distributes shared keys to all legitimate vehicles through road-side units (RSUs). All vehicles use the same shared key at the same time which hides the sender identity. Moreover, authorized tractability by a trusted authority can be achieved without affecting privacy. The vehicle identity is encrypted with the public key of the trusted authority. This scheme allows only the trusted authority to reveal the vehicle's identity because it is the only entity which has the corresponding private key. The protocol is designed to be resistant to attacks like replay attack.

The proposed protocol suggests a new scheme for revocation which strongly reduces the size of revocation lists. In this scheme, each vehicle has its own revocation key which is updated periodically. The revocation key is used to encrypt a defined value which is included in the message. The receiver tries to decrypt the encrypted value with all revocation keys which are included in the revocation list. The trusted authority is responsible for creating and maintaining the revocation list. When a misbehaved vehicle is detected, its revocation key will be added to the revocation list. The vehicle will be anonymous until it is revoked. The revocation list is broadcasted periodically to all vehicles. This suggested scheme keeps the revocation list small by removing expired revocation keys.

A security analysis was performed which demonstrates that PPSCP is secure and provides privacy preservation and liability effectively. PPSCP performance is evaluated through a simulation which shows that the proposed protocol is robust and efficient in compare with the S3P protocol.

بروتوكول اتصال آمن ومحافظ على الخصوصية لشبكة المركبات اللاسلكية الخاصة

إعداد: يوسف مصطفى منصور

ملخص

جذبت شبكة المركبات اللاسلكية الخاصة (VANET) الكثير من الاهتمام مؤخرا لما لها من تطبيقات ومزايا. إن الغرض الرئيسي لتبني تقنية (VANET) هو زيادة الأمان والكفاءة على الطرقات، في شبكة (VANET)؛ تبث المركبات رسائل أمان على نحو دوري، إن أخذ الأمن بعين الاعتبار مع المحافظة على خصوصية المركبات في تقنية (VANET) أمر مهم للغاية، كما أن تعقب المركبات غير المصرح به يشكل مشكلة كبيرة تنتهك متطلب الخصوصية، لذلك يتوجب استخدام أسلوب المصادقة المجهولة للرسائل.

في هذه الدراسة؛ تم اقتراح بروتوكول اتصال آمن ومحافظ على الخصوصية لشبكة المركبات اللاسلكية الخاصة يدعى اختصارا (PPSCP) وذلك لمعالجة مسألة المصادقة المجهولة لرسائل الأمان مع إمكانية التعقب المصرح به، في هذا البروتوكول؛ تقوم المركبات بمصادقة رسائل الأمان باستخدام مفتاح تشفير متناظر ومشترك باستخدام خوارزمية (MAC) لمصادقة الرسائل. تقوم سلطة موثوقة بتوليد ونشر المفاتيح المشتركة إلى كل المركبات الشرعية عبر الوحدات الجانبية للطريق (RSUs)، تستخدم جميع المركبات نفس المفتاح المشترك في نفس الوقت مما يؤدي إلى إخفاء هوية المرسل. علاوة على ذلك، يمكن للتعقب المصرح به من قبل سلطة موثوقة أن يتم دون التأثير على الخصوصية، حيث أن هوية المركبة تشفر باستخدام المفتاح العام للسلطة الموثوقة، هذا الأسلوب يسمح فقط للسلطة الموثوقة بالكشف عن هوية المركبة لأنها الكيان الوحيد الذي يملك المفتاح الخاص. لقد تم تصميم البروتوكول المقترح بحيث يكون مقاوما للهجمات مثل هجوم إعادة الإرسال.

يقدم البروتوكول المقترح أسلوبا جديدا لعملية النقض بحيث يقلل من حجم القوائم الخاصة بها، في هذا الأسلوب؛ تمتلك كل مركبة مفتاح النقض الخاص بها والذي يحدث على نحو دوري، يستخدم مفتاح النقض لتشفير قيمة معروفة ومضمنة في الرسالة، يحاول المستقبل فك تشفير القيمة المشفرة السابقة باستخدام جميع مفاتيح النقض الموجودة ضمن قائمة النقض، إن السلطة الموثوقة هي المسئولة عن إنشاء وتحديث قائمة النقض، فعندما يتم اكتشاف مركبة تسيء التصرف؛ يتم إضافة مفتاح النقض الخاص بها إلى قائمة النقض، ستبقى المركبة مجهولة الهوية إلى أن يتم نقضها، يتم بث قائمة النقض بشكل دوري إلى جميع المركبات، كما أن الأسلوب المقترح يبقي قائمة النقض صغيرة عبر إزالة المفاتيح منتهية الصلاحية.

تم إجراء تحليل للأمن أظهر أن البروتوكول (PPSCP) آمن ويوفر المحافظة على الخصوصية والمسئولية بشكل فعال، كما تم تقييم أداء البروتوكول (PPSCP) من خلال المحاكاة والتي أظهرت أنه كفء وفعال مقارنة بالبروتوكول (S3P).

DEDICATION

To my great father and my great mother

To my wife and my daughter

To my sister and my brothers

And to my beautiful country "Palestine"

ACKNOWLEDGMENTS

First of all, all praises be to Allah for helping me to finish this work.

I would like to record my gratitude to Prof. Mohammed Mikki for his supervision, advice, and guidance from the very early stage of this research as well as giving me extraordinary experiences throughout the work. He provided unflinching encouragement and support in various ways.

Lastly, the deepest thanks are due to my family members who have been a pillar of support during the arduous times of my research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter One: Introduction

Transportation systems play an important role in our life. However, current road transportation systems have several deficiencies and inefficiencies which cause problems like traffic jams and accidents. Many researchers are trying to enhance safety and efficiency on roads. They proposed to use intelligent transportation systems (ITSs). Some organizations proposed ITS projects and applications such as the U.S.A. Vehicle Safety Consortium [21], the Japan Road and Traffic Intelligence Society Organization [22], the Taiwan Intelligent Transport Society [23].

An important component of ITS in various designs is VANET. Vehicular Ad hoc Network (VANET) is a kind of Mobile Ad hoc Network (MANET) which is a wireless ad hoc network consisting of mobile nodes. VANET consists of vehicles as mobile nodes and road side units (RSUs) as fixed nodes. Vehicles can communicate with each other (V2V) and with Infrastructure of RSUs (V2I), see Figure 1-1.

Using these communications, a wide variety of applications can be employed. The Applications are classified into two categories [3]: safety-related applications and infotainment applications. In safety-related applications, vehicles are broadcasting safety-related messages or beacons to warn vehicles about traffic situations like congestions and collisions. Safety-related message contains information like location, speed and acceleration. They are divided into two types: periodic and event-driven messages which are sent when a hazardous situation occurs [13]. Infotainment applications include other kind of applications like payment services, internet-access and so on.

Many researches focus on securing safety-related messages or shortly safety messages because attacks on these messages may lead to catastrophic results [1]. Safety messages need to be authenticated to prevent attackers from impersonating other legitimate vehicles and sending false messages. However, providing authentication using traditional ways disagrees with privacy.

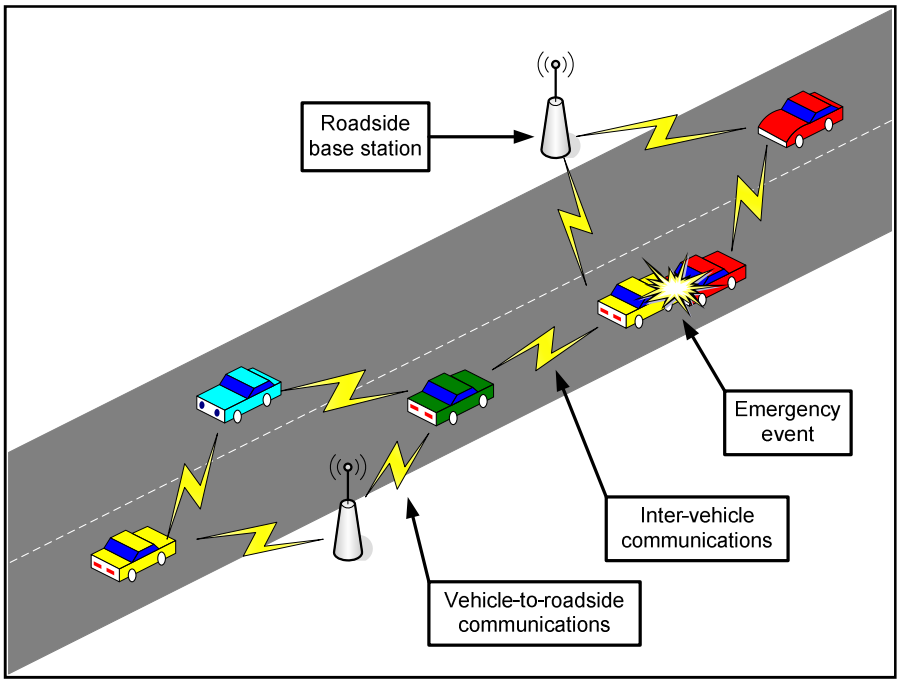

Figure 1-1: VANET Architecture

Privacy is very important factor to adopt VANET widely. Drivers want to protect their identities from others. They dislike to be traced or eavesdropped by other vehicles or entities. On opposite, the vehicle identity must be determined if that vehicle misbehaves and harms other vehicles. Drivers must be responsible for their messages.

1.1. Problem Statement

Safety messages sent by vehicles in VANET need to be authenticated to ensure that they are from legitimate vehicles. Traditional techniques of authentication add some information to message which indicates to the identity of sender vehicle. However, drivers of vehicles want to keep their identities hidden from others. Hiding identity makes it difficult to achieve non-repudiation when a vehicle misbehaves or denies responsibility.

Preserving driver privacy without affecting authentication and non-repudiation needs extra effort. Many papers try to balance between privacy and vehicle identification. In [3], every vehicle stores many private/public key certificates obtained from authority to sign messages which are referred to as pseudonyms. A pseudonym is a short-lifetime

certificate that does not contain identity-linking information [14]. A pseudonym is used in one period of time to prevent others from tracking the vehicle. This approach has many problems. For instance, a large storage space is needed at every vehicle. Moreover, including the certificate in the safety message leads to larger message size and needs more computations to verify every certificate at receiver side.

According to [3], each vehicle needs 43,800 keys per year. The authority must certify all vehicle keys. That leads to a big overhead. For liability, the authority should store all those keys to identify the misbehaving vehicle from safety messages. Moreover, the authority needs to search in a very huge number of keys which costs a time. Another overhead will happen when the authority needs to revoke all vehicle keys because it leads to very large Certificate Revocation Lists (CRLs).

Furthermore, pseudonyms are vulnerable to temporal and spatial locality. Hence, the attacker may be able to link between successive pseudonyms according to adjacency between two pseudonyms in time or location. Some papers like [6], [7] and [8] tries to solve this problem but their solutions are not effective.

1.2. Motivation

Many people are killed in car accidents every day. Approximately 35,000 people are killed on roads in European Union every year [51]. In USA, Around 34,000 people are killed in car accidents every year [52] (statistics of year 2009). Besides, the number of vehicles is increasing rapidly which leads to frequent traffic jams.

The promising solution to increase road safety and traffic efficiency is the use of inter-vehicle communications system which is referred to as Vehicular Ad hoc Network (VANET). However, the use of this technology has a deep implication for security and privacy. Security issue of VANET has been discussed in many researches. Howver, traditional security protocols do not preserve privacy. Drivers do not want to be tracked by other entities. The privacy issue may postpone the adoption of VANET by drivers.

1.3. Objective

The objective of this thesis is to introduce a novel privacy preserving secure communication protocol (PPSCP) for Vehicular Ad hoc Network (VANET). The objectives of the proposed protocol are:

- To authenticate safety messages anonymously to preserve vehicle privacy and prevent unauthorized tracking. Authentication should not reveal the sender identity. Authentication ensures that the safety message is sent from a legitimate vehicle and the message is not altered by an attacker.
- To give a trusted authority the ability to recognize the vehicle identity from its safety messages. That ability is used in special cases such as accidents and misbehaviors.
- To make a new efficient revocation scheme reducing revocation list size. When a misbehaved vehicle is detected, it will be revoked by a trusted authority. Revocation lets vehicles to discard safety messages from all revoked vehicles.
- To prevent replay attack. This attack can be carried out by repeating or delaying a valid safety message.
- To increasing efficiency by decreasing the time needed to authenticate and verify safety messages at every vehicle.

1.4. Literature Review

Early papers propose to use pseudonyms to preserve privacy like [3] and [15]. Pseudonyms are many private-public key pairs installed on each vehicle by an authority. Each vehicle uses one of those pseudonyms in one period and does not use it again. This method protects vehicle identity to be tracked by unauthorized observers. One major problem is the linkability of pseudonyms. The attacker may identify the targeted vehicle by linking the previous pseudonym with current one by temporal or spatial locality. Another problem is the overhead of large number of pseudonyms needed to be preloaded at each vehicle.

A solution was proposed in [16] to reduce the large number of pseudonyms which are preloaded on each vehicle. The solution reduces the number by half, on average. It depends on using two certificates: encryption certificate and signing certificate. In this approach, the initiator of connection sends its encryption certificate to the responder vehicle. The responder vehicle sends a safety message signed with its signing certificate concatenated to signing certificate itself then encrypts all with the initiator encryption public key. The responder can use its signing certificate in following sessions because it was sent encrypted.

To alleviate linkability problem, some approaches like in [17] apply a strategy called "hiding in crowd". In this approach, the pseudonyms are updated regularly according to spatial or temporal criteria. However, there are situations where linkability is unavoidable. One of these situations is driving on a long road without junctions. In that case, the vehicle can be traced or linked to its group in spite of changing its pseudonyms.

Other approaches like in [6] and [18] try to solve the locality problem by using a random silent period between changing of pseudonyms. In the silent period, vehicle does not transmit any message. The period duration is random but it should be short. It is hard to link between vehicles before and after the silent period. In this approach vehicles must change their pseudonyms in adjacent times. This approach is not practical because of the necessity of broadcasting safety messages regularly.

Another solution is proposed in [7] and [8]. In this solution, vehicles are belongs to regions called mix-zones. Each vehicle in the same mix-zone changes its pseudonym at the same time. This solution decreases the linkage problem, but it depends on the number of vehicles in each mix-zone.

Some researchers employ Identity-Based Cryptography (IBC) where certificates are not needed for authentication. IBC was proposed in 1984 by Shamir [28]. IBC differs from public key infrastructure (PKI). In this system, the sender can use the identity information of the receiver such as name or email address to encrypt the message without the need to use a public key certificate to verify the sender. In 2001, Boneh and Franklin [29]

introduced the first functional and efficient identity-based encryption scheme that was based on bilinear pairings property of an elliptic curve.

Kemat et al. [24], [25] proposed an approach based on identity-based cryptography (IBC) which provides authentication, non-repudiation and privacy. In this approach, each pseudonym which is an anonymous identity is generated by the RSU. The approach enables a single authority to reveal the identity. However, their approach is very dependent on the RSUs which may not be reachable or very busy. Other approaches were proposed like in [26], [27] which try to avoid the disadvantages of previous approach.

Some schemes use the group signature approach as in [9] and [10]. In this approach, vehicles are arranged into groups. Each group has a group manger. The manger is responsible for signing vehicle messages. The identity of the vehicle can be detected only by the group manger.

Another group-based approach is described in [19]. In this approach, the group manger signs vehicles pseudonyms to reduce certificate authority workload. Each vehicle produces its pseudonyms and signs its messages. However, group-based approaches are not scalable especially if number of vehicles in the group is very large.

Paper [20] proposes an approach which does not depend on pseudonyms. This approach uses Hash-based Message Authentication Code (HMAC). Before a vehicle sends a message, it requests a symmetric key from RSU. Then, the vehicle signs its message with that key using HMAC Algorithm. The receiver vehicle authenticates the message from any adjacent RSU. This approach offers anonymity but it depends highly on RSU which may be not available.

The scheme proposed in [2] does not use pseudonyms or groups to preserve privacy. It proposed to use a shared private/public key which is given to all legitimate vehicles. This key is renewed regularly by an authority. Furthermore, each vehicle has its owned public/private key to communicate with the authority. When a vehicle sends a safety message, it first signs it with its owned private/public key and then encrypts the signature by the authority public key to hide its identity. After that, the vehicle signs the message by the shared private/public key for authentication. However, there is a problem arises when a

vehicle are revoked, because the shared key must be renewed at all vehicles. Moreover, signing the message two times and encrypting it one time by a public key encryption needs heavy computations and leads to larger message size.

1.5. Contributions

To the best of our knowledge, this study is one of the rare studies in this field which uses the shared keys instead of pseudonyms or anonymous certificates to authenticate safety messages. One of those studies is S3P protocol which was proposed in [2]. S3P uses shared public-private key pairs to authenticate messages. Our proposed protocol, PPSCP, has major contributions which can be described as follows:

- Authenticating safety messages with shared symmetric keys using Message Authentication Code (MAC) Algorithm enhances the protocol performance. The shared keys are shared between all vehicles and updated periodically. They are generated and distributed by a trusted authority through road-side units (RSUs).

- The encrypted identity of the sender vehicle, which is included in the safety message, is not related to the message itself and can be pre-initialized to reduce the time needed before sending a message. The identity is encrypted with the public key of the trusted authority which can use its own private key to reveal the vehicle identity.

- Designing a novel revocation scheme which depends on symmetric keys called revocation keys. Each vehicle has a revocation key which is used to encrypt the timestamp. When a vehicle needs to send a safety message, it will add the encrypted timestamp and the timestamp itself to the message. When a vehicle receives a message, it will try to decrypt the encrypted timestamp with all revocation keys existing in the revocation list. A trusted authority is responsible of maintaining and broadcasting the revocation list.

- Reducing the revocation list size by including the revocation keys of the revoked vehicles instead of all pseudonyms or all certificates. When the revocation key lifetime expires, it will be removed from the list which keeps the list small.

1.6. Thesis Organization

The rest of this thesis is organized as follows. Chapter 2 describes the VANET Architecture. Chapter 3 demonstrates a literature review on anonymous authentication techniques. Chapter 4 describes the proposed protocol in detail. Chapter 5 presents the simulation and results that demonstrate the effectiveness of our protocol. Chapter 6 concludes our work and provides directions for future work.

Chapter Two: VANET Architecture

2.1. Introduction

Vehicular Ad hoc Network (VANET) is an example of Mobile Ad hoc Network (MANET). Mobile Ad hoc Network (MANET) is an infrastructure-less wireless network. The term "Ad hoc" implies that the network is formed in a spontaneous manner, so MANET can be formed on the fly [40]. Mobile nodes are autonomous devices that can move freely. Mobile nodes are mobile wireless devices such as Cellular Phones, Smart Phones, PDAs, Laptops, Tablet PCs, and so on.

A Mobile ad hoc Network (MANET) is an adaptive, self-configurable, self-organizing, infrastructure-less multi-hop wireless network with dynamic topologies [41]. MANETs consists of mobile nodes that communicate over multi-hop wireless links without the need of any infrastructure such as base stations [39].

A large number of routing protocols for MANETs are proposed. Routing protocols used in MANETs can be classified mainly into three types: proactive, reactive and hybrid. Proactive protocols maintain fresh routing tables at each node. The routing table contains all destination nodes and the paths to reach them. An example of proactive protocols is DSDV (Highly Dynamic Destination-Sequenced Distance Vector routing protocol) [45]. In reactive protocols, routes are found on-demand by flooding the network with route request messages. DSR (Dynamic Source Routing) [42] and AODV (Ad-Hoc On-Demand Distance Vector) [43] [44] are examples of reactive protocols. Hybrid protocols merge between proactive and reactive protocols like ZRP (Zone Routing Protocol) [46].

MANETs have been emerged to provide a large number of applications and services which need multi-hop wireless communications without infrastructure. The applications of MANETs can be employed in many situations such as emergency disaster relief, battlefield control, space exploration and wireless classrooms, etc [38].

In VANET, mobile nodes are vehicles. Both MANETs and VANETs are self-organizing and decentralized systems [31]. In MANET, mobile nodes are moved freely and randomly. Unlike MANET nodes, vehicles are moving in organized manner on determined paths or roads. Furthermore, in VANET, there are fixed nodes established on sides of roads which are referred to as Road-Side Units.

2.2. System Model

Vehicular Ad hoc Network consists of two entity types: Vehicles as mobile nodes, and Road Side Units (RSUs) as fixed nodes.

Each vehicle in VANET possesses a network of sensors connected to a central computing platform which is called On-Board Unit (OBU). OBU provides communication facilities like IEEE 802.11 interface [30]. OBUs enable vehicles to communicate among themselves.

Vehicles of VANET are smart vehicles because they are equipped with recording, processing, positioning, and location capabilities. Besides, they can run wireless networking protocols [30].

Road Side Unit (RSU) is fixed station with wireless networking capability positioned on the side of road. RSU acts as an access point and communicates with vehicles in its transmission range.

2.2.1. On-Board Unit

Vehicle On-Board Unit OBU is a central computing platform connected with communication facility and other devices like: sensors and data recorders (See Figure 2-1). Some of the most used devices are:

1- Event Data Recorder (EDR): to record vehicle data for crash reconstruction or determination of misbehaved vehicles.
2- GPS receiver: to get the position of vehicle for using it in navigation system.
3- Front-end and rear radars: for detecting obstacles at front and rear of vehicle. They can be used for parking. [30]

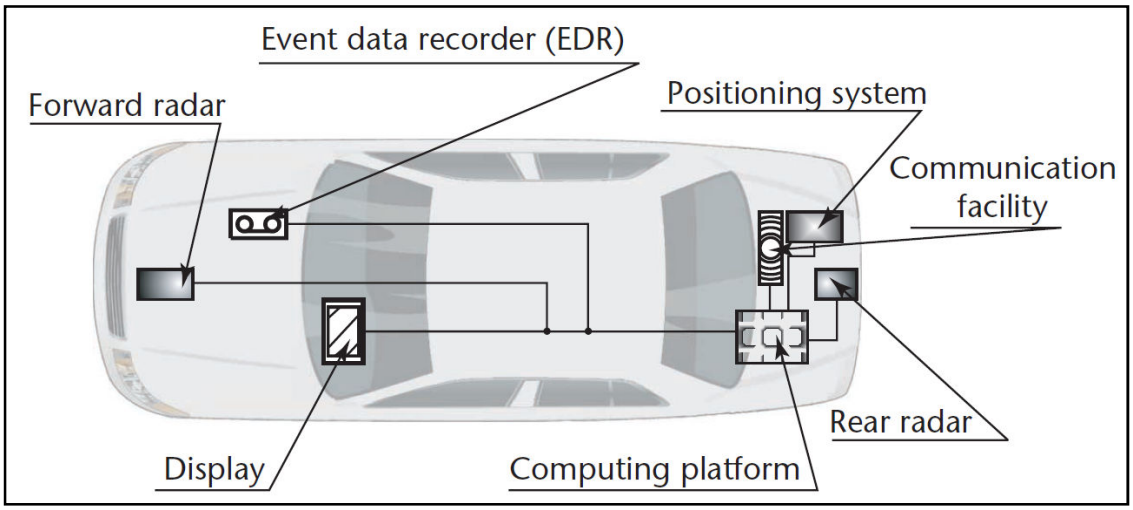

Figure 2-1: A smart vehicle's onboard instrumentation.

2.2.2. Road Side Unit

Infrastructure of VANET consists of RSUs which are connected to location server by a wired network. The location server records all the location data forwarded by the RSUs. In addition, a trusted Registration Authority (RA) provides authentication and authorization service to vehicles through RSUs [18].

2.2.3. Communication

Vehicles in VANET can communicate with each other (V2V communications) or communicate with infrastructure of RSUs (V2I communications) [32]. (See Figure 2-2)

In October 1999, the US Federal Communications Commission allocated 75 MHz of the spectrum in 5.9 GHz band in America for dedicated short-range communications (DSRC) for vehicular communication in ITS. [30]

The networking standard adopted in VANET is IEEE 802.11p which is based on DSRC. According to vehicle speed, messages are sent every 100ms to 300ms [5]. The transmission distance ranges from 110m to 300m as mentioned in [3].

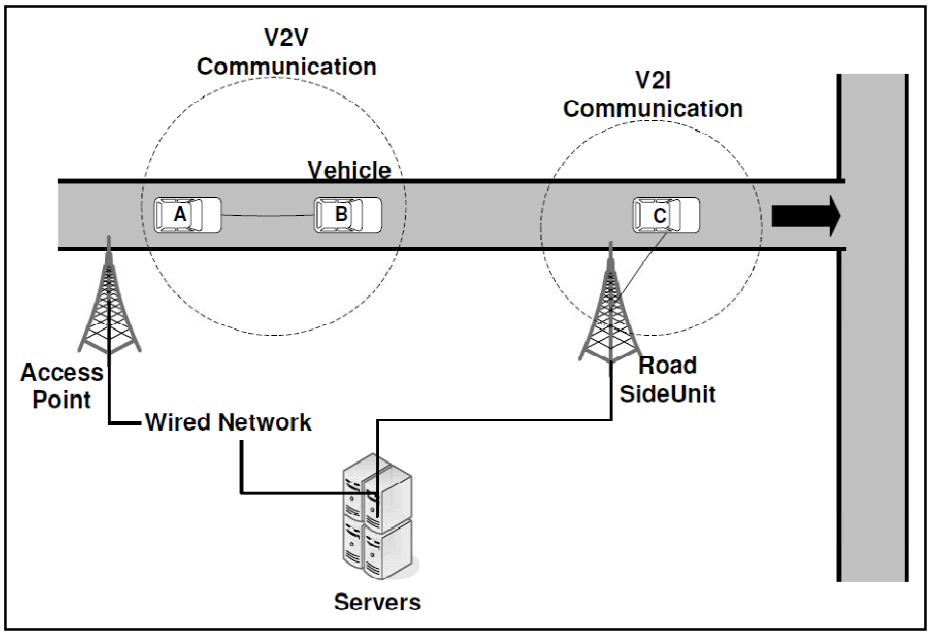

Figure 2-2: Illustration of VANET communications

2.2.4. Electronic License Plate

A License Plate is a metal or plastic plate attached to a vehicle for identification purposes. The identifier on plate is a number or alphanumeric code which uniquely identifies the vehicle within the issuing authority's area. Today, tracking vehicles depends on reading license plates.

Some efforts in the US and EU have made toward electronically identifying vehicles. Electronic License Plate (ELP) is a certified identity provided by a vehicle via a wireless link [30]. ELP needs an onboard device to transmit the certified identity.

ELP can referred to as Vehicle Identification Number (VIN) as used in [11]. ELP is used in VANET to establish the liability of drivers. For example, ELP can be used to find drivers who escape from accident location. In the same time, identity of vehicle should be protected to preserve privacy. Therefore, ELP is sent after encrypting it using a cryptographic mechanism.

Vehicle ELP must be protected against attacks like impersonation attack. In this attack, the attacker steals another vehicle's identity and uses it as his original identity. To

prevent this type of attack, we can store vehicle certified identity in a tamper-proof device [30].

2.5. Tamper-Proof Device

Many VANET researches, like [2] and [3], use a tamper proof device (TPD) which is tamper resistant hardware. TPD device is the place which the secret information like certified identity and cryptographic keys are stored in. TPD is responsible for securing and authenticating messages. It is fabricated such that no one can reveal or compromise its information. TPD should erase all of secret information, if it was removed from the vehicle. TPD is installed and preloaded by secret information from a transportation authority. OBU uses the TPD to secure and authenticate messages. Before sending a message, OBU passes the message to TPD as input, and get the secured message as output. Moreover, when OBU receives a message, it passes that message to TPD to check if it is authenticated or not (See Figure 2-3).

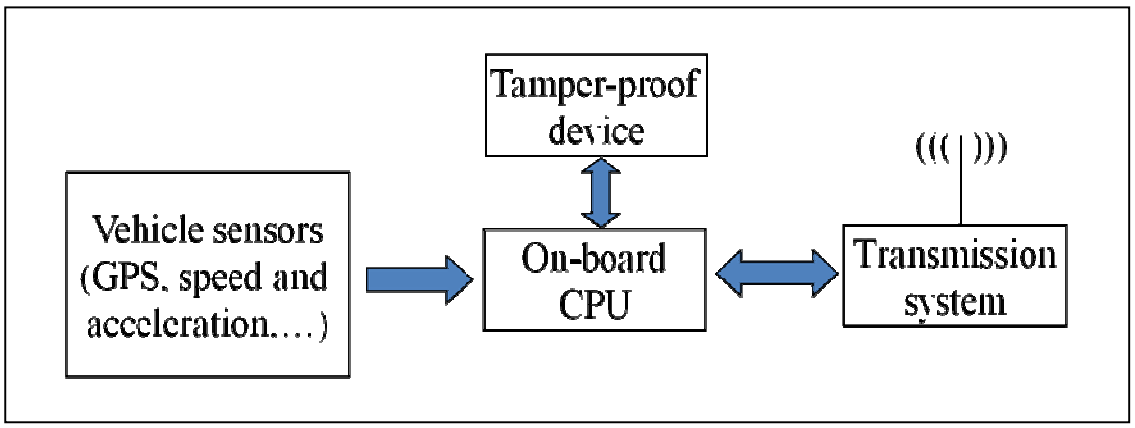

Figure 2-3: OBU Interfacing with other devices

2.3. Security Requirements

VANET should be protected from attacks by a security system. VANET security system should satisfy the following requirements as mentioned in [3] and [4]:

2.3.1. Authentication

When a vehicle receives a message, it should ensure that the message is from a legitimate vehicle. Thus, the sender must authenticate each message before sending it. This prevents intruders from sending false messages.

2.3.2. Data Integrity

Message integrity is very important because false massages may cause severe consequences. If a received message was changed by an attacker, the receiver should be able to detect that. Therefore, it is not enough to get a message from a legitimate sender but also the message itself should be verified. Furthermore, this requirement encompasses detecting message repetition by an attacker which may be very harmful.

2.3.3. Non-repudiation

A misbehaving vehicle may send incorrect information, but the vehicle itself is legitimate and the message is consistent. That behavior may lead to bad situations like accidents. The sender should not deny that he sent that message.

2.3.4. Privacy

Drivers do not want be identified and tracked by others. This is a very critical requirement which is needed to adopt VANET as a solution. However, the problem is that privacy contradicts with authentication and non-repudiation. Hence, many researches try to solve that problem.

2.3.5. Availability

The system should be available all the time because absence of connection for short time may be dangerous. The system should be protected against Denial of Service (DoS) Attack. This attack may be done by jamming the communication channel. Therefore, many solutions were proposed to protect VANET from this attack.

2.4. Applications

Drivers need more safety and efficiency on roads. VANET can be used to enhance these requirements by various applications. VANET applications can be classified into four main categories:

1- Safety-related applications like: local danger warning, jam detection and accident avoidance.
2- Payment services: to simplify the payment process for the driver. Driver can use these applications to pay for toll-roads, parking, and fuel. This can be accomplished by transferring money from the driver account to the service-provider account.
3- Location-based services: to help the driver find an available parking place, the nearest gas station and so on.
4- Infotainment: car-to-car messaging, information download at gas stations and car-to-car information exchange such as points of interest [31].

Utilizing safety–related applications is the most important factor to adopt VANET and motivates drivers to use it. One of the most important applications is Local Danger Warning as described in [31]. In this application, vehicle generates messages on safety-related events, such as accidents, jams, emergency braking and so on. The message is sent to neighboring vehicles which will resend the message to their neighbors. Vehicle system will gather these messages from other vehicles and inform and warn the driver about the situation (see Figure 2-4).

If an accident occurs, inter-vehicle communication could support rescue teams to reach the accident site faster [33]. Moreover, stored messages at each vehicle can help to reconstruct the accident and determine liability of drivers.

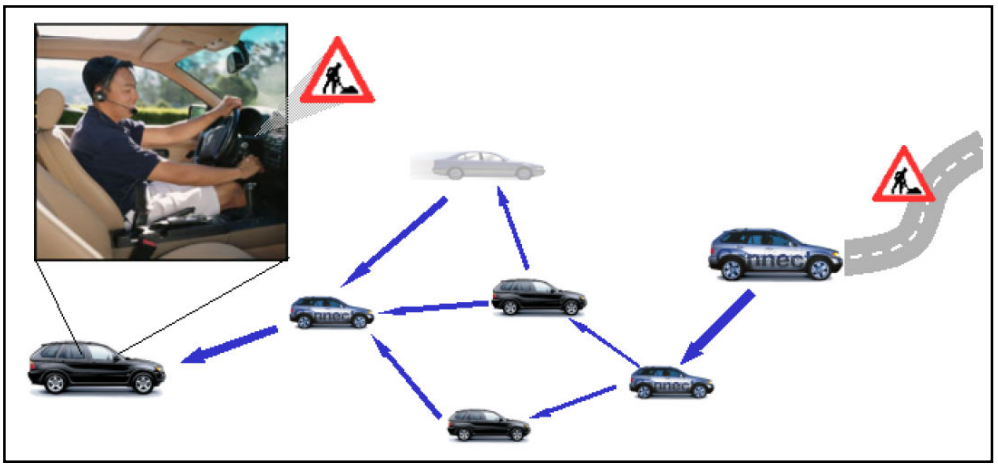

Figure 2-4: Local Danger Warning

The messages used in safety-related applications are referred to as safety-related messages or beacons [13]. Safety-related message contains information like location, speed and acceleration. They are sent periodically or when a hazardous situation occurs.

Safety-related messages, or shortly safety messages, must contain an authentication code. This code is needed to prevent attacks and protect the message. One possible method is to use the digital signature as authentication code. This code ensures that the sender of the message is who claims to be. Moreover, previous code ensures that the message was not modified.

2.5. Projects

Many VANET research projects were proposed like: FleetNet, NOW, WILLWARN, VSC-A, etc. However, only NOW and VSC-A concerned about privacy issue and tried to accommodate it. Also, some consortiums and societies were formed like: the U.S.A. Vehicle Safety Consortium [21], the Japan Road and Traffic Intelligence Society Organization [22], the Taiwan Intelligent Transport Society [23] and European Car-2-Car Communication Consortium [51]. Some details about previous projects are discussed below.

FleetNet – Internet on the Road [47] is a research project carried out by six companies and several German research institutes. It is partly funded by German government. The project duration was between 2000 and 2003. FleetNet project develops a wireless multi-hop ad hoc network for inter-vehicle communication. FleetNet evolved some applications such as cooperative driver assistance, decentralized floating car data and information services. Cooperative driver assistance can be used for emergency notifications and obstacle warning. Decentralized floating car data may be used for traffic jam monitoring by authorities. Information services examples are internet access and inter-vehicle chat. One of FleetNet tasks is to make a specification of radio protocols for mobile location-aware ad-hoc networks. Another task is to plan an implementation of prototype to demonstrate proof of concept.

Network on Wheels (NoW) [48] is a German research project implemented by many car manufactures, research institutes and universities. It was started in May 2004 and ended in May 2008. The NoW project is the successor of the research project FleetNet. The NoW project develops a vehicular communication platform for vehicle-to-vehicle and vehicle-to-infrastructure communications which is based on ad hoc principles and wireless LAN technology. These communications are used for road safety, traffic efficiency and infotainment applications.

WILLWARN (Wireless Local Danger Warning) [49] is a subproject of PReVENT project which is an integrated project funded by European Commission for road safety. WILLWRAN was a three years project. It develops safety applications that warn the driver when a critical situation occurs. The developments of project include systems such as on-board hazard detection, in-car warning management and decentralized warning distribution. The project was based on ad-hoc networks for vehicle-to-vehicle and vehicle-to-infrastructure communications. The system demonstrated was functioning well on rural road and highway scenarios.

The US Department of Transport and Vehicle Safety Communications 2 Consortium (VSC2) and many car manufactures started in December 2006 a tree-year project in the field of wireless based safety applications. The project is called VSC-A (Vehicle Safety Communications–Applications) [50]. The goal of project was to develop

and test communication-based safety systems. It was aimed to determine if these communications can make improvement to traditional safety systems. The project developed vehicle safety communication architecture, protocols, and messaging framework and standardize them.

2.6. Challenges

Adopting VANET encounters many obstacles. One of these is that, for a long time, only a small number of vehicles will be smart. Another obstacle is the negative perception which the drivers may have about such mechanism. They may feel that they can be monitored by other persons [30].

Another challenge is the possibility to locate and track a vehicle based on its transmitted safety messages to other vehicles or road-side units [32]. This happens because the safety message contains identity of the sender. Drivers need to preserve their location privacy. However, privacy preservation must not prevent liability determination by authorities.

The focus in this thesis is on securing safety messages and balancing between privacy requirement and liability existence.

Chapter Three: Literature Review on Anonymous Authentication Techniques in VANET

Vehicles in VANET broadcast beacons or safety messages periodically which contain information about road status and warnings. These messages should be authenticated to avoid altered messages or messages from illegal entities. Regular authentication reveals the sender identity. However, each driver in VANET needs to protect his privacy by hiding its vehicle identity which prevents tracking by other entities. The solution is to use anonymous authentication techniques which authenticate messages and protect privacy. Anonymous authentication techniques in VANET can be classified into: Pseudonym-based techniques, RSU-aided techniques, Group Signature techniques, Ring Signature techniques and Shared-key techniques.

3.1. Pseudonym-based techniques

Pseudonym is a pseudo identifier assigned to a vehicle. Each vehicle in VANET can have multiple pseudonyms [55]. Vehicle uses pseudonyms in its communications with other vehicles to hide its true identity. Vehicle does not use the same pseudonym all the time, but the vehicle changes it to prevent tracking. There is a legal authority which knows the correspondence between the vehicle pseudonyms and its true identity. There are many techniques that employ pseudonyms to preserve privacy. These techniques can be classified in the following sections.

3.1.1. Anonymous key pairs

In this technique, each vehicle is preloaded with a set of many public-private key pairs and their corresponding certificates. The key pairs are generated by the transportation authority and stored in vehicle TPD. Each certificate is authenticated by Certificate Authority (CA), and it does not contain any information about vehicle's true identity. The set of key pairs have to be renewed periodically (every year for example). In [3], each vehicle needs to use and store 43,800 key pairs per year.

Before sending a message, a vehicle chooses randomly one of the available anonymous certificates for signing the message. The signed message contains the signature and the anonymous certificate in addition to message itself. The receiver vehicle verifies the message by validating the anonymous certificate and checking the signature. The operation is illustrated simply as follows:

1- Vehicle N has a set of anonymous certificates. Each certificate $Cert_{Ni}$ contains corresponding public key Pub_{Ni} which is signed with CA's private key Prv_{CA}.

$SIG_{Ni} = Sign(Pub_{Ni}, Prv_{CA})$

$Cert_{Ni} = Pub_{Ni} \| SIG_{Ni}$

2- If vehicle N needs to send a message M, it chooses $Cert_{Ni}$ with key pairs (Pub_{Ni}, Prv_{Ni}) to produce secure message SM.

$SIG_M = Sign(M, Prv_{Ni})$

$SM = M \| SIG_M \| Cert_{Ni}$

3- When another vehicle receives message SM, first it verifies the signature of $Cert_{Ni}$ using CA's public key Pub_{CA} which is stored on TPD of each vehicle.
If $Verify(SIG_{Ni}, Pub_{CA})$ *is true, then accept* $Cert_{Ni}$

4- If certificate $Cert_{Ni}$ is accepted, the receiver vehicle verifies the signature of message by public key Pub_{Ni} attached with the certificate $Cert_{Ni}$
If Verify(SIGM, PubNi) is true, then accept M

The major disadvantages of this technique which uses pre-installed pseudonyms are the large storage space needed at every vehicle and the large size of Certificate Revocation List (CRL). CRL is a list of revoked certificates which is broadcasted to all vehicles. Each vehicle must discard any received safety message signed with a certificate that exists in CRL. If any vehicle's keys are compromised, all certificates of that vehicle are added to CRL which leads to a large size of CRL.

3.1.2. Silent Periods

Each vehicle updates it pseudonym between broadcasts to avoid tracking by other entities. However, this scheme is vulnerable to linkability which is the ability to find a relation between the new and old pseudonyms of the vehicle. The linkability depends on the temporal and spatial relation. The temporal relation is the relation between the times of pseudonyms. Spatial relation is relation between locations of the vehicle. The tracker can link the old and new pseudonyms to the same vehicle by comparing the order of the emerging times of the new pseudonyms with the disappearing time of old pseudonyms in adjacent locations.

A scheme was proposed in [6] to solve this problem. The scheme uses random silent period between the updates of pseudonyms. A silent period is defined as a transition period between the use of new and old pseudonyms. Silent period consists of constant period and variable period. The constant period is used to hide spatial relation between the vehicle's disappearing locations and emerging locations. The variable period hide the temporal relation between the vehicle's disappearing times and emerging times.

Paper [59] tries to enhance previous scheme by applying the silent period at crossroads, waiting for the traffic light, before entering or leaving the road or some other areas. However, previous schemes, which depend on silent period, suffer from a major problem. The vehicles need to broadcast safety messages periodically, but this requirement may be affected by a long silent periods or a large number of silent vehicles.

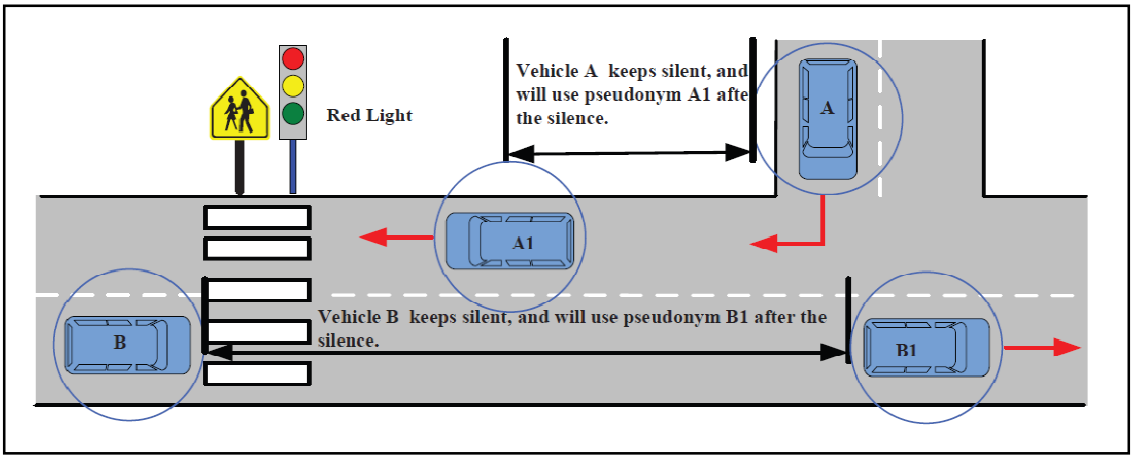

Figure 3-1: Silent Period at special points

3.1.3. Mix-zones

To mitigate linkability problem, other papers like [7] and [8] propose a scheme which uses mix-zones. In this scheme, adjacent vehicles belong to regions which called mix-zones. All vehicles of one mix-zone update their pseudonyms at the same time.

This scheme alleviate linkage problem. However, it is difficult to implement because it needs a precise synchronization between vehicles. Moreover, it needs a sufficient number of vehicles in each mix-zone.

3.2. RSU-aided Techniques

These techniques depend on RSUs in some operations like generation of pseudonyms, issuance of certificates and authentication of messages. These techniques can be classified into two categories: RSU-aided Anonymous Certificates and RSU-aided Message Authentication.

3.2.1. RSU-aided Anonymous Certificates

To decrease CRL size and minimize OBU storage overhead, papers [56], [57], [58] and [60] propose to use RSU-aided design to distribute short-time pseudonyms. In ECPP protocol proposed in [56], Each OBU issues a request for short-time anonymous key certificate from RSU when the vehicle passing by the RSU. The RSU will check if the vehicle is in the revocation list or not. If the vehicle identity exists in the revocation list, RSU will discard the request. Otherwise, the RSU will issue a short-time anonymous key certificate. When adding a vehicle to the revocation list, there is no need to add all of its certificates because their validity period is small and they will not be used after their lifetime expires. However, this scheme is not efficient when the number of nodes is very large. It consumes channel bandwidth and affects network performance.

3.2.2. RSU-aided Message Authentication

This technique is similar to previous one in that it depends on RSUs. An example of this technique is MAPWPP protocol which was proposed by Subhashree Behera et al. in [53]. MAPWPP is an RSU aided message authentication scheme which preserves privacy.

When a vehicle passes by RSU, it sends a request to the RSU for a temporary ID which is known as pseudoID. The vehicle can use this pseudoID till the vehicle enters the range of another RSU.

Each vehicle has a pre-installed private key which is used to sign its safety message. MAPWPP uses Elliptic Curve Digital Signature Algorithm (ECDSA) to sign message. The pseudoID is used by the vehicle in its safety messages instead of its true identity. After adding the pseudoID to safety message, the sender vehicle signs the message with its private key using ECDSA signature scheme.

When a vehicle receives a safety message, it extracts pseudoID and sends a request to RSU for the public key of that pseudoID. The RSU searches for true identity of the sent pseudoID and sends a replay containing the corresponding public key. After that, the receiver vehicle verifies the received safety message with received public key and thus authenticates the message. Therefore, the sender remains anonymous to other vehicles.

This technique depends heavily on RSUs to obtain pseudonyms and to authenticate messages. Furthermore, it consumes channel bandwidth and decreases the performance of the network.

3.3. Group signature techniques

Some researchers suggest techniques that use group signatures like in [9], [10] and [63]. In these techniques, vehicles are divided into groups. Each group has a group manager which is responsible for managing the group. Group signatures allow member vehicles of the group to sign their messages on behalf of the group. Every vehicle can verify the received signature with the sender group public key. However, no entity can reveal the true identity of the sender except the group manager. The main advantage of using group signature schemes is that they guarantee the unlinkability of the messages

GSIS protocol was proposed in [9] which is based on group signatures described in [61] and [62] and ID-based signatures descried in [28]. In this scheme, the group manager generates a group public key and issues private keys for vehicles. Each vehicle signs its

messages with the group public key and its private key. The receiver vehicle verifies the message with the group public key only and does not know the true identity of the sender. Only group manager can detect the true identity of a member vehicle.

This scheme suffers from some problems. If a vehicle is compromised or exits from the group, there is a need to change all vehicles' private keys and the group public key. Besides, the sizes of group signature and group public key depend on the group size. If the number of vehicles in the group is very small, it is difficult to achieve unlinkability.

3.4. Ring signature techniques

These techniques are based on Ring Signature which was introduced by Rivest, Shamir and Tauman [64]. Ring signature is a type of digital signature. In this scheme, a group of n entities forms a ring. Each entity i in the ring has a public and private key (Prv_i, Pub_i). When an entity i needs to send a message, it signs the message with its private key Prv_i and all public keys in the ring (Pub_1, … , Pub_n). The receiver can verify the signed message with the ring's public keys (Pub_1, … , Pub_n). Ring signature does not allow anyone to revoke the signer anonymity. However, Liu et al. [65] propose a variant for the ring signature, called revocable ring signature. This scheme allows a set of authorities to revoke the anonymity of the real signer.

Rings are differs from groups in that there is no manager and rings are formed randomly. The signer vehicle can form a ring arbitrarily and there is no need to member operations like add and delete [66].

Some researchers apply ring signature in VANET like [67] which provides traceability of illegal users besides to privacy and anonymity. These techniques suffer from a problem which is that the length of ring signature depends on the size of the ring.

3.5. Shared-keys techniques

These techniques use a shared key for all vehicles instead of using different pseudonyms for each vehicle. The trusted authority establishes a shared key and distributes it to all legal vehicles. Each legal vehicle uses that shared key to sign safety messages before sending and to verify received safety messages from other vehicles.

S3P protocol, described in [2], proposes to use a shared key pair which consists of public and private key. Each key pair is used for one period of time. To reduce communications, CA issues a set of shared key pairs called anonymity key set which contains a number of shared key pairs. Shared key pair in anonymity key set at index *i* is referred to as (Prv_{Ai}, Pub_{Ai}). Only legitimate vehicles can obtain an anonymity key set from CA.

For example, if CA chooses the size of anonymity key set to be 4, the anonymity key set will contain four key pairs with corresponding certificates as follows:

$$\{(Prv_{A1}, Pub_{A1}), (Prv_{A2}, Pub_{A2}), (Pub_{A3}, Pub_{A3}), (Prv_{A4}, Pub_{A4})\}$$

These keys will be used respectively for one period of time. If the key pair lifetime is chosen to be one week. The first pair (Prv_{A1}, Pub_{A1}) will be used by all vehicles in the first week. The second pair (Prv_{A2}, Pub_{A2}) will be used in the second week, and so on.

Each vehicle has a tamper-proof device (TPD) which is used to store secret keys and to execute cryptographic operations. Each vehicle has a pre-installed public-private key pair to communicate with CA. For vehicle *N*, the pre-installed public-private key pair is referred to as (Prv_N, Pub_N). Besides, there is an identity for each vehicle ID_N.

Before sending a safety message, each vehicle signs the message with the private key Prv_{Ai} of active shared key pair. Receiver vehicle verifies the message with the public key Pub_{Ai} of active shared key pair.

To achieve liability, the sender vehicle *N* signs the message with its private key Prv_N then encrypts the signature with CA's public key Pub_{CA} to produce *EP* packet. *EP* is included in the message before signing it with the shared key. Only CA can decrypt *EP* with its private key Prv_{CA} to get the real identity of the vehicle.

Securing safety messages steps are listed as follows:

1- TPD receives safety message m generated by OBU.
2- TPD produces m' adding timestamp t to m. $m'=m \| t$
3- TPD calculates signature Sig_N of the m' with the vehicle N's private key. $Sig_N=Sign(m', Prv_N)$
4- TPD encrypts Sig_N with the CA's public key Pub_{CA} and create encrypted packet EP. $EP=Enc(Sig_N, Pub_{CA})$
5- TPD generates packet M by concatenating m' and EP. $M=m' \| EP$
6- TPD produces signature Sig_A, over M, with the active private key Prv_{Ai} of the anonymity key set. $Sig_A=Sign(M, Prv_{Ai})$
7- TPD passes over M and Sig_A to OBU to broadcast the safety message.

If CA receives a report about misbehaving vehicle N, following steps are followed:

1- CA obtains the safety message ($M \| Sig_A$) generated by that vehicle.
2- CA extracts the EP from M.
3- CA calculates the Sig_N from EP by decrypting it with its private key Prv_{CA}. $Sig_N=Dec(EP, Prv_{CA})$
4- CA extracts the vehicle's certificate from Sig_N which contains its identity.
5- CA validates the signature Sig_N using the vehicle's public key Pub_N. $Verify(Sig_N, Pub_N)$
6- If the signature is valid, CA successfully identifies the vehicle.

After a misbehaving vehicle identity is detected, CA will revoke the vehicle by sending a revocation message to its TPD. When the TPD receives a revocation message which contains its identity, the TPD erases all keys and stops. Other vehicles will switch form shared keys to emergency keys.

This protocol eliminates the need to use multiple pseudonyms, but it has additional overhead. This protocol uses public key encryption one time and digital signature two times to secure safety message which consumes time and power. Furthermore, if a vehicle revoked, the shared keys need to be updated and distributed to all other vehicles.

Chapter Four: Privacy Preserving Secure Communication Protocol for VANET

4.1. Introduction

Vehicular Ad-hoc Network (VANET) consists mainly of vehicles and Road-Side Units (RSUs). Communication in VANET needs to be secure. However, vehicles privacy must be preserved. We propose a new protocol called Privacy Preserving Secure Communication Protocol (PPSCP). The proposed protocol fulfills previous requirements in an innovative way.

The proposed protocol depends on the existence of an entity called Certificate Authority (CA). CA is responsible for managing and distributing keys of the system. CA also manages and publishes revocation lists. Revocation list is a list of revoked keys. Any message uses revoked keys must be discarded.

Each vehicle is equipped with a Tamper-Proof Device (TPD). TPD stores cryptographic keys provided by CA. Besides, TPD performs any cryptographic operation needed by the protocol.

4.2. Notion Description

Notion used in this chapter are described in Table 4-1.

Table 4-1: Notion Description

Notion	Description
VID_N	Identity of vehicle *N*.
||	Concatenation symbol.
K_i	Shared key at index *i*, in shared key set.
KR_N	A revocation key of vehicle *N*
KR_{Ncurr}	Current revocation key of vehicle *N*.
KR_{Nnext}	Next revocation key of vehicle *N*.
Pub_{CA}	Public key of Certificate Authority CA.
Prv_{CA}	Private key of Certificate Authority CA.
Pub_N	Public key of vehicle *N*.
Prv_N	Private key of vehicle *N*.
K_C	Common key for all vehicles.
$Enc_{pub}(m, K)$	Encrypting *m* with key *K* using a public-key cipher.
$Dec_{pub}(m, K)$	Decrypting *m* with key *K* using a public-key cipher.
$Enc_{sym}(m, K)$	Encrypting *m* with key *K* using a symmetric-key cipher
$Dec_{sym}(m, K)$	Decrypting *m* with key *K* using a symmetric-key cipher
t	Timestamp
r_1, r_2	Secure random numbers
L	Revocation list
KR_j	Revocation key at index j in revocation list *L*
ET	Encrypted timestamp value.
EVID	Encrypted vehicle identity value.
SIG_N	Signature of Vehicle *N*.
SIG_{CA}	Signature of CA.
Sign(m, K)	Signing message *m* with key *K*
Verify(SIG, K)	Validating signature *SIG* with key *K*.
MAC	Message Authentication Code
$MAC_{algorithm}(m, K)$	Algorithm generates *MAC* of message *m* with key *K*.

4.3. Key Management

Each vehicle N has a unique identity called VID_N. CA is responsible for generating and installing VID_N on a vehicle before giving the vehicle a license to work. VID_N is stored in a tamper-proof device TPD which is installed on each vehicle. In our protocol, VID_N is chosen to be a value of 64-bit length. This length can represent more than 18 billion of billion values. Other lengths can be used.

To secure communications with Certificate Authority, CA has public-private key pair (Pub_{CA}, Prv_{CA}). CA also generates public-private key pair (Pub_N, Prv_N) for each vehicle N. CA preinstalls Pub_N, Prv_N and Pub_{CA} on each vehicle N's TPD in addition to VID_N . Public and private keys are generated according to the adopted public-key cipher which can be any of known public-key ciphers like RSA and ECC.

To present anonymity, CA generates periodically a set of keys called shared key set which contains n keys. These keys are used to authenticate safety messages between vehicles. Each key K_i in shared key set acts like a secret key for Message Authentication Code (MAC) algorithm. Many MAC algorithms can be used such as HMAC and CMAC. However, HMAC (hashed-MAC) is preferred here because it is more efficient and popular. Key size is selected here to be 128-bit which is a common size for symmetric ciphers like AES.

Each vehicle communicates with CA to get current or next shared key set. The vehicle N uses CA's public key Pub_{CA} to encrypt the message of shared key set request. CA replies with a message containing a shared key set with other data all are encrypted with N's public key Pub_N.

Each shared key K_i in shared key set is valid for a selected constant period. When the period ends, the next shared key K_{i+1} will be used at all vehicles. Before current shared key set period ends, each vehicle communicates with CA to get the next shared key set. If last shared key K_n period ends, all vehicles will use the next shared key set as the current shared key set.

For example, as in S3P protocol, if we choose shared key set size n to be 4 and choose shared key period to be one week, CA will generate 13 shared key set per year

because one year consists of 52 weeks approximately. Each shared key set will be used for 4 weeks and contains 4 shared keys. Therefore, CA will generate 52 shared key per year approximately.

If current or next shared key sets are compromised, CA will broadcast a warning message to all vehicles. Any vehicle receives that warning message will communicate with CA to refresh its current and next shared key sets.

The main goal of using shared keys K_i is to achieve security and privacy. Security is achieved by authenticating messages by Message Authentication Code (MAC). Using the same shared key by all vehicles at the same time to authenticate safety messages provides anonymity which achieves privacy. Only trusted entities can obtain shared key sets from CA. if an entity becomes not trusted, it will not be able to obtain new shared key sets from CA. Moreover, CA will send that entity's revocation keys to all vehicles to discard all messages from it.

The mobility of vehicles between different cities and countries managed by different CAs is a frequent situation. Therefore, CAs should be connected together in a network. When vehicle N enters the region of different CA, it will communicate with CA using its public key Pub_N and certificate. The new CA verifies vehicle's certificate from old CA. Then, the new CA sends a new shared key set to that vehicle.

4.4. Timestamps

Safety message should be protected against replay attack. Protection is achieved by adding time information to message called timestamp t. When a vehicle receives a message, it will check the validity of its timestamp t. TPD should have an internal clock, and synchronize it periodically from RSUs. TPD is responsible for adding timestamps to messages before sending them. Besides, TPD checks the validity of timestamps in received messages.

4.5. Hiding Vehicle Identity

For liability, vehicle's identity must be recognized from its safety messages. However, this requirement contradicts with privacy. Therefore, vehicle identity should be hidden from other entities. Only authorized entities like CA can determine vehicle's identity.

Each vehicle N stores its identity VID_N in its TPD. Besides, each vehicle is preloaded by a common key K_C which is stored at TPD. K_C is a symmetric key of 128-bit length. K_C is identical at all vehicles and is installed by CA.

When a vehicle N sends a safety message, TPD will add $EVID$ value to it. $EVID$ value is produced by concatenating VID_N to secure random number r_1 then encrypting all with the common key K_C to produce EVR value. Previous value, EVR, is encrypted with CA's public key Pub_{CA}. Random number r_1 is a 64-bit value generated randomly.

$$\boldsymbol{EVR = Enc_{sym}\ (VID_N \,||\, r_1, K_C)}$$

$$\boldsymbol{EVID = Enc_{pub}\ (EVR, Pub_{CA})}$$

Concatenating r_1 to VID_N every time before encryption ensures that $EVID$ value is different for every message. This method mitigates the linking between two successive messages from the same vehicle. Encrypting with K_C mixes the values of VID_N and r_1 which makes the partial decryption of $EVID$ value useless.

Only CA can reveal vehicle's identity by decrypting $EVID$ value. CA can use its private key Prv_{CA} to decrypt $EVID$ to produce EVR. Then, CA uses common key K_C to decrypt EVR to VID_N and r_1.

$$\boldsymbol{EVR = Dec_{pub}\ (EVID, Prv_{CA})}$$

$$\boldsymbol{VID_N \,||\, r_1 = Dec_{sym}\ (EVR, K_C)}$$

Pub_{CA}, Prv_{CA} and K_C should be renewed periodically. These keys lifetime should be long (a year for example). When a vehicle renews its license, CA will install the new keys Pub_{CA} and K_C on vehicle TPD. Pub_N and Prv_N of each vehicle N should be refreshed and renewed with previous keys.

4.6. Key Revocation

The ability to revoke vehicles is very important. When a misbehaved vehicle is detected, it must be revoked. To achieve revocation, each vehicle should have a key called revocation key KR_N. To revoke a vehicle, its revocation key will be sent to all vehicles. Then, they can discard messages from that revoked vehicle using its revocation key. If a revocation key is revealed, the revoked vehicle can be tracked. Hence, revocation key is updated periodically which makes tracking limited to last revocation key period.

When CA sends a shared key set to a vehicle N, it will also send the revocation key KR_N within the same message. KR_N is a unique key for each vehicle N and it is refreshed periodically with each new shared key set. If vehicle N requests current shared key set from CA, it will receive its current revocation key KR_{Ncurr} within the message. Otherwise, If vehicle N request next shared key set, it will receives its next revocation key KR_{Nnext} which will be used with next shared key set. Revocation key KR_N is a private value which means that it is unknown to other vehicles. Only CA knows all vehicles revocation keys. KR_N is chosen to be 128-bit length. KR_N is stored in TPD.

KR_N is used by vehicle N to encrypt timestamp t to produce IET value. Then, IET is encrypted with current shared key K_i to produce ET value. Timestamp t and ET value are added to safety message by TPD before sending it. Because t here is 64-bit, it is concatenated to a random number r_2 before encryption. Random number r_2 is 64-bit value. ET is generated as follows:

$$\boldsymbol{IET = Enc_{sym}(t \,||\, r_2, KR_N)}$$

$$\boldsymbol{ET = Enc_{sym}(IET, K_i)}$$

Encrypting with current shared key K_i ensures that only legitimate vehicles can try to decrypt ET value. That keeps revoked vehicles safe from tracking by illegal entities which do not have shared keys.

KR_N is used by CA to revoke validity of misbehaved vehicles. When a misbehaved vehicle is detected, CA will add its current and next revocation keys, KR_{Ncurr} and KR_{Nnext}, to revocation list L.

$$L = L + \{KR_{Ncurr}, KR_{Nnext}\}$$

Revocation list L is broadcasted to all vehicles. When a vehicle receives a safety message, it will try to decrypt ET using all revocation keys KR_j existing in L. if the decrypted value equals attached timestamp t, the message is discarded. For each KRj in L, this check is applied:

$$IET = Dec_{sym}(ET, K_i)$$

$$t_{rev} \,||\, r_2 = Dec_{sym}(IET, KR_j)$$

if ($t = t_{rev}$) then discard message.

Any vehicle N receives a revocation list L must check if its current or next revocation keys, KR_{Ncurr} and KR_{Nnext}, exist in L. if one of them at least exists in L, vehicle N's TPD will stop working and erase all shared and revocation keys.

This scheme provides conditional privacy. If vehicle is not revoked, then its privacy will be protected. For misbehaved vehicles, messages can be linked to those vehicles for the lifetime of revocation key.

When current shared key set period ends, revocation list L is updated by removing all current revocation keys and keeping next revocation keys. This method keeps L small and avoids performance issues.

4.7. Message Authentication

Before sending a safety message m, TPD will add values: t, $EVID$, ET and MAC. Message Authentication Code MAC is calculated using a MAC algorithm for the message consisting of: m, t, $EVID$ and ET. The key used in MAC algorithm is the current shared key K_i.

$$MAC = MAC_{algorithm}\,(m \,||\, t \,||\, EVID \,||\, ET, K_i)$$

When a vehicle receives a message, it recalculates *MAC* code and compares it with received *MAC* code. If they are not equals, the message is discarded.

$$MAC_{new} = MAC_{algorithm}(m \,||\, t \,||\, EVID \,||\, ET, K_i)$$

If (MAC_{new} <> MAC) then discard message.

Securing safety message by TPD before sending is depicted in Figure 4-1.

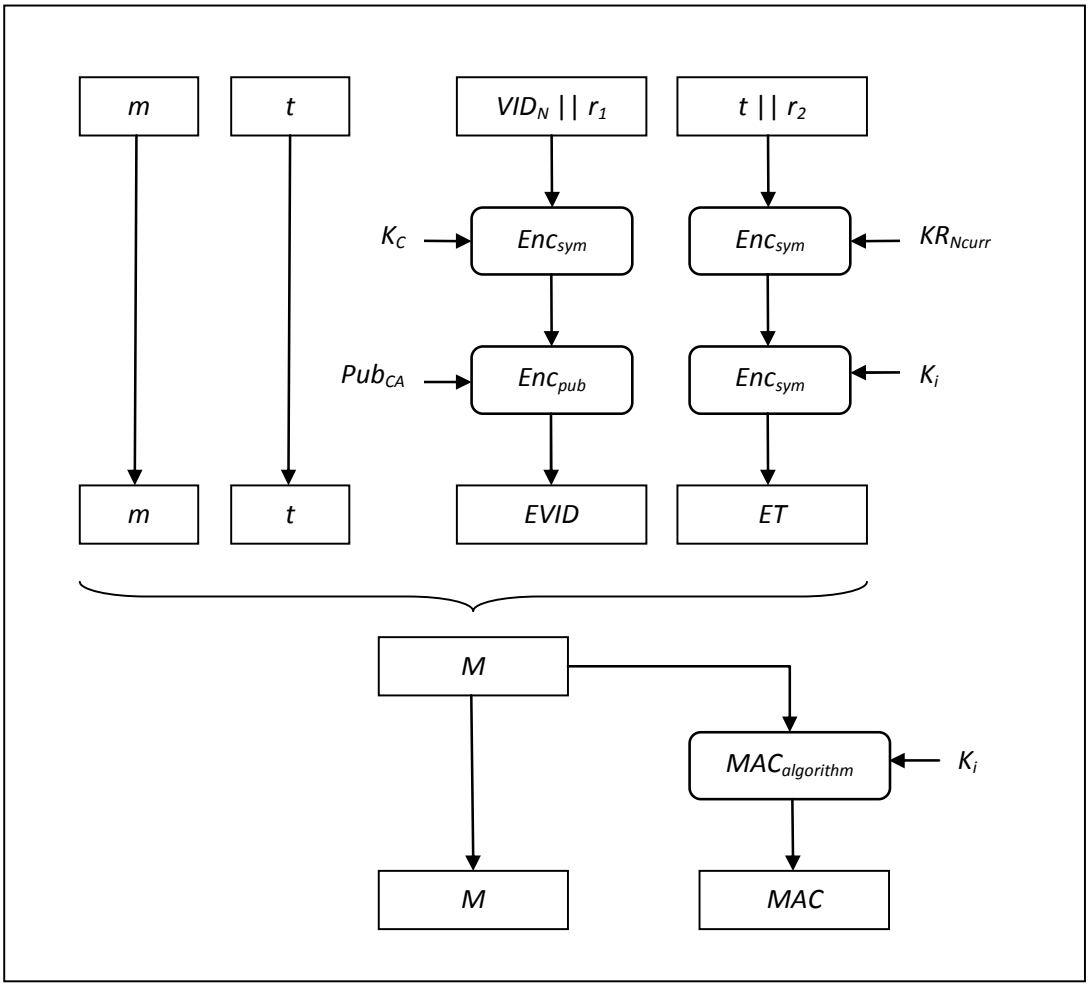

Figure 4-1: Securing safety message before sending.

MAC code ensures message authentication and integrity. Any change in the message will produce different *MAC*. Hence, message modification can be detected and integrity is ensured. If a sender does not use the same key of receiver, it is recognized that the message is not authenticated because *MAC* codes at sender and receiver does not match.

4.8. PPSCP Protocol

The proposed protocol is organized in algorithms. Some algorithms are used for vehicle-to-vehicle (V2V) communications such as securing safety messages and verifying received messages. Other algorithms are applied when vehicles communicates with infrastructure (V2I) such as requesting shared key sets. CA uses some algorithms for identifying and revoking misbehaved vehicles. Algorithms are described in the following sections.

4.8.1. Securing Safety Message

Before sending a safety message m from vehicle N to other nodes, it should be secured. Message m is generated by OBU according to information gathered from available sensors and GPS. OBU passes it to Tamper-Proof Device (TPD). TPD uses Algorithm 4-1 to secure safety message:

Algorithm 4-1: Securing Safety Message

Input: safety message m

Output: secure safety message SM

1: Get current timestamp t

2: Generate new random number r_1

3: $EVR = Enc_{sym}(VID_N \,||\, r_1, K_C)$

4: $EVID = Enc_{pub}(EVR, Pub_{CA})$

5: Generate new random number r_2

6: $IET = Enc_{sym}(t \,||\, r_2, KR_{Ncurr})$

7: $ET = Enc_{sym}(IET, K_i)$

8: $M = (m \,||\, t \,||\, EVID \,||\, ET)$

9: $MAC = MAC_{algorithm}(M)$

10: $SM = M \,||\, MAC$

11: return SM

Algorithm 4-1 steps are illustrated as follows:

1- TPD obtains current timestamp t.
2- TPD generates random number r_1.
3- TPD concatenates VID_N to r_1 to produce 128-bit value, then encrypts that value with common key K_C using a symmetric-key cipher Enc_{sym} and creates EVR.
$EVR = Enc_{sym}(VID_N \| r_1, K_C)$
4- TPD encrypts EVR with CA's public key Pub_{CA} using a public-key cipher Enc_{pub} and creates encrypted vehicle identity $EVID$.
$EVID = Enc_{pub}(EVR, Pub_{CA})$
5- TPD generates random number r_2.
6- TPD concatenates timestamp t to r_2 to produce 128-bit value, then encrypts previous value with vehicle N's current revocation key KR_{Ncurr} using a symmetric-key cipher Enc_{sym} and creates intermediate encrypted timestamp IET.
$IET = Enc_{sym}(t \| r_2, KR_{Ncurr})$
7- TPD encrypts IET with current shared key K_i using a symmetric-key cipher Enc_{sym} and creates encrypted timestamp ET. $ET = Enc_{sym}(IET, K_i)$
8- TPD concatenates m, t, $EVID$ and ET to produce M. $M = (m \| t \| EVID \| ET)$
9- TPD calculates Message Authentication Code MAC of M with current shared key K_i. $MAC = MAC_{algorithm}(M)$
10- TPD concatenates M to MAC to produce SM. $SM = (M \| MAC)$
11- TPD passes SM to OBU to broadcast it over the network.

4.8.2. Receiving Safety Message:

When a vehicle N receives a secured safety message SM from other vehicle, OBU passes the message to TPD. TPD verifies the message using Algorithm 4-2. If verification succeeds, TPD extracts m from SM and passes it to OBU, else TPD discards the message.

Algorithm 4-2: Verifying Secured Safety Message

Input: secured safety message *SM*.

Output: safety message *m* or *null*

1: Extract *M* and *MAC* from *SM*

2: $MAC_{new} = MAC_{algorithm}(M, K_i)$

3: if $MAC <> MAC_{new}$ then return *null*, stop

4: Extract timestamp *t* from *M*

5: if *t* is invalid then return *null*, stop

6: Extract *ET* from *M*

7: $IET = Dec_{sym}(ET, K_i)$

8: for each KR_j in *L* do

8.1: $TR = Dec_{sym}(IET, KR_j)$

8.2: Extract t_{rev} from *TR*

8.3: if $t = t_{rev}$ then return *null*, stop

9: Extract *m* from *M*

10: return *m*

Algorithm 4-2 is used to check the validity of the received message. It is described in the following steps:

1- TPD extracts *M* and *MAC* from *SM*.
2- TPD calculates new message authentication code MAC_{new} of *M* with current shared key K_i and then compares MAC_{new} with *MAC*.
3- If *MAC* and MAC_{new} match, the message is authenticated. Else, the message is discarded and steps are stopped.
4- TPD extracts timestamp *t* from *M*.
5- TPD checks *t* validity. If it is not valid, the message is discarded and steps are stopped.

6- TPD extracts *ET* from *M*.

7- TPD decrypts *ET* with current shared key K_i using a symmetric-key cipher Dec_{sym} to get intermediate encrypted timestamp *IET*.
$IET = Dec_{sym}(ET, K_i)$

8- For each revocation key KR_j in current revocation list *L*:
TPD decrypts *IET* with KR_j using a symmetric-key cipher Dec_{sym} to get *TR*, TPD extracts timestamp from $TR = (t \parallel r_2)$ and calls it t_{rev}
if t_{rev} equals *t*, the sender is revoked
if sender is revoked, the message is discarded and steps are stopped.

9- TPD extracts *m* from *M*.

10- TPD passes *m* to OBU for farther processing.

4.8.3. Identifying Vehicles

Vehicle identity is protected. Only CA can identify a vehicle from its messages. CA applies Algorithm 4-3 to identify a vehicle and gets its VID_N.

Algorithm 4-3: Identifying Vehicle

Input: secured safety message *SM*.

Output: VID_N or *null*

1: Extract *M* and *MAC* from *SM*

2: $MAC_{new} = MAC_{algorithm}(M, K_i)$

3: if $MAC <> MAC_{new}$ then return *null*, stop

4: Extract *EVID* from *M*

5: $EVR = Dec_{pub}(EVID, Prv_{CA})$

6: $VR = Dec_{sym}(EVR, K_C)$

7: *Extract* VID_N from *VR*

8: return VID_N

Algorithm 4-3 steps are described as follows:

1- CA extracts M and MAC from SM.
2- CA calculates new message authentication code MAC_{new} of M with current shared key K_i and then compares MAC_{new} with MAC.
3- If MAC and MAC_{new} match, the message is authenticated. Else, steps are stopped.
4- CA extracts $EVID$ from M.
5- CA decrypts $EVID$ with CA's private key Prv_{CA} using a public-key cipher Dec_{pub} and gets EVR. $EVR = Dec_{pub}(EVID, Prv_{CA})$
6- CA decrypts EVR with common key K_C using a symmetric-key cipher Dec_{sym} and gets VR. $VR = Dec_{sym}(EVR, K_C)$
7- CA extracts VID_N from VR. $VR = (VID_N \| r_1)$
8- CA identifies the vehicle N form VID_N.

4.8.4. Revoking Vehicle Validity

If a misbehaving vehicle is detected, a report containing messages belong to that vehicle is sent to CA. CA identifies that vehicle and gets its VID_N. CA uses VID_N to obtain from a lookup table the current and next revocation keys of vehicle N, KR_{Ncurr} and KR_{Nnext}. Then, CA adds them to current revocation list L.

Algorithm 4-4: Updating Revocation List

Input: secured safety message SM, revocation list L

1: VID_N = Algorithm_4-3(SM)

2: Get KR_{Ncurr} and KR_{Nnext} of vehicle N

3: $L = L + \{KR_{Ncurr}, KR_{Nnext}\}$

Algorism 4-4 as described as follows:

1- CA uses Algorithm 4-3 to identify the vehicle and gets its VID_N
2- CA searches in a lookup table by VID_N to get vehicle N's revocation keys KR_{Ncurr} and KR_{Nnext}
3- CA adds KR_{Ncurr} and KR_{Nnext} to revocation list L.

4.8.5. Sending Revocation Message

When CA identifies a misbehaving vehicle, revocation list L is refreshed by adding revocation keys KR_{Ncurr} and KR_{Nnext}. CA applies Algorithm 4-5 to generate revocation message RM, then broadcasts it over network by RSUs.

Algorithm 4-5: Sending Revocation Message

Input: revocation list L.

Output: revocation message RM

1: $EL = Enc_{sym}(L, K_i)$

2: Get current timestamp t

3: $M = (EL \| t)$

4: $SIG_{CA} = Sign(M, Prv_{CA})$

5: $RM = (M \| SIG_{CA})$

6: return RM

Algorithm 4-5 is described in the following steps:

1- CA encrypts L with current shared key K_i using a symmetric-key cipher Enc_{sym} and generates EL. $EL = Enc_{sym}(L, K_i)$
2- CA obtains current timestamp t.
3- CA concatenates EL with timestamp t to produce M. $M = (EL \| t)$
4- CA signs M with its private key Prv_{CA} and generates SIG_{CA}.

5- CA concatenates M with SIG_{CA} to produce RM. $RM = (M \| SIG_{CA})$

6- CA broadcasts revocation message RM to all vehicles.

4.8.6. Receiving Revocation Message

When a vehicle N receives a revocation message RM, OBU passes it to TPD. If TPD detects that its vehicle revocation keys KR_{Ncurr} or KR_{Nnext} exist in the message, then TPD erases all shared and revocation keys. If not, the TPD updates its revocation list. Algorithm 4-6 is applied when TPD receives a revocation messages.

Algorithm 4-6: Receiving Revocation Message

Input: revocation message RM.

1: Extracts M and SIG_{CA} from RM

2: if $Verify(SIG_{CA}, Pub_{CA}) = false$ then stop

3: Extracts timestamp t from M

4: if t is invalid then stop

5: Extracts EL from M

6: $L = Dec_{sym}(EL, K_i)$

7: if (L contains KR_{Ncurr} or KR_{Nnext})

7.1: then, erase all keys, turn off TPD

7.2: else, store L

Algorithm 4-6 steps are illustrated as follows:

1- TPD extracts M and SIG_{CA} from RM.

2- TPD verifies the message M and CA's signature SIG_{CA} with CA's public key Pub_{CA}. If verification fails, message is discarded and steps are stopped.

3- TPD extracts timestamp t from M.

4- TPD checks its validity. If timestamp is not valid, the message is discarded and steps are stopped.

5- TPD extracts *EL* from *M*.

6- TPD decrypts *EL* with current shared key K_i using a symmetric-key cipher Dec_{sym} and generates *L*. $L = Dec_{sym}(EL, K_i)$

7- If vehicle *N*'s revocation keys KR_{Ncurr} or KR_{Nnext} exist in *L*, TPD will erase all stored shared and revocation keys and turn off TPD. Else, TPD will update it current revocation list *L*.

4.8.7. Sending Request for Shared Key Set

Each vehicle starts by requesting current shared key set from CA. Before current shared key set period ends, vehicle *N* sends a request to CA for next shared key set. Request message is formed by TPD using Algorithm 4-7.

Algorithm 4-7: Sending Request for Shared Key Set

Input: *current* or *next*

Output: request message *REQM*

1: *req* = Input (*current* or *next*)

2: Get current timestamp *t*

3: $M = (req \,||\, t)$

4: $SIG_N = Sign(M, Prv_N)$

5: $REQM = (M \,||\, SIG_N)$

6: return *REQM*

Algorithm 4-7 receives type of requested shared key set (current or next). Then, it signs the request with vehicle *N*'s private key Prv_N. The steps are described as follows:

1- TPD generates *req* value according to input, current or next shared key set.

2- TPD obtains current timestamp *t*.

3- TPD concatenates *req* with timestamp *t* to produce *M*. $M = (req \,||\, t)$

4- TPD signs *M* with vehicle *N*'s private key Prv_N and generates SIG_N.

5- TPD concatenates M with SIG_N to produce $REQM$. $REQM = (M \,||\, SIG_N)$

6- TPD sends request message $REQM$ to CA.

4.8.8. Receiving Request for Shared Key Set

When CA receives a request for current or next shard key set from vehicle N, CA verifies the message using Algorithm 4-8.

Algorithm 4-8: Receiving Request for Shared Key Set

Input: request message $REQM$

Output: request req

1: Extracts M and SIG_N from $REQM$

2: if $Verify(SIG_N, Pub_N) = false$ then return $null$, stop

3: Extracts timestamp t from M

4: if t is invalid then return $null$, stop

5: Extracts req from M

6: return req

Algorithm 4-8 receives request message $REQM$, verifies it and returns request req which can be current or next according to requested shared key set. Its steps are illustrated as follows:

1- CA extracts M and SIG_N from $REQM$.

2- CA verifies the message M and vehicle N's signature SIG_N with vehicle N's public key Pub_N. If verification fails, message is discarded and steps are stopped.

3- CA extracts timestamp t from M.

4- CA checks its validity. If timestamp is not valid, the message is discarded and steps are stopped.

5- CA extracts req from M.

6- CA gets the value of req to use it for sending requested shared key set.

4.8.9. Sending Shared Key Set

After CA receives a request from vehicle *N* for current or next shared key set, CA sends it with current or next revocation keys of vehicle *N*. CA uses Algorithm 4-9 for sending shared key sets.

Algorithm 4-9: Sending Shared Key Set

Input: req, VID_N

Output: shared key set message KM

1: if req = *current* then

1.1: Let $KR_N = KR_{Ncurr}$ of vehicle N

1.2: Let K_1, K_2, K_3, K_4 = keys of current shared key set

1.3: else if req = next then

1.4: Let $KR_N = KR_{Nnext}$ of vehicle N

1.5: Let K_1, K_2, K_3, K_4 = keys of next shared key set

1.6: else return *null*, stop

2: $M = req \parallel K_1 \parallel K_2 \parallel K_3 \parallel K_4 \parallel KR_N$

3: Generate session key K_S

4: $EM = Enc_{sym}(M, K_S)$

5: Get current timestamp t

6: $EK = Enc_{pub}(K_S, Pub_N)$

7: $MM = EM \parallel t \parallel EK$

8: $SIG_{CA} = Sign(MM, Prv_{CA})$

9: $KM = (MM \parallel SIG_{CA})$

10: return KM

Algorithm 4-9 steps are explained as follows:

1- CA checks the request *req*. if *req* equals *current*, CA gets KR_{Ncurr} of vehicle *N* from lookup table using its VID_N and assigns it to KR_N. If *req* equals *next*, CA gets and assigns KR_{Nnext} to KR_N. Values K_1, K_2, K_3 and K_4 are set to shared keys of current or next shared key set according to request. If *req* does not equal *current* or *next*, steps are stopped.
2- CA concatenates request *req*, shared keys and revocation key of vehicle *N* to produce message *M*. $M = K_1 \| K_2 \| K_3 \| K_4 \| KR_N$
3- CA generates a session key K_S which is a symmetric key of length 128-bit.
4- CA encrypts *M* with K_S using symmetric-key cipher Enc_{sym} to produce *EM*. $EM = Enc_{sym}(M, K_S)$.
5- CA obtains current timestamp *t*.
6- CA encrypts K_S with vehicle *N*'s public key Pub_N using public-key cipher to produce *EK*. $EK = Enc_{pub}(K_S, Pub_N)$
7- CA concatenates *EM*, *t*, and *EK* to produce *MM*. $MM = EM \| t \| EK$
8- CA signs *MM* with its private key Prv_{CA} and generates SIG_{CA}.
9- CA concatenates *MM* with SIG_{CA} to produce *KM*. $KM = (MM \| SIG_{CA})$
10- CA sends *KM* to vehicle *N*.

4.9. Security Analysis

4.9.1 Authentication and integrity

Proposed protocol concerns with safety messages. Safety message is not encrypted because its information is not secret. Safety message contains general information like location, speed and acceleration. However, the protocol protects message integrity which ensures that safety message content is not tampered or altered. Furthermore, it provides authenticity which ensures that the received message is from a valid sender.

Message integrity and authenticity are achieved by adding Message Authentication Code *(MAC)* to the massage. *MAC* value is generated by a MAC algorithm which accepts

as input a secret key and a message. The strength of algorithm depends upon the size of the secret key that is used. Secret key used here is the current shared key K_i which is 128-bit length. This length is large enough to make brute-force attack impractical.

4.9.2. Replay attack

Proposed protocol prevents replay attack by adding timestamp *t* to safety message. When a vehicle received a message, its TPD will check the validity of timestamp *t*. This check is done be ensuring that the received message is sent recently. The check is valid only if internal clock of TPD is synchronized.

4.9.3. Privacy and liability

Proposed protocol preserves vehicle's privacy by hiding its identity. Moreover, the protocol provides liability by enabling authorized entities like CA to reveal vehicle's identity if necessary. Misbehaved vehicle cannot repudiate its messages which fulfills non-repudiation requirement.

This is done by encrypting vehicle's identity with CA's public key before sending of safety message. Thus, only CA can recognize vehicle's identity because no one has CA's private key. However, if vehicle *N*'s identity VID_N is encrypted alone, the result will be the same every time. The encrypted value can be linked to vehicle, so vehicle *N*'s identity VID_N is concatenated to random number r_1. Then, all are encrypted symmetrically with the common key K_C to obscure value. Finally, the result is encrypted with CA's public key Pub_{CA} to produce Encrypted Vehicle Identity *EVID*.

Concatenating random number r_1 to VID_N ensures that the result is different for every message sent by vehicle *N*. Hence, *EVID* in a safety message cannot be linked to vehicle *N*. Encrypting with the common key K_C prevents the possibility of partial decryption of *EVID* value. If adversary partially decrypts a part of *EVID*, he is not capable of recover VID_N without obtaining the other part.

4.9.4. Revocation

To make it possible to revoke validity of any vehicle, TPD adds Encrypted Timestamp *ET* value to safety message. Vehicle *N*'s TPD concatenates timestamp *t* to a random number r_2 and encrypts all with the current revocation key KR_{Ncurr}. The result is

encrypted with the current shared key K_i to eliminate adversary from trying to track revoked vehicles.

When validity of any vehicle is revoked by CA, then vehicle's revocation keys will be added to revocation list L. If any recipient vehicle can obtain t from ET using KR_j in L, then the received message is discarded. Moreover, that message can be linked to its revocation key KR_j. Any next message received from the same sender can be linked to it and the vehicle can be tracked. However, it is supposed that trusted vehicles will not try to track others, because all security operations are performed by TPD.

In the worst case, the protocol provides conditional privacy. Privacy of vehicle is preserved as long as its validity is not revoked which is similar to approach in [32]. Revocation keys are refreshed periodically when obtaining a new shared key set. Revealing vehicle's revocation key can affect on that vehicle privacy for a short period which is the lifetime of its revocation key.

Using revocation key KR_N to encrypt timestamp t is resistant to brute-force attack if KR_N size is large enough. KR_N is 128-bit value which needs 2^{128} attempts at maximum. For adversary from outside the network, it needs also to know current shared key K_i which is a key of 128-bit length. Brute-force attack by adversary to obtain these keys needs $2^{128} + 2^{128} = 2^{129}$ attempts not 2^{256}. That number is a result of Meet-in-the-middle attack.

Meet-in-the middle attack can be applied if a value is encrypted twice with two different keys. The attacker here should know a plain text and its cipher text. Plain text here is ($t \parallel r_2$), and cipher text is ET value. The attacker builds a lookup table containing all possible values of $Enc_{sym}(t \parallel r_2, KR_N)$. Then, the attacker tries to find the values of $Dec_{sym}(ET, K_i)$ in lookup table for all possible values of K_i, value by value. If there is a match, the correct keys are revealed.

$$\boldsymbol{Enc_{sym}(t \parallel r_2, KR_N) = IET = Dec_{sym}(ET, K_i)}$$

Meet-in-the middle attack is impractical here. It needs a huge storage of data for saving lookup table. The storage size should be of size $16 * 2^{128}$ bytes (approximately $5.4 * 10^{39}$).

4.9.5. Denial-of-Service Attack

DoS attack is one of the most dangerous attacks which can affect VANET communications. It is very difficult to prevent, especially in a wireless medium. To mitigate DoS, we propose to use the solution described in [3]. The protocol switches between different channels. Besides, protocol may switch between different communication technologies such as DSRC, cellular or Bluetooth. When the default communication (DSRC) is down, one of them can be used. If communications in VANET becomes not available, vehicles will turn off safety applications and inform the driver until communications come back again.

4.10. Performance

The proposed protocol uses symmetric and asymmetric ciphers. Symmetric encryption and decryption is much faster than public-key operations. The protocol uses only one public-key encryption when sending a safety message. Receiving safety message does not need public-key decryption by receiver vehicle. Public-key decryption is used only by CA when there is a need to reveal vehicle identity. This has a good impact on performance compared to other protocols. For example, the protocol described in [3] performs one public-key operation before sending and another one after receiving a safety message. Proposed protocol in [2] performs three public-key operations before sending and one operation after receiving a message.

Proposed protocol uses public-key encryption to generate *EVID* value before safety message is sent. Unlike other protocols, public-key operation does not depend on safety message content. To increase performance, TPD can initialize many *EVID* values. Every day and before vehicle drives on road, TPD can generate many *EVID* values and save them. Before sending, TPD uses a saved *EVID* value from storage instead of generating a new one. Number of *EVID* values needed every day equals to average number of safety messages sent per day. If we assume that an average vehicle drives 2 hours per day, and number of safety messages sent per second is 3, then the number of needed *EVID* values is ($2 \times 60 \times 60 \times 3 = 21600$) values. If *EVID* value size is 256 byte, all values amounts to

around 5.2 MB. If it needs 1 ms to generate one *EVID* value, all values need around 22 seconds to be generated.

In this protocol, Message Authentication Code (MAC) Algorithm is used to sign safety messages. This increases performance, if it compared to other protocols which uses public-key operations for signing. Like digital signature, *MAC* code provides data integrity and authentication but its size is smaller.

Revocation list used here is a list of keys which belong to invalid vehicles. Revoking validity of a vehicle does not add more than two keys to revocation list. When any revocation key period ends, it is removed from list. That keeps revocation list small and does not cause scalability issues. Revocation list is broadcasted to all vehicles periodically or when it is refreshed.

When vehicles communicate with CA to update shared key set in a small amount of time, a performance issues may arise. Therefore, shared key set lifetime should be large enough to reduce communication congestion. Another solution is that CA encrypts next shared key set with current shared key K_i and broadcast the encrypted set to all vehicles. However, revocation key KR_N still needs to be transmitted separately to each vehicle.

Chapter Five: Simulation and Results

5.1. Introduction

Testing networking protocols in real world is important but expensive. In VANET for example, testing a new protocol needs a large number of vehicles equipped with special devices participating in the experiment. This is difficult and costs so much time and money. Safety of driver is important, but this experiment can expose their lives to dangers.

To avoid previous issues, researchers developed simulators. Simulator is a computer program that models a real-life situation. For computer networks, there are many network simulators like ns2, OPNET and NetSim. GloMoSim and ns2 are the most popular simulators for wireless networking. Other simulators are specialized in simulation of Wireless Ad hoc Networks like JiST/SWANS. Vehicular Ad hoc Network (VANET) is a type of Wireless Ad hoc Networks.

Some researchers in Northwestern University extended and customized JiST/SWANS to work with VANET. They called their extension SWANS++ which are used here to test the proposed protocol performance.

5.2. Simulator

SWANS++ simulator is chosen here in this thesis to test proposed protocol performance. SWANS++ is an extension to JiST/SWANS simulator which is consists of two parts JiST and SWANS. Brief descriptions of these simulators are provided in the following sections.

5.2.1. JiST

JiST is a discrete event simulation engine based on Java [68]. JiST stands for Java in Simulation Time. It is created by Rimon Barr at Cornell University. JiST is designed to be efficient, transparent and to use a standard language. It is efficient because it out-performs prior highly optimized simulators both in time and memory consumption. JiST transparently transforms simulation code automatically to run with simulation time semantics. Simulation code need not be written in a domain-specific language invented

specifically for JiST. Simulation code is written in plain Java and executed on unmodified virtual machine. Therefore, JiST is considered as a virtual machine-based simulator.

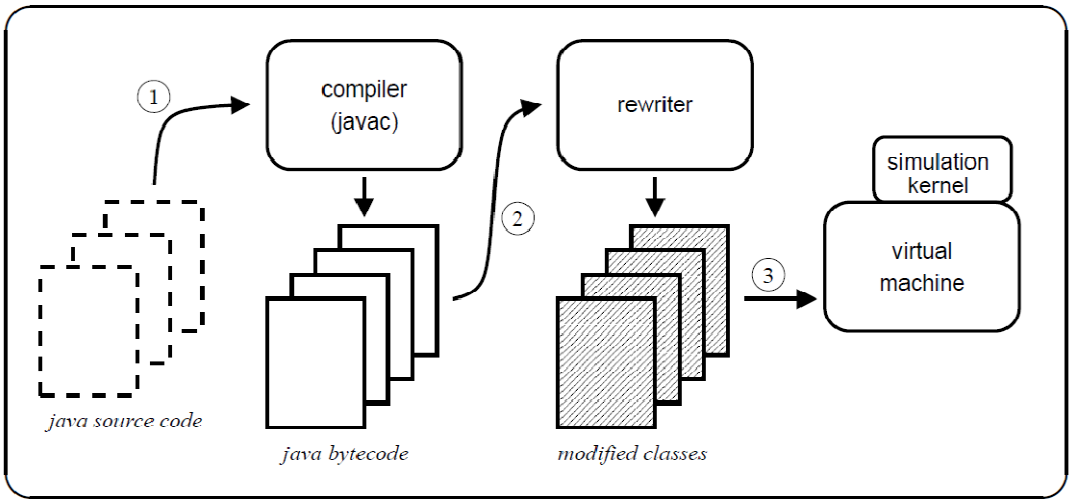

Figure 5-1: JiST Architecture

JiST architecture is described in Figure 5-1. It consists of four components: compiler, bytecode rewriter, simulation kernel and virtual machine. First, simulation program source code is written in a plain Java. Then, the source code is compiled by Java compiler to a Java bytecode which is stored in files called Java classes. Before running, bytecode rewriter modifies previous classes to run over a JiST simulation kernel and to support the simulation time semantics. Finally, the simulation kernel executes the modified classes using Java Virtual Machine (JVM). Simulation program, bytecode rewriter and simulation kernel are all written in pure Java.

This approach has many benefits. It allows simulator users to reuse any program from the large number of existing programs written in Java. Besides, standard libraries of Java can be used. Users can choose from existing Java compilers. JiST simulator engine utilizes Java language properties like automatic garbage collection, type-safety, reflection and many others. Furthermore, using Java reduces the time needed by users to learn and use JiST simulator.

5.2.2. SWANS

SWANS is a Scalable Wireless Ad hoc Network Simulator built atop the JiST platform, a general-purpose discrete event simulation engine [69]. SWANS simulator consists of event-driven components that can be configured and composed to form the desired wireless network simulation. It is similar to ns2 [70] and GloMoSim [71], but it is able to simulate much larger networks.

SWANS components are depicted in Figure 5-2.

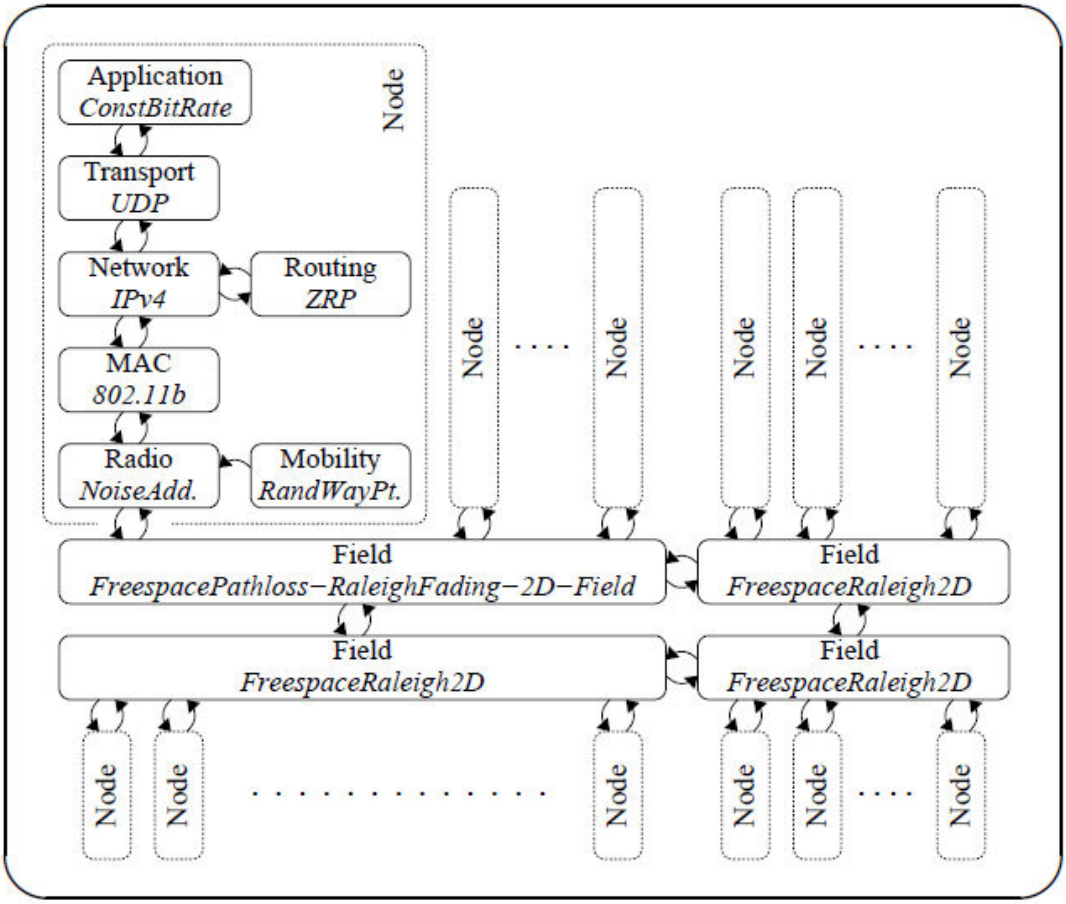

Figure 5-2: SWANS Architecture

Simulation program executed by SWANS should contain at least one field. The field represents a two dimensional area that affects radio signals. The field includes a number of wireless nodes. Each node consists of components. Some components can be

used to build node stack of layers like Application, Transport, Network and Link components. Other components have other usage like Routing and Mobility components.

5.2.3. SWANS++

SWANS++ was developed by Northwestern University as an extension to JiST\SWANS simulator to be used in Car-to-Car Cooperation (C3) project [72]. C3 project is concern with exploring and applying Vehicular Ad hoc Network (VANET). C3 project proposes an intelligent transportation system that does not rely on any infrastructure installed on the roads. It is supposed to be scalable, fault-tolerant and adaptable. The system adopts a cooperative model depending solely on the information collected by vehicles sensors and on information gathered from nearby vehicles obtained by short-range communication.

SWANS++ simulator adds many new features to JiST/SWANS simulator like runtime visualization, GPSR implementation, new DSR implementation and new mobility models for streets called STRAW.

Visualization tool used in SWANS++ is called Ceratias (see Figure 5-3). Ceratias is a tool for generically visualizing and steering an ongoing simulation [73]. It is built to be generic by providing basic abstractions for visualization, such as nodes icons, nodes colors, transmission circles and display of text based output. It enables the user to change certain visualization features while it is running like zoom level. Furthermore, it provides interface to access and change simulation settings at runtime which is a functionality added by SWANS++ and called simulation steering. Ceratias shows the visualization interactively while the simulation is running. It does not use or create trace files. It does not wait and start after simulation finishes, but it starts immediately after simulation starts.

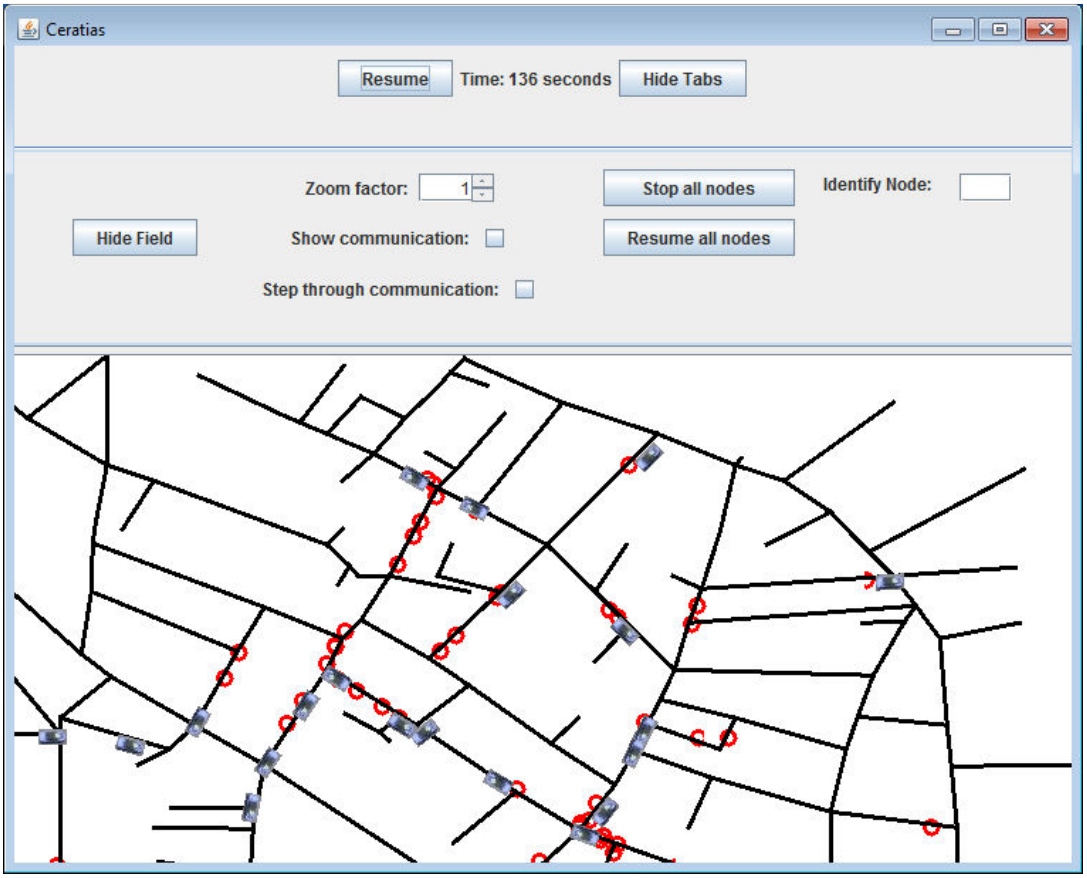

Figure 5-3: Ceratias visualization tool

5.2.4. STRAW Mobility Model

Modeling node mobility is very important in wireless ad hoc networks. JiST/SWANS simulator implements static and Random Waypoint mobility models. In Random Waypoint model, each node picks a random location (waypoint) and travels towards it with some random velocity. After the node arrives at the destination, it pauses for a predetermined period of time and travels towards another randomly selected location.

For VANET simulation, using of Random Waypoint mobility model leads to inaccurate results. Vehicles mobility is constrained to streets. SWANS++ extension inventors create a new mobility model for VANET called STRAW.

STRAW stands for STreet RAndom Waypoint [74]. It provides more accurate simulation results by using a vehicular mobility model on real cities. It is based on the behavior of real vehicular traffic. It constraints vehicles mobility to streets defined by maps for real cities. It bounds vehicles mobility according to traffic congestion and simplified traffic control mechanisms.

5.3. Cryptographic Algorithms

Our proposed protocol PPSCP needs to use some cryptographic algorithms. PPSCP needs the following:

1- Symmetric-key cryptographic algorithm.
2- Public-key cryptographic algorithm.
3- Cryptographic Hash function.
4- Message Authentication Code (MAC) Algorithm
5- Digital Signature Scheme

5.3.1. Symmetric-key cryptographic algorithm

PPSCP uses 128-bit keys for symmetric-key cryptographic operations. Besides, PPSCP needs to encrypt blocks of 128-bit length. We choose AES-128 because it is standard and popular for that purpose.

AES stands for Advanced Encryption Standard. AES is a symmetric-key block cipher announced by the National Institute of Standards and Technology (NIST) in December 2001. AES is published by NIST as FIPS 197 which stands for Federal Information Processing Standard [76]. AES was developed as a replacement for DES. AES cipher encrypts and decrypts data blocks. Each data block size is 128 bits. AES key size can be 128, 192 or 256 bits. According to key size, AES has three versions: AES-128, AES-192 and AES-256. AES has two ciphers: one for encryption and the other for decryption which is referred to as the reverse cipher (See Figure 5-4).

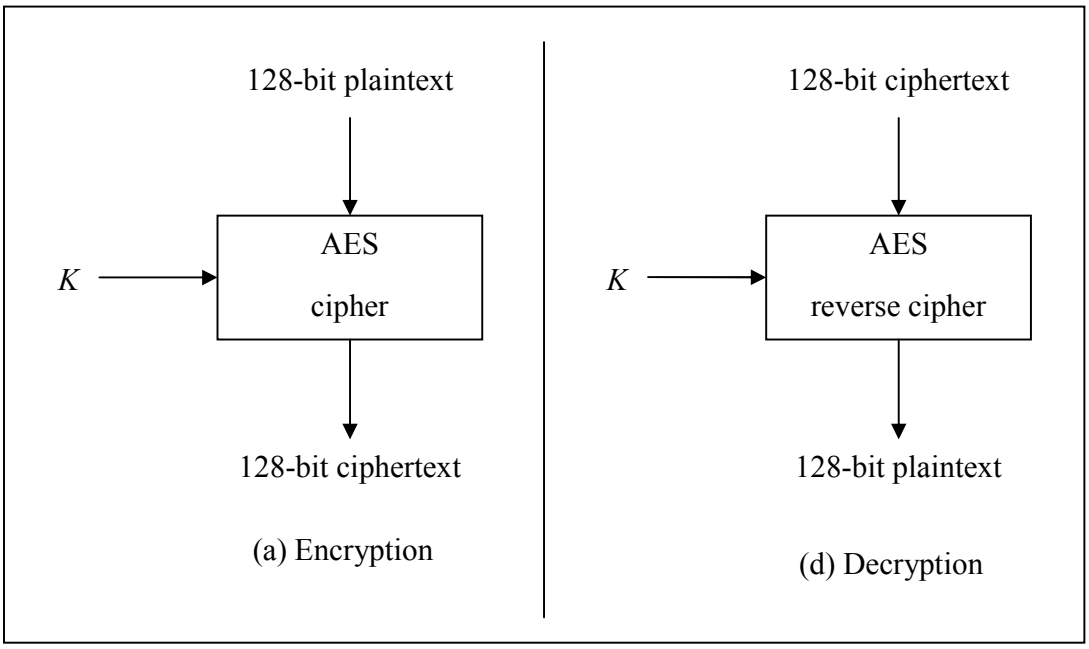

Figure 5-4: AES Encryption/Decryption

5.3.2. Public-key cryptographic algorithm

Public-key cryptography is called also asymmetric-key cryptography. Public-key cryptography uses two separate keys: private key and public key. Private key is kept secret whereas public key is published. If one of them used in encryption, then the other one is used in decryption. (See Figure 5-5)

There are many public-key cryptographic algorithms like: RSA, Rabin, ElGamal and Elliptic Curve Cryptosystems (ECC).

The most common public-key cryptosystem is RSA. It stands for its inventors names: Rivest, Shamir and Adleman. The security of RSA is based on the factorization problem which refers to the difficulty of factoring a very large number. There is no yet an efficient factorization algorithm. RSA algorithm multiplies two large prime numbers p and q to produce a very large number n called the modulus. it is difficult to factorize n into p and q.

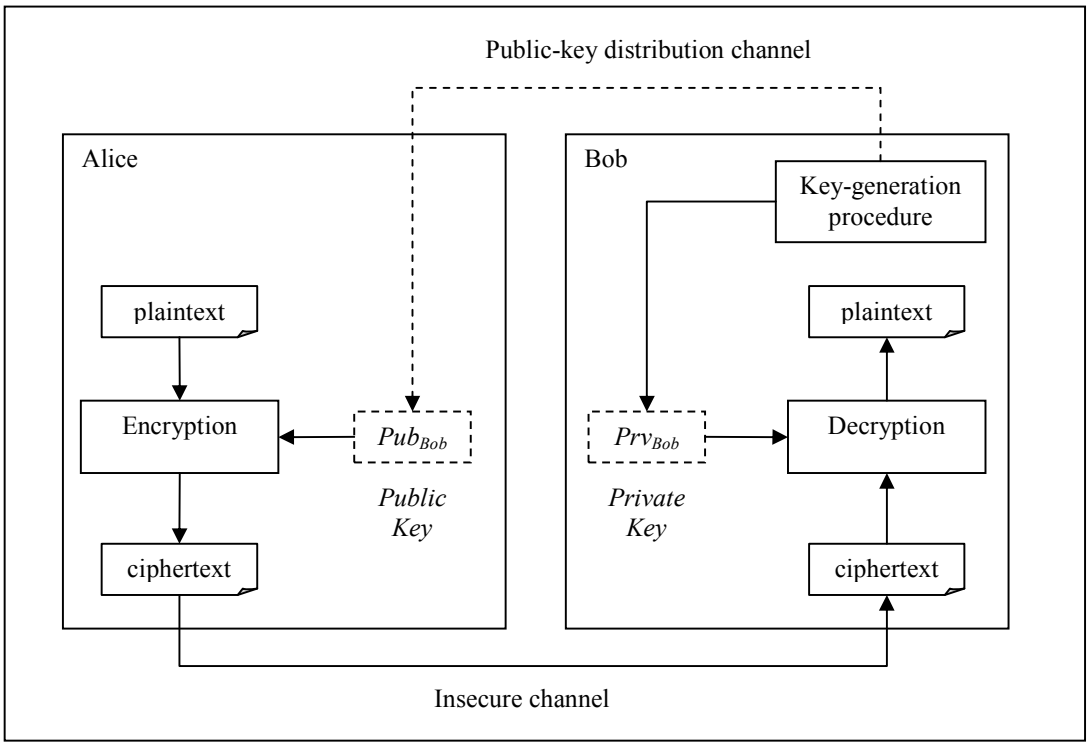

Figure 5-5: Public-key cryptography idea

RSA is classified according to the modulus number *n* size. RSA-1024 uses modulus of size 1024 bits. In 2009, some researchers factorized RSA-768 in two years by hundreds of workstations [77]. Factoring a 1024-bit RSA modulus would be about a thousand times harder.

We choose here to use RSA as a public-key cryptographic algorithm in simulation because of its simplicity and popularity. RSA is easy to implement and well described. However, the biggest disadvantage of RSA is its large key size which consumes more power and needs so much time. Trend today is to use other alternatives like Elliptic Curve Cryptosystem (ECC) which uses smaller key sizes with the same security level. ECC of 160 bits has the same security level of RSA with a key size of 1024 bits [53].

5.3.3. Cryptographic Hash function

Cryptographic hash function is one-way function that takes a message of arbitrary length and returns a fixed-length value which is called hash value or message digest. If any bit of message is changed, the hash value will change. It is infeasible to modify the message without affecting the hash value. Many security algorithms, like digital signature and Message Authentication Code (MAC), uses cryptographic hash functions.

The most popular cryptographic hash function is MD5, SHA-1 and SHA-2. MD5 is version 5 of Message Digest algorithm which is described in RFC 1321 [78]. It divides the message into blocks of 512 bits and creates a 128-bit digest. The digest size is too small and prone to collision attack. This attack is achieved by finding a different message with the same digest.

SHA stands for Secure Hash Algorithm and published by NIST as FIPS-180-2 [79]. SHA-1 uses blocks of size 512 and generates a 160-bit digest. SHA-1 is popular and widely used. SHA-2 has 4 versions: SHA-224, SHA-256, SHA-384 and SHA-512. Table 5-1 shows block and digest size of some cryptographic hash functions.

Table 5-1: Block and digest size of some cryptographic hash functions

Algorithm	Block size (bits)	Digest size(bits)
MD5	512	128
SHA-1	512	160
SHA-224	512	224
SHA-256	512	256
SHA-384	1024	384
SHA-512	1024	512

5.3.4. Message Authentication Code (MAC) Algorithm

MAC algorithm is a cryptographic algorithm that accepts a message and a key as inputs and produce MAC value as output. MAC value is used to authenticate the message and ensuring its integrity. (See Figure 5-6)

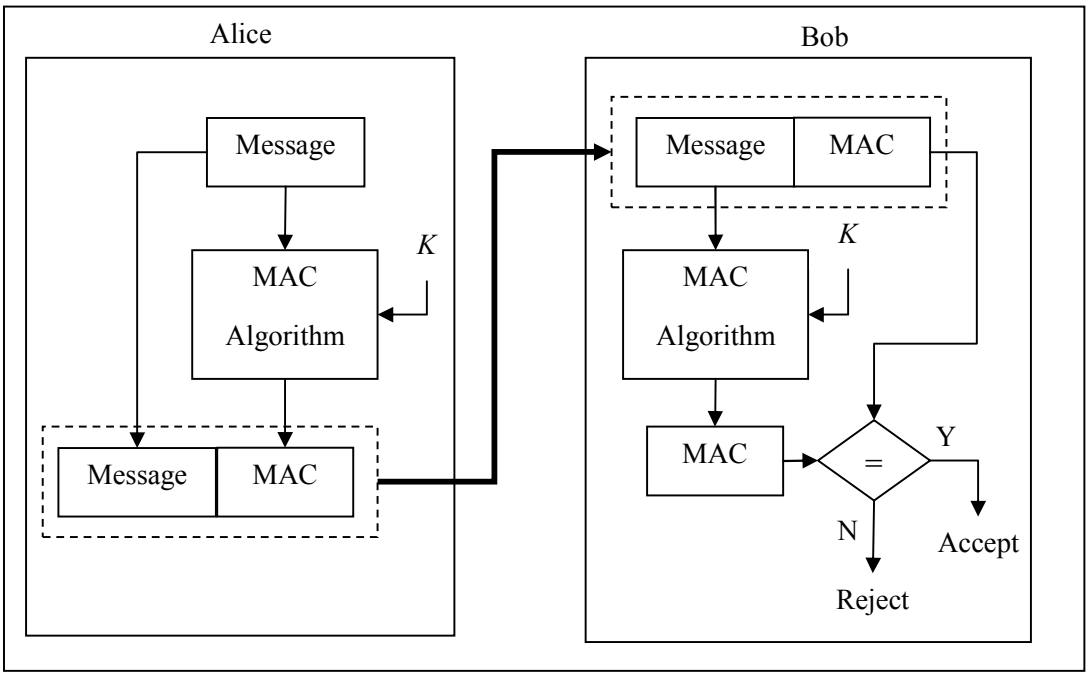

Figure 5-6: Message Authentication Code idea

The used key is secret and shared only between the sender and receiver of the message. The sender adds MAC value to message before sending it. When receiver gets the message, he recalculates MAC value and compares it with arrived one. If they are matches, the message is not modified, and it is sent from the claimed sender.

There are many MAC algorithms like HMAC and CMAC. HMAC stands for hashed MAC. It is issued by NIST as FIPS-198 [80]. HMAC uses an underlying hash function. Any hash function can be used like MD5 and SHA-1. HMAC is referred to by its hash function like HMAC-MD5 and HMAC-SHA-1. (See Figure 5-7)

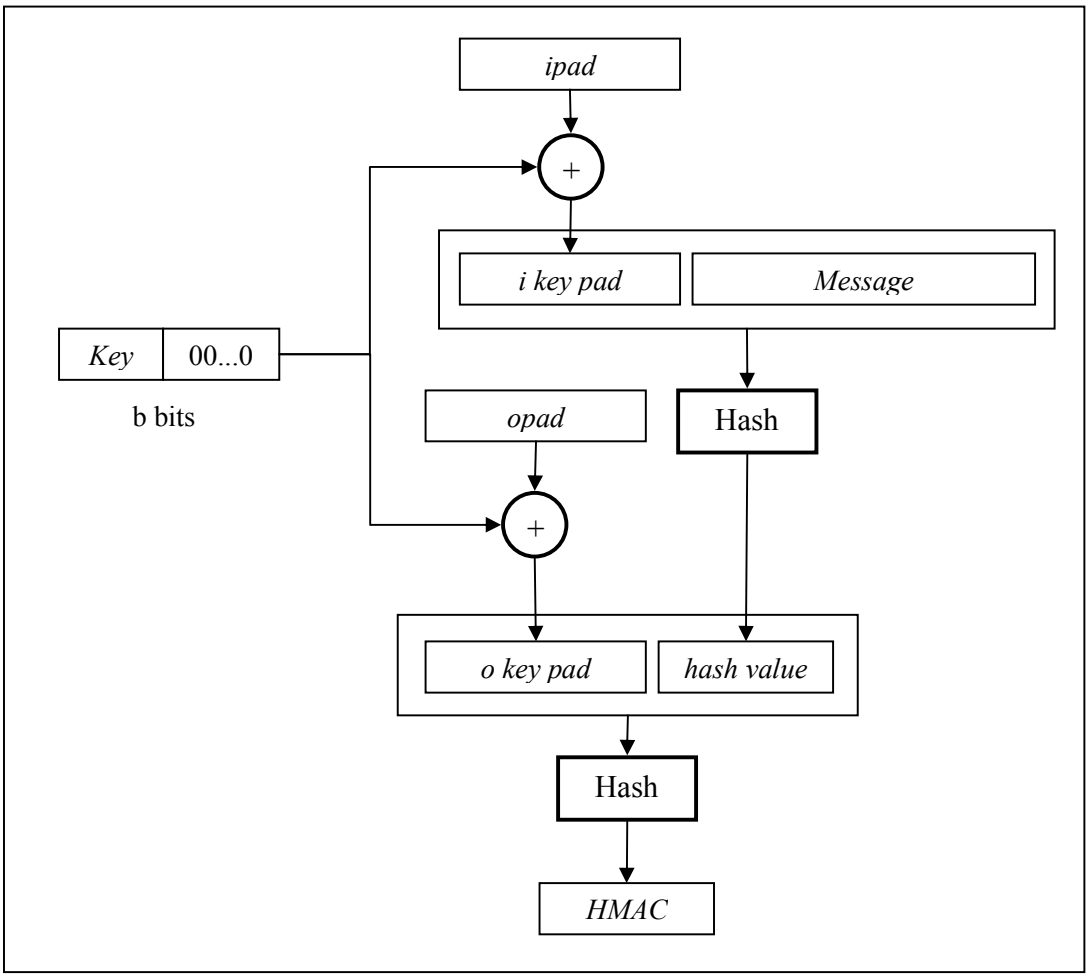

Figure 5-7: HMAC Algorithm

CMAC stands for Cipher-based MAC which is published by NIST in [81]. CMAC depends on a block cipher algorithm like AES. It divides the message into blocks. The first block is encrypted with the key. The result is XORed with next block and encrypted again with the key. We choose here to use HMAC because it is more efficient and popular.

5.3.5. Digital Signature Scheme

A digital signature scheme is a cryptographic method to ensure message authenticity and integrity through using public-key cryptography. Sender signs the message by encrypting message hash with his private key to produce signature. Message is sent with the signature and certificate which contains information about the sender like his name and public key. The receiver verifies the message by decrypting the signature with sender

public key and comparing it with message hash. If they are match, the message is accepted. (See Figure 5-8)

There are many digital signature schemes like: RSA Digital Signature scheme, ElGamal Digital Signature scheme and Digital Signature Standard (DSS). DSS was adopted by NIST in 1994. NIST published DSS as FIPS 186 [83]. DSS uses Algorithm called DSA which stands for Digital Signature Algorithm. DSA is a variant of ElGamal Digital Signature scheme. RSA Digital Signature is simpler and very similar to RSA cryptography. We select RSA Digital Signature scheme to be used in the simulation.

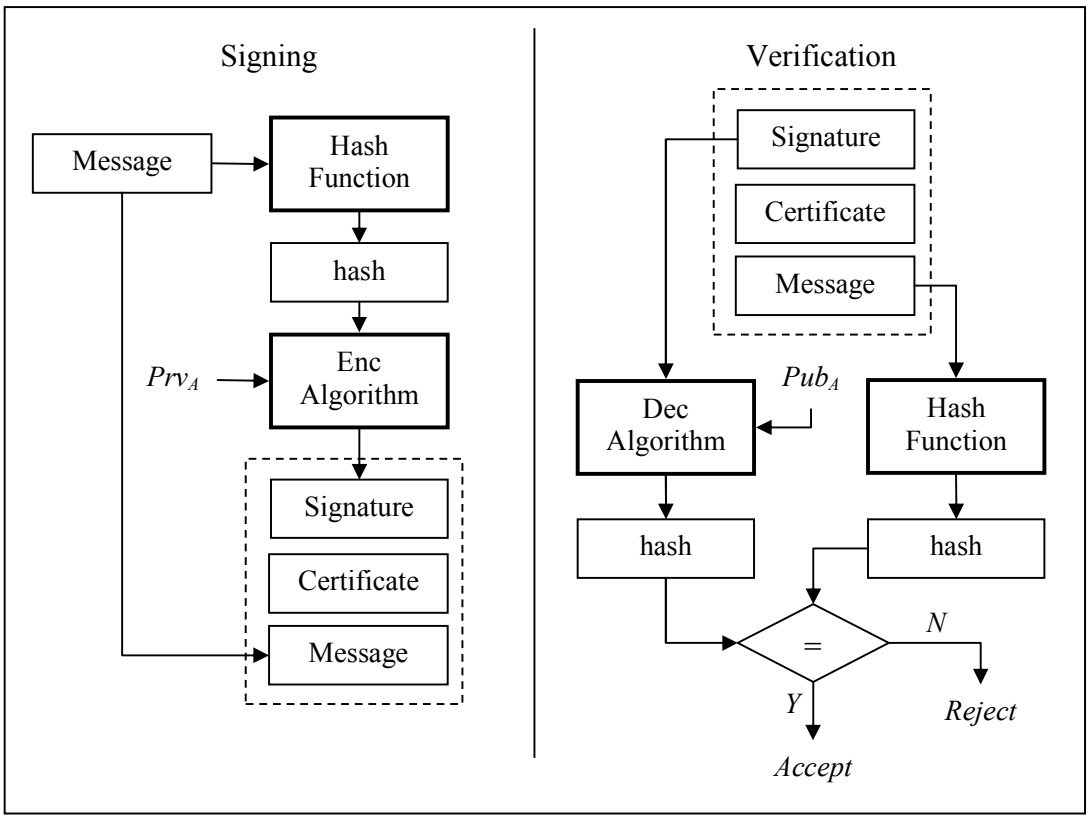

Figure 5-8: Digital Signature

5.4. Benchmarks

The simulation of proposed protocol PPSCP needs to use a speed benchmark for selected cryptographic algorithms. In [75], many cryptographic algorithms are tested on three different machines:

1- Intel Pentium 4 (Prescott) processor. Algorithms are coded in C++ and compiled with MS Visual C++ 2005 SP1. Operating system is Windows Vista 32-bit.
2- Intel Core 2 1.83 GHz processor. Only one core of the CPU was used. Algorithms are coded in C++ and compiled with MS Visual C++ 2005 SP1. Operating system is Windows Vista 32-bit.
3- AMD Opteron 8354 2.2 GHz processor. Algorithms are coded in C++ and compiled with GCC 4.1.2. Operating system is Linux

Table 5-2 shows the speed of AES-128 and HMAC-SHA-1 on three machines. The speed here is measured by megabytes encrypted or hashed per second.

Table 5-2: AES and HMAC Result

MB/Second	Intel Pentium 4 2.93 GHz	Intel Core 2 1.83 GHz	AMD Opteron 8354 2.2 GHz
AES/ECB (128-bit key)	106	109	153
HMAC(SHA-1)	138	147	187
SHA-1	138	153	192

Table 5-3 and Table 5-4 show the speed of RSA-1024 and RSA-2048 respectively on the selected machines. They show the time needed by RSA operations. Operations may be encryption, decryption, signing or verifications.

Table 5-3: RSA-1024 Results

Millisecond/Operation	Intel Pentium 4 2.93 GHz	Intel Core 2 1.83 GHz	AMD Opteron 8354 2.2 GHz
RSA 1024 Encryption	0.09	0.08	0.04
RSA 1024 Decryption	2.28	1.46	0.67
RSA 1024 Signature	2.25	1.48	0.67
RSA 1024 Verification	0.10	0.07	0.04

Table 5-4: RSA-2048 Results

Millisecond/Operation	Intel Pentium 4 2.93 GHz	Intel Core 2 1.83 GHz	AMD Opteron 8354 2.2 GHz
RSA 2048 Encryption	0.22	0.16	0.08
RSA 2048 Decryption	10.53	6.08	2.90
RSA 2048 Signature	10.64	6.05	2.91
RSA 2048 Verification	0.22	0.16	0.08

We choose Pentium 4 benchmark results because CPUs installed on vehicles have lower performance than those used in desktop computers.

5.5. Implementation

JiST/SWANS Simulator consists of components. We added two components: Safety Manager and TPD. Safety Manager is responsible for sending and receiving safety messages using PPSCP. Safety Manger is interfacing with TPD to secure or verify safety messages. Safety Manager sends and receives its messages through Network Layer. (See Figure 5-9)

Safety Manager uses Network Layer to broadcast safety messages to other vehicles. Network Layer was modified to clear sender IP from sent message if the payload is safety message. Network Layer passes the message to Medium Access Control (MAC) Layer. MAC layer was modified to clear MAC address if IP address is cleared.

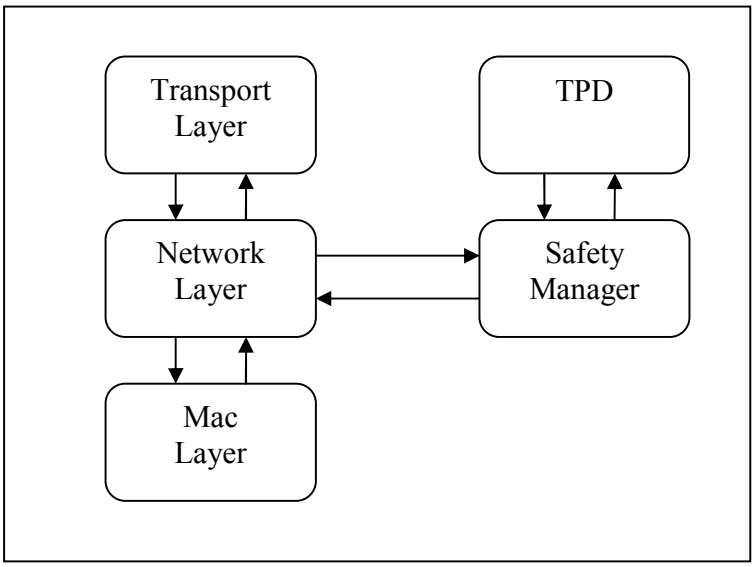

Figure 5-9: Protocol Implementation

IEEE 802.11 is the most adopted wireless protocol. The IEEE 802.11 MAC protocol uses RTS/CTS handshake for unicasting to reduce collisions. Before sending a message, source node transmit request-to-send (RTS). Destination node replies with clear-to-send (CTS). After receiving CTS, source node sends the message to destination. Any other node receiving RTS or CTS does not send or receive until the message is sent.

RTS and CTS contain the MAC address of source and destination which may violate the privacy. Therefore, the proposed protocol uses broadcasting to send safety messages. For broadcasting, the RTS/CTS exchange is not used because there are multiple destinations [82].

5.6. S3P Protocol

S3P protocol is chosen here to compare with PPSCP because it is using shared keys like PPSCP does. In S3P protocol, each vehicle N has a public-private key pair (Pub_N, Prv_N) and a certificate $Cert_N$ which contains information about vehicle identity. Besides, each vehicle stores CA's public key Pub_{CA} in its TPD. Active shared key pair (Pub_{Ai}, Prv_{Ai}) of the anonymity key set A is used by all vehicles in the same time with its corresponding

certificate $Cert_{Ai}$. Each vehicle secures its safety messages before sending it as described in Algorithm 5-1.

Algorithm 5-1: Securing Safety Message in S3P

Input: safety message m

Output: secure safety message SM

1: Get current timestamp t

2: $m' = m \parallel t$

3: $SIG_N = Sign(m', Prv_N)$

4: $EP = Enc_{pub}(SIG_N, Pub_{CA})$

5: $M = m' \parallel EP$

6: $SIG_A = Sign(M, Prv_{Ai})$

7: $SM = M \parallel SIG_A$

8: return SM

As illustrated in Algorithm 5-1. The safety message m is concatenated to a timestamp t to generate m' value. Then, m' is signed with Prv_N to produce SIG_N which contains signature and certificate $Cert_N$. SIG_N is encrypted with CA's public key Pub_{CA} to produce Encrypted Packet (EP). Then, m' is concatenated to EP to produce M value. M value is signed with Pub_{Ai} to produce SIG_A which contains signature and certificate $Cert_{Ai}$. M and SIG_A are broadcasted to other vehicles.

When a vehicle receives a message, it verifies it with Prv_{Ai} to ensure its validity. It is obvious that EP packet can be decrypted by CA only because no entity has CA's private key Prv_N which is needed for decryption. EP contains vehicle N's certificate $Cert_N$ which indicates to vehicle identity. Table 5-5 describes the differences between S3P and PPSCP protocols in securing safety messages.

Table 5-5: Securing safety messages in PPSCP vs. S3P.

Value	PPSCP	S3P
Shared key	- Uses symmetric keys K_i as shared keys.	- Uses public-private key pairs (Pub_{Ai}, Prv_{Ai}) with the corresponding certificate $Cert_N$ as shared keys.
Hiding vehicle identity	- Hides vehicle identity VID_N by decrypting it with Pub_{CA} to produce *EVID* value. - VID_N is concatenated to random number r_1 to ensure different *EVID* every time.	- Hides vehicle identity which is contained in SIG_N by decrypting it with Pub_{CA} to produce *EP* value. - SIG_N is different for every message because it depends on *m* and *t*, so *EP* differs for every message
Revocation	- Encrypts timestamp *t* with revocation key KR_N, then with shared key K_i to produce *ET* value. - Sends KR_N to all vehicles, if vehicle *N* is revoked. - Does not need to update shared keys.	- Sends a message to revoked vehicle to stop its TPD. - Other vehicles switch to emergency keys until they update their shared keys.
Authentication	- Uses *MAC* code with K_i to authenticate messages. - Uses *MAC* code with K_i to verify messages	- Uses digital signature SIG_A with Pub_{Ai} to authenticate messages. - Uses digital signature SIG_A with Prv_{Ai} to verify messages.

S3P is modified here to let comparison fair. We omit adding certificates $Cert_{Ai}$ to SIG_A value produced by signing with shared keys. Moreover, we remove vehicle *N*'s certificate $Cert_N$ from SIG_N and concatenate SIG_N to vehicle identity VID_N to identify the vehicle.

To generate *EP*, a random session key K_S is selected. SIG_N and VID_N are concatenated and padded to be a multiple of 16 bytes (plaintext block size for AES). The result is encrypted by K_S. Finally, K_S is encrypted with Pub_{CA}. This procedure is described as follows:

$X = SIG_N \,||\, VID_N \,||\, Padding$

$Y = Enc_{sym}(X, K_S)$

$EP = Y \,||\, Enc_{pub}(K_S, Pub_{CA})$

RSA signature is chosen for simulation. Signature length is 1024 bit or 2048 bit for RSA-1024 or RSA-2048 respectively. For RSA-1024 bit, *EP* size is calculated as follows:

SIG_N size = 128 bytes

VID_N size = 8 bytes

Padding size = 8 bytes

$Enc_{pub}(K_S, Pub_{CA})$ size = 128 bytes

EP size = 128 + 8 + 8 + 128 = 272 bytes

5.7. Simulation

Our simulation considers vehicles moving in a region of size 938.5m x 747.5m. The selected mobility model in simulation is STreet RAndom Waypoint (STRAW) mobility model. In STRAW, vehicles move on segments representing streets in the selected roadmap. The map used here is a part of Suffolk country map which is a country of Massachusetts State in United States. The simulation map is depicted in Figure 5-10.

Each vehicle will turn at any intersection with probability of (0.3). The standard deviation of 4 is used for each vehicle to select a random speed above or below the speed limit for the road. Speed limit varies according to road type.

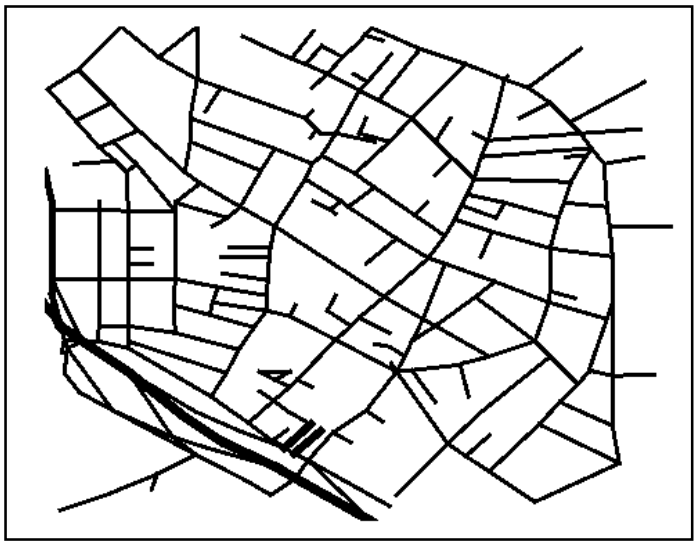

Figure 5-10: Simulation roadmap

For communications modeling, MAC protocol IEEE 802.11 is used with transmission band of 2.4 GHz and bandwidth of 11Mbps. Selected radio propagation model is Free Space propagation model.

Simulation is executed for different number of nodes: 25, 50, 75, 100, 125 and 150 vehicle node with payload of 500 bytes. Moreover, simulation is executed for different payload sizes: 250, 500, 750, 1000, 1250 and 1500 bytes with 50 nodes. Simulation time is 900 seconds for two protocols: PPSCP and S3P.

For PPSCP protocol, safety message size is the sum of payload size, timestamp (8 bytes), *EVID* size (128 bytes), *ET* size (16 bytes) and HMAC size (20 bytes). For S3P, safety message size is the sum of payload size, time stamp size (8 bytes), *EP* size (272 bytes) and SIG_A size (128 bytes).

Table 5-6 shows the number of operations needed by each protocol to secure a safety message before sending. In PPSCP, *ET* value needs 2 AES operations which can be calculated in parallel with *EVID* value which needs one AES and one RSA encryption operations. Therefore, one AES operation are considered in time needed by PPSCP.

Table 5-6: Operations needed for securing a safety massage

Algorithm	PPSCP	S3P
AES	3	9
SHA-1	0	2
HMAC	1	0
RSA Encryption	1	1
RSA Signature	0	2

5.8. Results

Performance metrics used here to compare between the two protocols are Average Message Delay, System Throughput, Message Delivery Rate and Aggregate Transmission Rate.

- **Average Message Delay**: is the average difference between transmission time of a message and the receiving time of it for all safety messages. Delay includes the time needed to secure the message by TPD at transmitter and the time needed to verify it by TPD at receiver.
- **System throughput** is the sum of all bits that are successfully received by all nodes in the network per second [36]. It is sometimes called aggregate throughput. It differs from regular throughput which is average rate of successful data delivery between two points. System throughput is measured in bits per second (bps).
- **Message Delivery Rate**: is the sum of successful received messages by all nodes in the network per second. It is measured in messages per second.
- **Aggregate Transmission Rate**: is the sum of all bits that are transmitted from all nodes in the network per second. It is measured in bits per second (bps).

5.8.1. Average Message Delay

Figures 5-11 and 5-12 show that S3P has a larger Average Message Delay than that of PPSCP because S3P protocol consumes more time in securing safety messages. S3P protocol uses the signature algorithm two times before sending and one time after receiving the message. PPSCP protocol does not use digital signatures to authenticate safety messages, but it uses *MAC* code instead. Figures 5-11 shows that the delay increases with the increase of node numbers. That is because the number of sent messages increases which leads to more message collisions. Figure 5-12 illustrates that the delay increases with the increase of message size. That is because more collisions occurs and channel capacity is consumed.

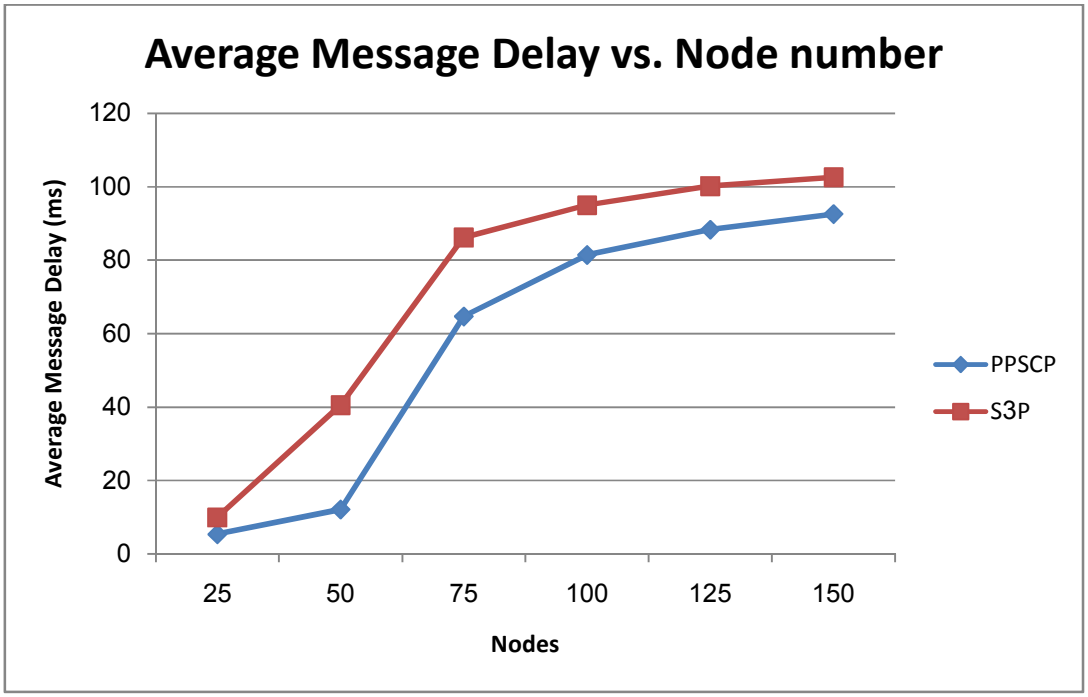

Figure 5-11: Average Message Delay vs. Node number

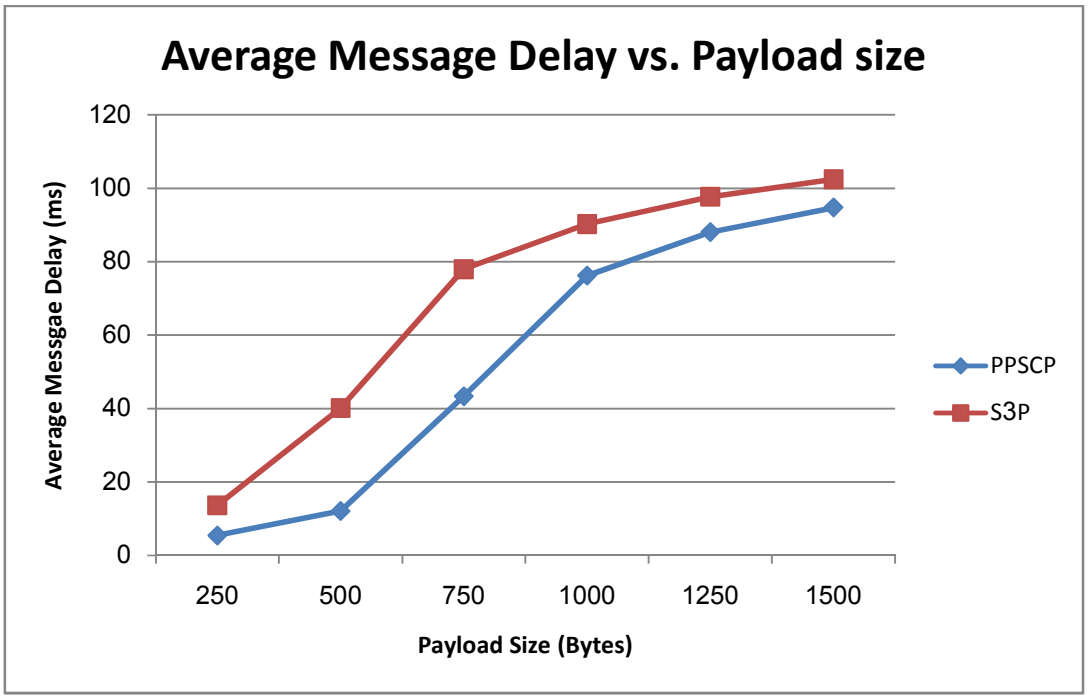

Figure 5-12: Average Message Delay vs. Payload size

5.8.2. System Throughput

Figures 5-13 and 5-14 show the System Throughput of S3P and PPSCP vs. node numbers and payload sizes respectively. In Figure 5-13, System Throughput of S3P is slightly larger than PPSCP because of larger message size produced by S3P. In Figure 5-14, S3P System Throughput is larger than PPSCP for payload sizes below 600 bytes. For payload sizes above 1000 bytes, S3P and PPSCP have the same System Throughput value. System Throughput measures the number of received bits per second but that does not indicate to number of messages.

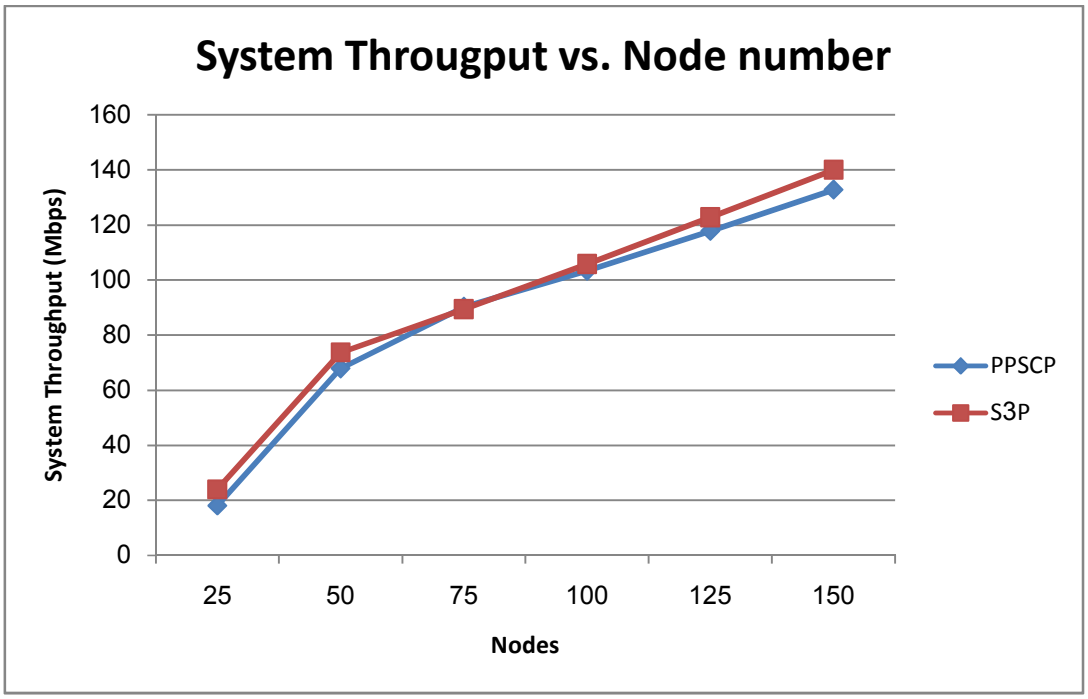

Figure 5-13: System Throughput vs. Node number

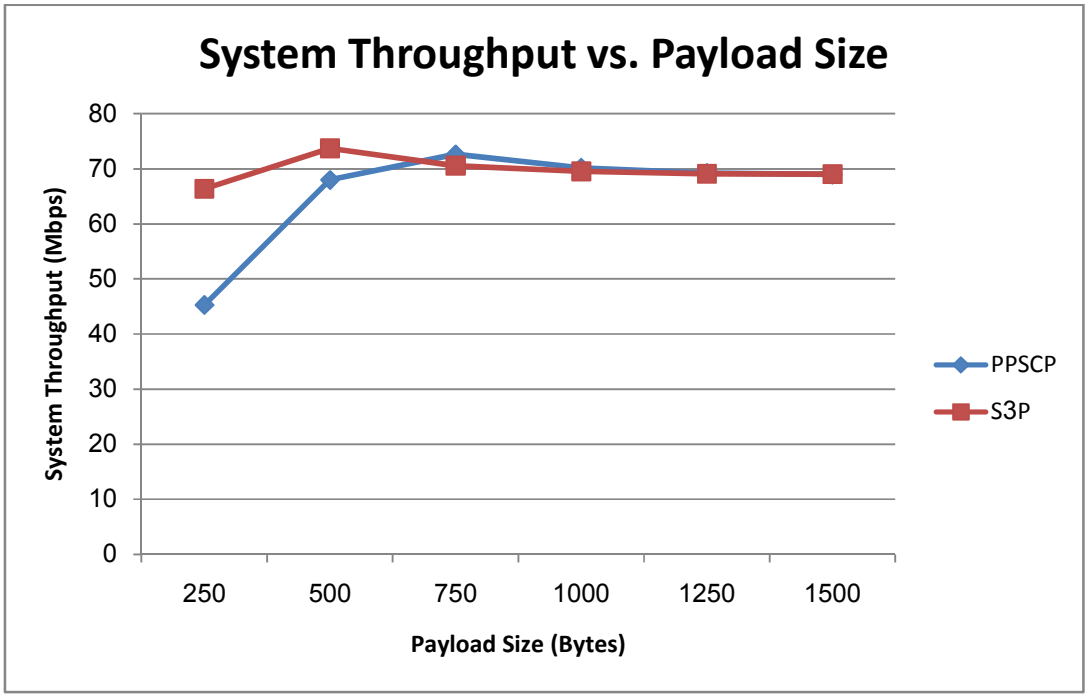

Figure 5-14: System Throughput vs. Payload size

5.8.3. Message Delivery Rate

Figure 5-15 shows the Message Delivery Rate vs. node numbers with constant payload size of 500 bytes. Message Delivery Rate increases with the increase of node numbers because number of senders increases. Figure 5-16 shows the Message Delivery Rate vs. different payload sizes with 50 nodes in the network. Message Delivery Rate decreases with the increase of payload size because larger messages causes more collisions and consumes bandwidth capacity.

The number of successful received messages by PPSCP is larger than that of S3P. This is because larger size of S3P messages leads to more drop in packets caused by collisions.

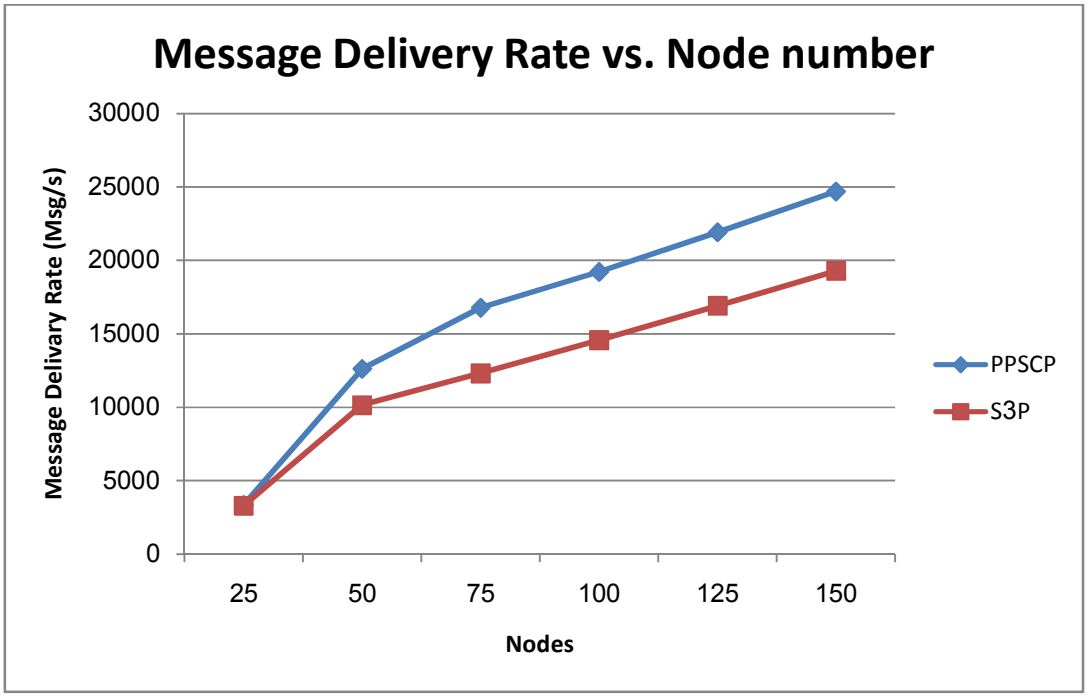

Figure 5-15: Message Delivery Rate vs. Node number

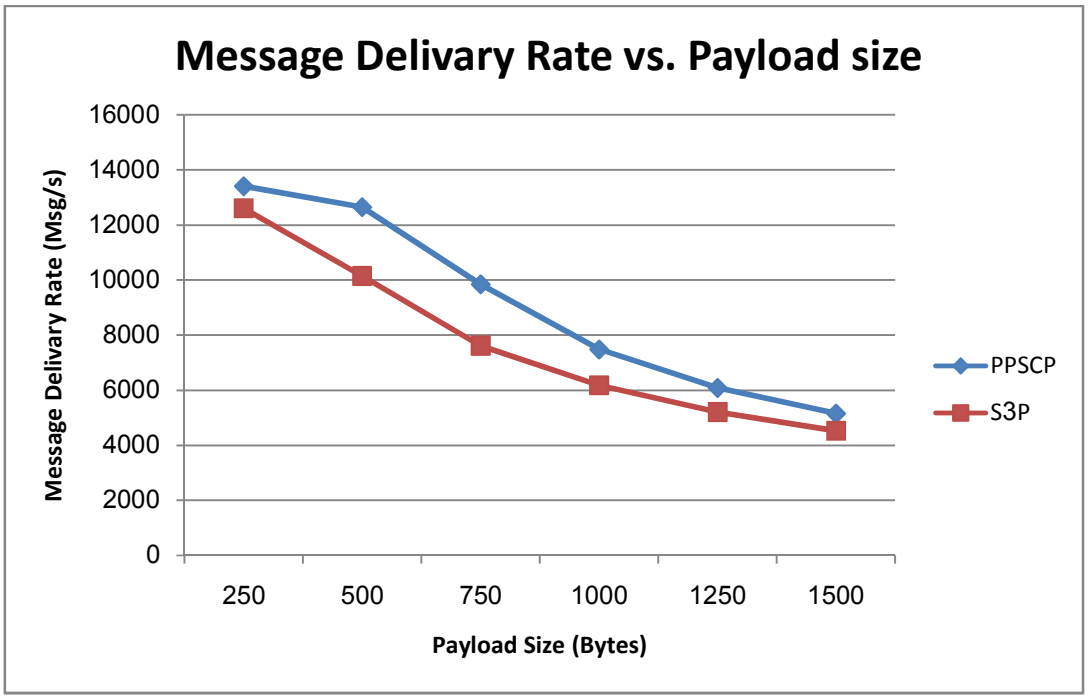

Figure 5-16: Message Delivery Rate vs. Payload size

5.8.4 Aggregate Transmission Rate

Figures 5-17 and 5-18 show the Aggregate Transmission Rate which represents the sum of all bits sent by all nodes per second. Aggregate Transmission Rate increases with the increase of node numbers and with the increase of payload sizes.

Aggregate Transmission Rate of S3P is larger than that of PPSCP because of larger message size of S3P. PPSCP is better because large Aggregate Transmission Rate consumes the bandwidth capacity.

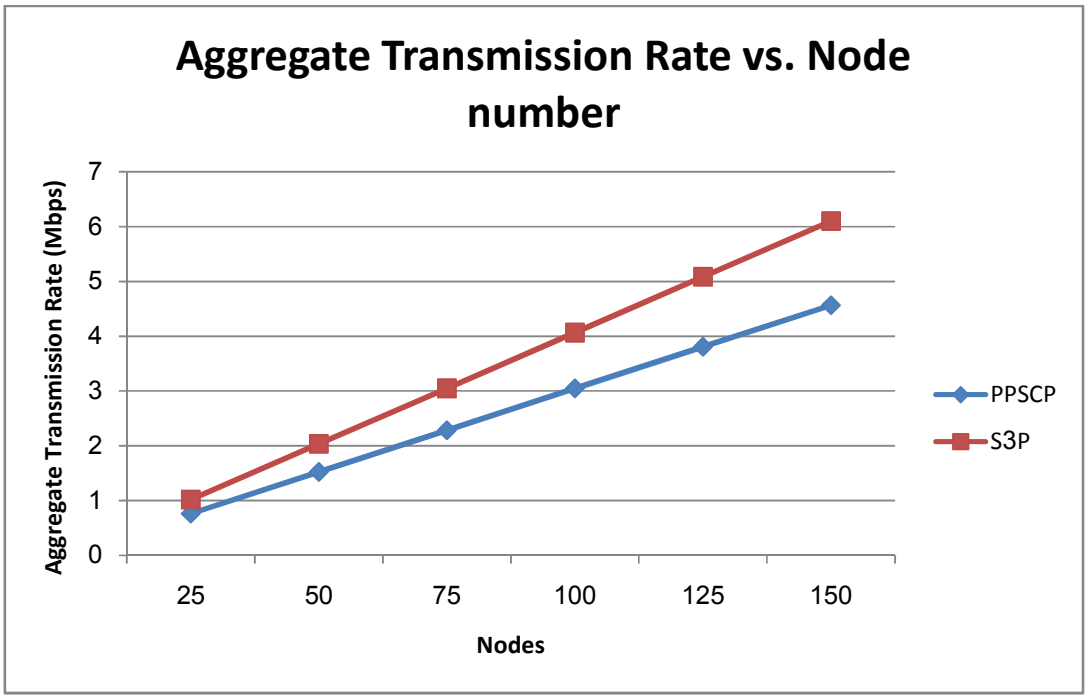

Figure 5-17: Aggregate Transmission Rate vs. Node number

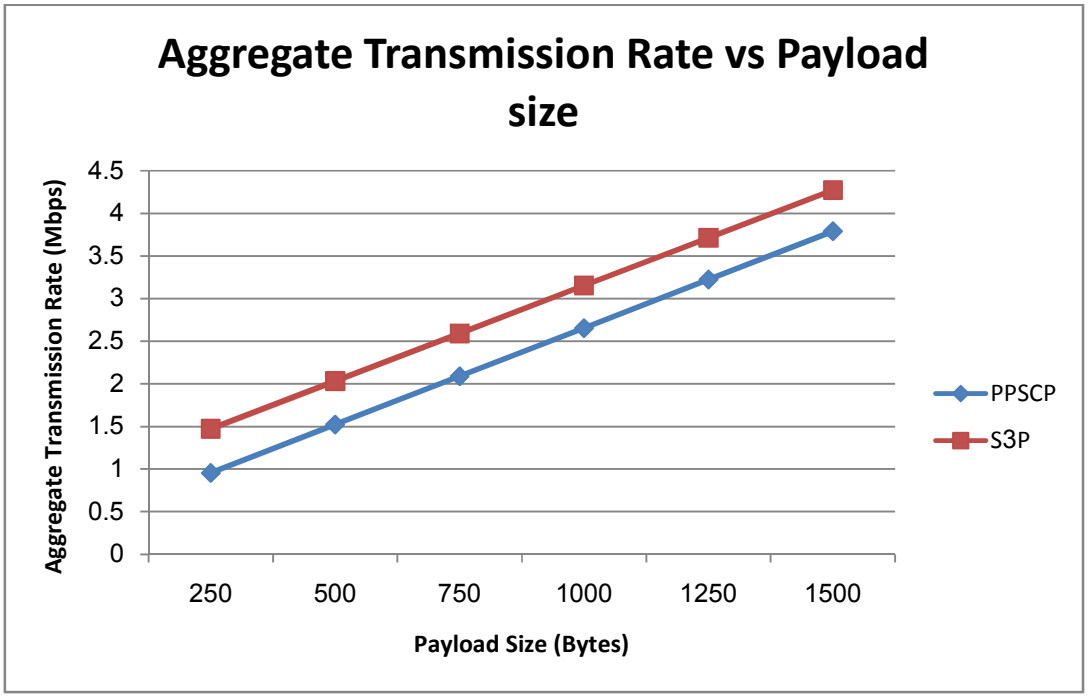

Figure 5-18: Aggregate Transmission Rate vs. Payload size

5.8.5. Results Conclusion

The proposed protocol PPSCP achieved better results than S3P. PPSCP consumes less channel bandwidth and results in larger Message Delivery Rate than S3P. Average Message Delay of PPSCP is less than that of S3P.

PPSCP needs less time to secure safety messages. PPSCP message size is smaller than that of S3P which consumes bandwidth capacity. PPSCP can be optimized by pre-initialization of *EVID* value. That reduces the time needed by every message before sending. No time is needed for using a public-key encryption. However, that needs more storage.

Bandwidth used in simulation is constrained to 11 Mbps. When a high number of vehicles exist in a small area, the bandwidth is consumed, and more collisions in data packet happen. If two or more vehicles transmit safety messages simultaneously, a collision occurs, and they must retransmit again. It is deduced from Figures 5-11 to 5-16 that most of bandwidth is consumed after node numbers go above 75 node or payload sizes exceed 750 byte. Collisions decrease the network efficiency. The solution is to increase the bandwidth capacity to higher value like 54 Mbps used in 802.11g specifications.

Chapter Six: Conclusion and Future Work

6.1. Conclusion

Vehicular Ad-hoc Network VANET is a promising technology which aims to increase safety and efficiency on roads. VANET consists of smart vehicles and road side units RSUs. The main application of VANET is the communication between vehicles about road status and warnings. Each vehicle broadcast bacons or safety messages to other vehicles.

VANET Communications need to consider security and preserving privacy. Many researchers work on the privacy issue in VANET. They propose different protocols to protect privacy using anonymous authentication techniques which are explored and categorized in this thesis. Every technique has some advantages and shortages.

In this thesis, a novel privacy preserving protocol for VANETs is proposed. The proposed protocol PPSCP stands for Privacy Preserving Secure Communication Protocol. The proposed protocol needs that each vehicle has a tamper-proof device (TPD). Each vehicle N has an identity VID_N and a public-private key pair which are pre-installed on TPD. Before sending a safety message, it is secured by cryptographic operations executed inside vehicle's TPD.

The proposed protocol uses shared symmetric keys to achieve anonymous authentication of safety messages. All vehicles use the same shared key at the same time to authenticate messages by a selected MAC algorithm. Used shared key is updated periodically. A trusted authority is responsible for distributing shared keys to legitimate vehicles. Shared keys are distributed in sets to reduce communication overhead.

To prevent replay attack, each vehicle appends a timestamp t to the safety message. When a vehicle receives the same message with same timestamp, it discards the message.

For liability, the protocol supports traceability by a trusted authority. Trusted authority here is any authorized authority like Certification Authority (CA). The sender vehicle encrypts its identity VID_N with CA's public key to produce Encrypted Vehicle's ID ($EVID$) value which is included in the message. Only CA can reveal vehicle's identity by

decrypting *EVID* value with CA's private key. This is need in some cases like accidents reconstruction and crimes.

When a misbehaved vehicle is detected, it must be revoked. Revocation is very important and enables other vehicles to discard messages that are received from a revoked vehicle. The proposed protocol suggests a novel revocation scheme. Each vehicle obtains its own revocation key KR_N from the trusted authority. The vehicle adds the Encrypted Timestamp (*ET*) value to safety message. *ET* is generated by encrypting the timestamp *t* with the revocation key KR_N. if any vehicle is revoked, the trusted authority adds its revocation key KR_N to the revocation list *L*. The trusted authority broadcasts *L* to all vehicles. When a vehicle receives a message, it attempts to decrypt *ET* with all keys in *L*. if the decryption succeeds, the message is discarded. The revoked vehicle will not be capable of obtaining next set of shared keys. Therefore, it will not be able to communicate with other vehicles and its revocation key is removed from *L* to keep it small.

The proposed protocol security and efficiency are analyzed. The protocol is resistant against attacks, and it fulfills security requirements. The protocol is efficient, and it uses one public key encryption to produce *EVID* value before sending the safety message. Furthermore, the protocol can be optimized by generating many *EVID* values before the vehicle starts moving on roads to use them later.

A simulation of the proposed protocol is implemented to test its performance against S3P protocol [2]. The simulation is executed on a part of a real roadmap with variant numbers of nodes and variant sizes of messages. The simulation results show that the proposed protocol PPSCP performs better results than S3P protocol. Average Message Delay of PPSCP is less than that of S3P. Moreover, PPSCP achieves higher Message Delivery Rate.

6.2. Future Work

Proposed protocol ensures non-repudiation of sender requirement for safety messages. However, non-repudiation of receipt is not implemented. When a vehicle

receives a safety message, it must not deny the receipt of that message. As a future work, we want to investigate the best methods to achieve this requirement.

In the future, we would like to evaluate PPSCP on larger roadmaps with more vehicles using varying mobility models. In addition, we would to implement more protocols to compare them with our protocol. Furthermore, we want to study the effect of bandwidth capacity on protocol efficiency and carry out more simulations with higher bandwidth values.

Simulation can only provide an estimated guess of how the approach works in real situation. In order to evaluate the proposed protocol performance and effect on the network, it needs to be implemented and tested in a real case.

References

[1] J. Isaac, S. Zeadally, J. Ca´mara, "Security attacks and solutions for vehicular ad hoc networks," Communications, IET, vol. 4, no. 7, pp. 894–903, 2010.

[2] A. Bayrak, T.Acarman, "S3P: A Secure and Privacy Protecting Protocol for VANET," in Proceedings of 6th International Conference on Wireless and Mobile Communications (ICWMC), pp. 441–447, 2010.

[3] M. Raya, J.P. Hubaux, "The security of vehicular ad hoc networks" in Proceedings of 3rd ACM Workshop on Security of ad hoc and Sensor Networks, Alexandria, 2005, pp. 11–21.

[4] G. Samara, W. Al-Salihy, R.Sures, "Security issues and challenges of Vehicular Ad Hoc Networks (VANET)," in IEEE 4th Int. Conf. on New Trends in Information Science and Service Science (NISS), 2010, pp. 393–398.

[5] S. Kim, H. Oh, "A Simple Privacy Preserving Route Tracing Mechanism for VANET," in IEEE Vehicular Technology Conference (VTC 2010-Spring), 2010, pp. 1–5.

[6] L. Huang, K. Matsuura, H. Yamane, and K. Sezaki, "Enhancing wireless location privacy using silent period," in IEEE Wireless Communications and Networking Conference (WCNC), 2005, pp. 1187–1192.

[7] G. J. Freudiger, M. Raya, M. Feleghhazi, "Mix zones for location privacy in vehicular networks," in Proceedings of First Int. Workshop on Wireless Networking for Intelligent Transportation Systems (WiNITS' 07), Vancouver, Canada, 2007.

[8] L. Buttyán, T. Holczer, I. Vajda, "On the Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs," Security and Privacy in Ad-hoc and Sensor Networks, Lecture Notes in Computer Science, Springer, Berlin / Heidelberg, vol. 4572, pp. 129–141, 2007.

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

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

[11] C. Langley, R. Lucas, H. Fu, "Key management in vehicular ad-hoc networks," in IEEE Int. Conf. on Electro/Information Technology (EIT 2008), pp. 223–226, 2008.

[12] G. Guette, O. Heen, "A TPM-based architecture for improved security and anonymity in vehicular ad hoc networks," in IEEE Vehicular Networking Conference (VNC), pp. 1–7, 2009.

[13] H. Hartenstein, K. P. Laberteaux, "A Tutorial Survey on Vehicular Ad Hoc Networks," IEEE Communications Magazine, vol. 46, no. 6, pp. 164–171, 2008.

[14] M. Nowatkowski, J. Wolfgang, C. McManus, H. Owen, "The effects of limited lifetime pseudonyms on certificate revocation list size in VANETS," IEEE SoutheastCon 2010, pp. 380–383, 2010.

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

[16] M. Burmester, E. Magkos, V. Chrissikopoulos, "Strengthening Privacy Protection in VANETs," in IEEE Int. Conf. Networking and Communications, 2008 (WIMOB '08), pp. 508–513, 2008.

[17] L. Butty´an, T. Holczer, I. Vajda, "On the effectiveness of changing pseudonyms to provide location privacy in vanets," in ESAS, 2007, pp. 129–141, 2007.

[18] K. Sampigethaya, L. Huang, K. Matsuura, R. Poovendran, K. Sezaki, "Caravan: Providing location privacy for VANET," in Escar 2005, 3rd Embedded Security in Cars Workshop, 2005.

[19] G. Calandriello, P. Papadimitratos, J.P. Hubaux, A. Lioy, "Efficient and Robust Pseudonymous Authentication in VANET," in Proceedings of 4th ACM int. Workshop on Vehicular Ad Hoc Networks, pp. 19–28, 2007.

[20] C. Zhang, X. Lin, R. Lu, P.-H. Ho, "RAISE: an efficient rsu-aided message authentication scheme in vehicular communication networks," in Proceedings of IEEE ICC 2008, Beijing, China, May 19–23, 2008.

[21] ITS America, http://www.itsa.org/, accessed Dec. 2011.

[22] Vehicle, Road and Traffic Intelligence Society Organization, VERTIS, Japan, http://www.mlit.go.jp/road/ITS/, accessed Dec. 2011.

[23] Intelligent Transport Systems of Taiwan, http://www.its-taiwan.org.tw/, accessed Dec. 2011.

[24] P. Kamat, A. Baliga, W. Trappe, "An identity-based security framework for VANETs," in Proceedings of International Conference on Mobile Computing and Networking, Los Angeles, California, USA, pp.94–95, 2006.

[25] P. Kamat, A. Baliga, W. Trappe, "Secure, Pseudonymous, and Auditable Communication in Vehicular Ad Hoc Networks," Journal of Security and Communication Networks, vol. 1, no. 3, pp. 233–244, 2008.

[26] C. Lai, H. Chang, Chei Chung Lu, "A secure anonymous key mechanism for privacy protection in VANET," 9th International Conference on Intelligent Transport Systems Telecommunications, (ITST), pp. 635–640, 2009.

[27] J. Sun, C. Zhang, Y. Zhang, Y. Fang, "An Identity-Based Security System for User Privacy in Vehicular Ad Hoc Networks," Parallel and Distributed Systems, IEEE Transactions, vol. 21 , no. 9, pp. 1227–1239, 2010.

[28] A. Shamir, "Identity-based cryptosystems and signature schemes," Advances in Cryptology -Crypto'84, LNCS, vol. 196, Springer-Verlag, pp. 47–53, 1984.

[29] D. Boneh, M. Franklin, "Identity-Based Encryption from the Weil Pairings," Advances in Cryptology -Asiacrypt, Springer-Verlag, pp. 514–532, 2001.

[30] J. P. Hubaux, S. Capkun, J. Luo, "The Security and Privacy of Smart Vehicles," IEEE Security and Privacy Magazine, vol. 2, no. 3, pp. 49–55, 2004.

[31] F. Dotzer, "Privacy Issues in Vehicular Ad Hoc Networks," Workshop on Privacy Enhancing Technologies, May 2005.

[32] D. Huang, S. Misra, M. Verma, Guoliang Xue, "PACP: An Efficient Pseudonymous Authentication-Based Conditional Privacy Protocol for VANETs," Intelligent Transportation Systems, IEEE Transactions on , vol .12, no. 3, pp. 736–746, 2011.

[33] E. Schoch, F. Kargl, T. Leinmuller, S. Schlott, P. Papadimitratos, "Impact of Pseudonym Changes on Geographic Routing in VANETs," in Proceedings of the European Workshop on Security and Privacy in Ad hoc and Sensor Networks (ESAS), Hamburg, Germany, October 2006.

[34] W. Franz, R. Eberhardt, T. Luckenbach, "Fleetnet - internet on the road," in Proceedings of the 8th World Congress on Intelligent Transportation Systems, 2001.

[35] T. Kosch, "Local danger warning based on vehicle ad-hoc networks: Prototype and simulation," in Proceedings of 1st International Workshop on Intelligent Transportation (WIT 2004), 2004.

[36] Sok-Sien Choy; H. Lee, "Efficient broadcast using link-state routing information in packet radio networks,", in Proceedings of IEEE International Conference on Networks (ICON 2000), pp.152–159, 2000.

[37] S. Kim, H. Oh, "A Simple Privacy Preserving Route Tracing Mechanism for VANET," in IEEE Vehicular Technology Conference (VTC 2010-Spring), pp. 1–5, 2010

[38] M. S. Kakkasageri, S. S .Manvi, B. M. Goudar, "An agent based framework to find stable routes in Mobile Ad hoc Networks (MANETs)," in IEEE Region 10 Conference, pp. 1–6, 2008.

[39] E. M. Royer, Chai-Keong Toh, "A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks," in IEEE Personal Communications Magazine, pp. 46–55, April 1999.

[40] U. G. Swarnapriyaa, V. Vinodhini, S. Anthoniraj, R. Anand, "Auto configuration in Mobile Ad hoc Networks," in National Conference of Innovations in Emerging Technology (NCOIET 2011), pp. 61–66, 2011.

[41]. X. Cheng, X. Huang, D. Z. Du, "Ad Hoc Wireless Networking," Kluwer Academic Publishers, pp.319–364, 2006.

[42] D. B. Johnson, D. A. Maltz, Yih-Chun Hu, "The dynamic Source Routing Protocol for Mobile Ad Hoc Networks," Internet Draft, July 2004, Available at: http://tools.ietf.org/html/draft-ietf-manet-dsr-10, accessed Dec. 2011.

[43] C. Perkins, "Ad hoc On-Demand Distance Vector (AODV) routing," RFC 3561, July 2003, Available at: http://www.ietf.org/rfc/rfc3561.txt, accessed Dec. 2011.

[44] Joo-han Hong, "Efficient on-demand routing for mobile ad hoc wireless access networks," in IEEE Journal on Selected Areas in Communications, pp. 11–35, vol. 22, 2004.

[45] C. E. Perkins, P. Bhagwat "Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers," in Proceedings of Conference on Communications Architectures, Protocols and Applications (SIGCOMM 1994), pp 234–244, Aug. 1994.

[46] Z. J. Haas, M. R. Pearlman, P. Samar, "The Zone Routing Protocol (ZRP) for Ad Hoc Networks," Internet Draft, July 2002,
Available at: http://tools.ietf.org/html/draft-ietf-manet-zone-zrp-04,
accessed Dec. 2011.

[48] A. Festag, et al, "NoW - Network on Wheels: Project Objectives, Technology and Achievements,"in proceedings of 6th International Workshop on Intelligent Transportation (WIT), Hamburg, Germany, March, 2008.

[49] PReVENT project, http://www.prevent-ip.org/en/home.htm, accessed Dec. 2011.

[50] National Highway Traffic Safety Administration, "Vehicle Safety Communications – Applications (VSC-A)," DOT HS 811 492A, September, 2011.

[51] Road Safety, European Commission,
http://ec.europa.eu/transport/road_safety/index_en.htm, accessed Dec. 2011.

[52] Fatality Analysis Reporting System (FARS) Encyclopedia, NHTSA, http://www-fars.nhtsa.dot.gov/Main/index.aspx, accessed Dec. 2011.

[53] S. Behera, B. Mishra, P. Nayak, D. Jena, "A secure and efficient message authentication protocol for vehicular Ad hoc Networks with privacy preservation (MAPWPP)," in IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application (IMSAA), pp.1–6, 2011

[54] Y. Sun, R. Lu, X. Lin, X. Shen, J. Su, "A Secure and Efficient Revocation Scheme for Anonymous Vehicular Communications," in IEEE International Conference on Communications (ICC), pp.1–6, 2010.

[55] S. Zhang; Jun Tao, Yijia Yuan, "Anonymous authentication-oriented vehicular privacy protection technology research in VANET," in International Conference on Electrical and Control Engineering (ICECE), pp.4365–4368, 2011.

[56] R. Lu, X. Lin, H. Zhu, P. H. Ho, X. Shen, "ECPP: efficient conditional privacy preservation protocol for secure vehicular communications," in Proceedings of INFOCOM 2008, pp. 1229–1237, 2008.

[57] A. Wasef, Yixin Jiang, Xuemin Shen, "DCS: An Efficient Distributed-Certificate-Service Scheme for Vehicular Networks," in IEEE Transactions on Vehicular Technology, vol. 59, no. 2, pp. 533–549, 2010.

[58] C. Jung, C. Sur, Y. Park, K. Rhee, "A robust conditional privacy preserving authentication protocol in VANET," in Proceedings of MobiSec 2009, pp. 35–45, Turin, Italy, 2009.

[59] K. Sampigethaya, Mingyan Li, L. Huang, R. Poovendran, "AMOEBA: Robust Location Privacy Scheme for VANET," in IEEE Journal on Selected Areas in Communications, vol. 25, no. 8, pp.1569–1589, 2007.

[60] Jian Wang; Nan Jiang, "A simple and efficient security scheme for vehicular ad hoc networks," in IEEE International Conference on Network Infrastructure and Digital Content, IC-NIDC 2009, pp. 591–95, 2009.

[61] D. Boneh, X. Boyen, H. Shacham, "Short group signatures," in Advances in Cryptology - CRYPTO, vol. 3152, Lecture Notes in Computer Science, Berlin, Germany, Springer-Verlag, pp. 41–55, 2004.

[62] D. Chaum, E. Van Heyst, "Group signatures," in Advances in Cryptology - Eurocrypt 1991, vol. 576, Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, pp. 257–265, 1991.

[63] Lei Zhang, Qian Hong Wu, Agusti Solanas, "A Scalable Robust Authentication Protocol for Secure Vehicular Communications," in IEEE Transaction on Vehicular Technology. vol. 59, no. 4, pp. 1606–1617, 2010.

[64] R. L. Rivest, A. Shamir, Y. Tauman, "How to Leak a Secret," In AsiaCrypt 2001, vol. 2248 of LNCS, pp. 552–565, 2001.

[65] D. Y.W. Liu, J. K. Liu, Y.Mu,W. Susilo,D.S.Wong. Revocable Ring Signature, Journal of Computer Science and Technology, vol. 22, no. 6, pp. 785–794, 2007.

[66] Hu Xiong, Zhi Guan, Jianbin Hu, Zhong Chen, "Anonymous Authentication Protocols for Vehicular Ad Hoc Networks: An Overview," Applied Cryptography and Network Security, pp. 53–72, 2012.

[67] J. Yim, I. Choi, K. Kim, "An efficient anonymous authentication protocol in vehicular ad-hoc networks," WISA 2009, pp. 24–26, 2009.

[68] Rimon Barr, "JiST User Guide," 19 March 2004, Available at: http://jist.ece.cornell.edu/docs/040319-jist-user.pdf, accessed Jan. 2012.

[69] Rimon Barr, "SWANS User Guide," 19 March 2004, Available at: http://jist.ece.cornell.edu/docs/040319-swans-user.pdf, accessed Jan. 2012.

[70] McCanne, S., and S. Floyd, "The Network Simulator – ns2," 1997, Available at: http://www.isi.edu/nsnam/ns/, accessed Jan. 2012.

[71] X. Zeng, R. L. Bagrodia, M. Gerla, "GloMoSim: a library for parallel simulation of large-scale wireless networks," In *PADS*, 1998.

[72] D. R. Choffnes, F. E. Bustamante John S. Otto. "C3 - Car-to-Car Cooperation for Vehicular Ad-hoc Networks," Available at: http://aqualab.cs.northwestern.edu/projects/111, accessed Feb. 2012.

[73] D. R. Choffnes, "SWANS++ User's Guide," 9 April 2007, Available at: http://aqualab.cs.northwestern.edu/component/attachments/download/171, accessed Feb. 2012.

[74] David R. Choffnes, Fabián E. Bustamante, "Modeling Vehicular Traffic and Mobility for Vehicular Wireless Networks," Tech. Report NWU-CS-05-03, Department of Computer Science, Northwestern University, 2005.

[75] Crypto++ 5.6.0 Benchmarks, http://www.cryptopp.com/benchmarks.html, accessed Mar. 2012.

[76] NIST, "FIPS PUB 197: Advanced Encryption Standard (AES)," 26 November 2001, Available at: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, accessed Mar. 2012.

[77] Kleinjung, et al., "Factorization of a 768-bit RSA modulus," International Association for Cryptologic Research, 2010.

[78] R. L. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, April 1992, Available at: http://tools.ietf.org/html/rfc1321, accessed Mar. 2012.

[79] NIST, "FIPS 180-2: Secure Hash Standard (SHS)," 1 August 2002, Available at: http://csrc.nist.gov/publications/fips/fips180-2/fips180-2withchangenotice.pdf, accessed Mar. 2012.

[80] NIST, "FIPS 198: The Keyed-Hash Message Authentication Code (HMAC)," 6 March 2006, Available at: http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf, accessed Mar. 2012.

[81] NIST, "NIST Special Publication 800-38B," May 2005, Available at: http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf, accessed Mar. 2012.

[82] Jiawei Xie; A. Das, S. Nandi, A. K. Gupta, "Improving the reliability of IEEE 802.11 broadcast scheme for multicasting in mobile ad hoc networks," Wireless Communications and Networking Conference, 2005 IEEE, vol. 1, pp. 126–131, 2005.

[83] NIST, "FIPS 186: Digital Signature Standard (DSS)," 19 May 1994, Available at: http://www.itl.nist.gov/fipspubs/fip186.htm, accessed Mar. 2012.