The Islamic University–Gaza
Research and Postgraduate Affairs
Faculty of Engineering
Master of Computer Engineering

الجامعـــة الإســلاميـة– غزة
شئون البحث العلمي والدراسات العليا
كليــة الهـندسـة
ماجستير هنـدسـة الحــاسوب

Design a Cloud Security model in VANET Communication

تصميم نموذج أمان لشبكة المركبات من خلال السحابة الإلكترونية

Alaaeddin B. AlQazzaz

Supervised by

Prof. Hatem M. Hamad

Prof. of Computer Engineering

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Computer Engineering

August /2016

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Design a Cloud Security Model in VANET Communication

تصميم نموذج أمان لشبكة المركبات من خلال السحابة الإلكترونية

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل الاخرين لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

Declaration

I understand the nature of plagiarism, and I am aware of the University's policy on this.

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted by others elsewhere for any other degree or qualification.

Student's name:	علاء الدين باسم القزاز	اسم الطالب:
Signature:		التوقيع:
Date:	13/8/2016	التاريخ:

Abstract

During the last view years, Intelligent Transportation System (ITS) has progressed at a rapid rate, which aimed to improve transportation activities in terms of safety and efficiency. The communication between cars is often referred to Vehicular Ad-Hoc Networks (VANET) and it has many advantages such as: reducing cars accidents, minimizing the traffic jam, reducing fuel consumption and emissions and etc. But, the secutiy issues are very important to take into the consideration within the VANET. There is no solid VANET security model framework to be compatible with different manufacturers in different countries in order to satisfy the VANET's security requirements.

In this thesis, we propose a model of VANET that based on the cloud, which is called, Vehicle-to-Cloud (V2Cloud), and design a security model framework that is hosted on the cloud to manage the security services, and provide a secure VANET communication between the different entities e.g. vehicles and authorities. This security model framework is called VANET Security as a Service (VSaaS).

We investigate the performance metrics measurements through the NS2, SUMO and Trans simulations, to evaluate the security overhead of the secure Vehicle Information Messages (VIMs), which are sent by the vehicles to the cloud as a coarse-grained information. Moreover, we analyze our proposed model framework (VSaaS) against the VANET's security requirements.

According to the simulation results and the security analysis, VSaaS model framework is secure, efficient, modular, managed by cloud, and fulfills the VANET's security requirements.

الملخص

خلال السنوات القليلة الماضية، تقدمت أنظمة المواصلات الذكية بشكل سريع، حيث أن هذه الأنظمة تهدف إلى تحسين المواصلات من حيث السلامة والكفاءة .إن التواصل بين السيارات كثيرا ما يشار إليه بشبكة المركبات (VANET) ولها العديد من المزايا مثل :الحد من حوادث السيارات، والتقليل من الازدحام المروري، والحد من استهلاك الوقود والانبعاثات وغير ذلك. ولكن يجب الأخذ بعين الإعتبار قضايا الأمان لشبكة المركبات حيث أنه لايوجد نموذج أمني متين يتوافق مع جميع الشركات المصنعة من مختلف البلدان من أجل تلبية متطلبات الأمان لشبكة المركبات.

في هذه الأطروحة، اقترحنا طريقة لعمل شبكة المركبات بالاعتماد على السحابة الإلكترونية وهي مايسمي بـ (V2Cloud) كما أننا قمنا بتصميم نموذج أمني مستضاف على السحابة الإلكترونية بهدف إدارة خدمات الأمان وتوفير بيئة أمنية لاتصالات شبكة المركبات بحيث تصبح مختلف العناصر مثل (المركبات والسلطات) قادرة على الإتصال بشكل آمن. أسمينا هذا النموذج الأمني بـ
(VANET Security as a Service (VSaaS)).

قمنا بدراسة قراءات مقاييس الآداء من خلال برامج المحاكاة (NS2, SUMO and Trans simulations) وذلك لتقييم عبء النموذج الأمني على الرسائل الآمنة التي ترسلها المركبات إلى السحابة الإلكترونية. علاوة على ذلك، قمنا بعمل تحليل لمقترح النموذج الأمني الذي قمنا بتصميمه بهدف مناقشة مدى تحقيقه لمتطلبات الأمان لشبكة المركبات.ووفقا لنتائج برامج المحاكاة والتحليل الأمني، فإن مقترحنا الأمني (VSaaS) آمن، وفعال، وقابل للتطور والتوسع، يتم إدارته من خلال السحابة الإلكترونية، ويحقق متطلبات الأمان اللازمة لشبكة المركبات.

Dedication

To my great father and my great mother

To my dear wife

To my son and my daughters

To my sister and my brothers

And to my beautiful country "Palestine"

Acknowledgment

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

I would like to record my gratitude to Prof. Hatem Hamad 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 Tables

List of Figures

List of Abbreviations

ACL	Access List
ADB	Authority Database
AES	Advanced Encryption Standard
AID	Authority Identification
CA	Certified Authority
CTDB	Cellular Towers Database
EDR	Event Data Recorder
EVDB	Event Viewer Database
GaaS	Gateway as a Service
GPS	Global Positioning System
HC	Hybird Cloud
I2I	Infrastructure to Infrastructure
ITS	Intelligent Transportation System
KDB	Keys Database
MANET	Mobile Ad-Hoc Network
MTU	Maximum Transmission Unit
OBU	On-Board Unit
OSM	Open Street Map
PKI	Public Key Infrastructure
RSA	Rivest, Shamir and Adleman
RSU	Road Side Unit
SP	Security Proccess
SUMO	Simulation of Urban Mobility
TDM	TIMs Dissemination Module
TIaaS	Traffic Information as a Service
TIDB	Traffic Information Messages Database
TIM	Traffic Information Message
TPD	Tamper-Proof Device
TraNS	Traffic and Network Simulation Environment
TRH	Tamper Resistant Hardware
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2C	Vehicle to Cloud
VANET	Vehicular Ad-Hoc Network
VC	Vehicular Cloud
VCC	Vehicular Cloud Computing
VDB	Vehicle Database
VID	Vehicle Identification
VIDB	Vehicle Information Database
VIM	Vehicle Information Message
VPN	Virtual Private Network
VPM	Vehicle Processing Module
VRM	Vehicle Revocation Module
VSaaS	VANET Security as a Service
VuC	Vehicles using Cloud

Chapter 1

Introduction

Chapter 1
Introduction

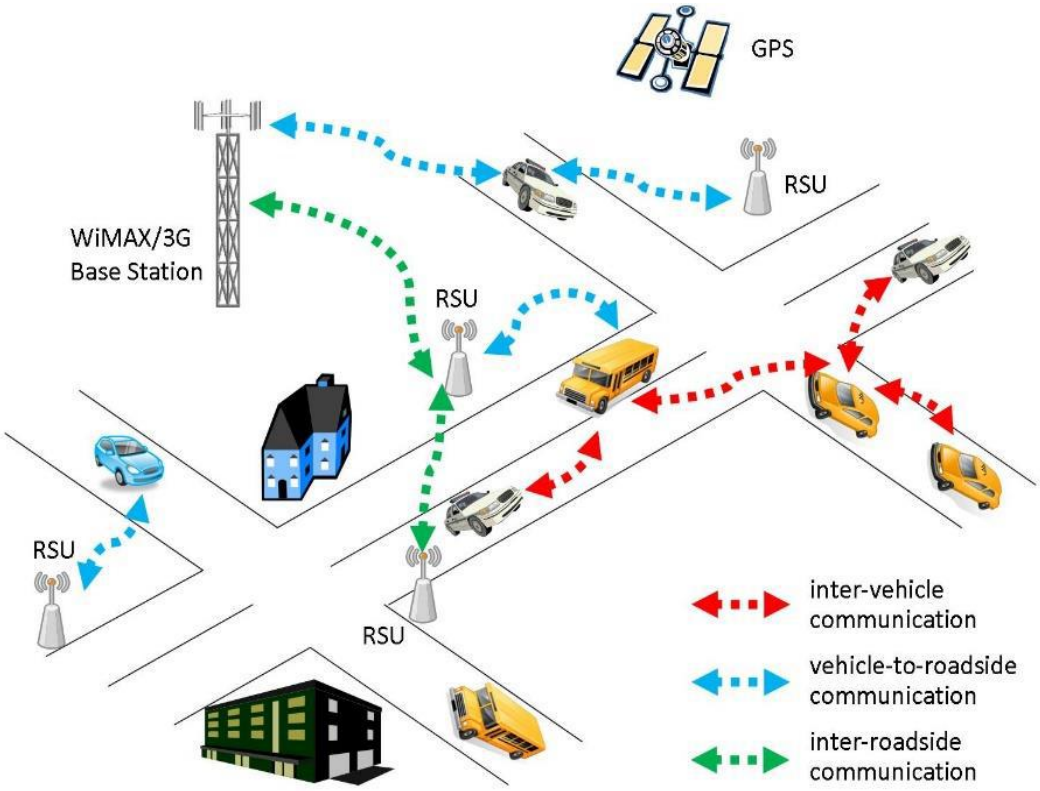
This chapter will present an overview to Vehicular Ad-Hoc Networks (VANET), and will give a short explaination about what are VANET, and the requirpments of VANET. Furthermore, this chapter will introduce the outline of the thesis, the scope and objectives.

1.1 Background

In the last years, one of the important domains in the computer and network science has been studied is the communication between cars (vehicles). Mentioned communication could be achieved with the using Vehicular Ad-Hoc Network (VANET), which is similar to the Mobile Ad-Hoc Networks (MANET), used for transferring information between close vechiles and between vechiles and Road Side Units (RSUs). The main goal for VANETs are providing comfort and safety for vehicles' passengers. To achieve mentioned goal, electronic devices such as a wireless modem, Global Positioning System (GPS) sensor and etc, should be implemented inside vehciles for providing the VANET communication.

In the VANET, the nodes (vehicles) can move very fast, and the considered network is highly dynamic which means that topology of the network is continuously changing with changing the position of the nodes and density. As a result, VANET is the technology for communication between vehicle to each other (which is called Vehicle-to-Vehicle (V2V) communications) and also with Road Side Units (RSUs) (which is called Vehicle-to-Infrastructure (V2I) communications) (See Figure 1.1) (Fiore, Haerri, Filali, and Bonnet, 2007; Khairnar and Pradhan, 2013). Nowadays, this type of VANET is called traditial VANET.

In the traditional VANET, a wide variety of applications can be employed. These applications are classified into two categories (Raya and Hubaux, 2007): safety-related applications and information 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. Safety-related messages are divided into two types (Hartenstein and Laberteaux, 2008): periodic and event-driven messages which are sent when a hazardous situation occurs. In information applications, sometimes called non safety applications, include other kind of applications like payment services, Internet-access, locations services and weather condition.



Figure(1.1): Different Communication types in Traditional VANET

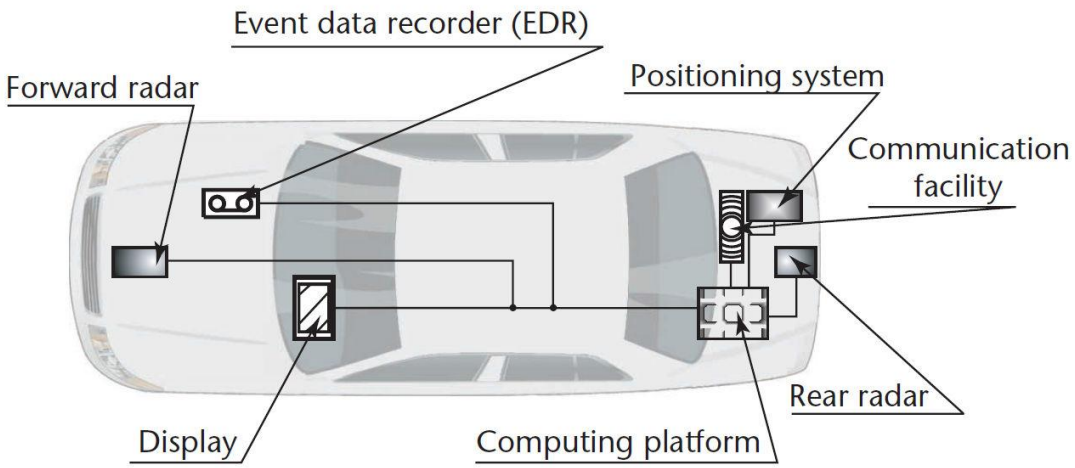
1.1.1 Road Side Units (RSUs)

In the traditional VANET, a Road-Side Unit (RSU) is a physical device located at fixed positions along roads and highways, or at dedicated locations such as gas station, parking places, and restaurants. An RSU is equipped with at least a network device for a wireless communications to participate in the VANET. An RSU can also be equipped with other network devices in order to allow communications with the infrastructure network. The main functions of the RSUs are listed below (Festag et al., 2008).

1. Extending the communication range of an Ad-Hoc network by means of re-distributionof the information to other OBUs and cooperating with other RSUs in forwarding orin distributing safety information.
2. Running safety applications.
3. Providing Internet connectivity to OBUs.
4. Providing gateways to servers and authorities

1.2 Smart Vehicles

Vehicles in VANET are smart vehicles because they are equipped with recording, processing, positioning, and location capabilities (See Figure 1.2). Besides, they can run wireless networking protocols (Hubaux, Capkun, and Luo, 2004).



Figure(1.2): A smart vechile

The smart vehicles are consists of two main units, they are On-Board Unit (OBU) and Tamper Proof Device (TPD).

1.2.1 On-Board Unit (OBU)

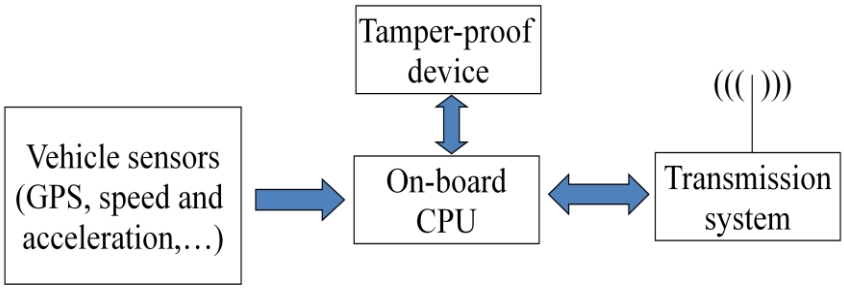
Vehicle's On-Board Unit (OBU) is a central computing platform connected with the wireless communication facilities and other devices like: sensors and data recorders (See Figure 1.2). Some of the most used devices are (Hubaux et al., 2004):

1. **Event Data Recorder (EDR):** to record the vehicle data for crash reconstruction or determination of the misbehaved vehicles.
2. **GPS Receiver:** to get the current position of the vehicle.
3. **Front-end and rear Radars:** for detecting obstacles at front and rear of vehicle. They can be used for parking.

Breifly, OBU functions include wireless radio access, geographical ad hoc routing, network congestion control, reliable message transfer, data security, IP mobility support and others.

1.2.2 Tamper Proof Device (TPD)

Tamper Proof Device (TPD) is the trusted computing base of a vehicle. The purpose of a TPD is to provide a physically protected environment for the storage of private keys (long-term and short-term keys), the execution of cryptographic operations (message signing, encryption and decryption) and key management functions (Papadimitratos et al., 2007). The TPD is physically separated from the OBU. The TPD must have an Application Programming Interface (API) through which it can provide services to the other modules of the security architecture that run on the onboard unit (OBU) (Kargl et al., 2008). For example, 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 1.3).



Figure(1.3): OBU Interfacing with others Devices

This API should support the digital signature, timestamping service (to prevent replay of messages signed by TPD) and the encryption/decryption services, as well as the key and device management services described in (Raya and Hubaux, 2007). The TPD should have its own battery, which can be recharged from the vehicle, and clock (for timestamping service), which can be securely resynchronised. The access to this device should be restricted to authorised people. For example, cryptographic private keys are generated and stored in TPD by vehicular authorities, short term keys can be renewed at the periodic technical checkup of the vehicle, and expired certificates can be changed by the user upon authentication (Plobl and Federrath, 2008). The use of secret information such as private keys requires that TPD be a Tamper Resistant Hardware (TRH).

TRH contains a set of sensors that can detect hardware tampering and erase all the stored keys to prevent them from being compromised. 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 (Papadimitratos et al., 2007).

1.3 Cloud Computing

Cloud Computing has changed the computation and communication mindset by decoupling the computational assets from the physical infrastructure, and thereby enabling virtualization (Armbrust et al., 2010). The main motivation of the Cloud Computing is to "exactly what you need and when you need". Since with the improvment in cloud technology, the Internet becomes high-speed and has low cost than before, it would be better if it is utilized for more than just browsing. Another reason is that the advancements in parallel and distributed computing compel the application developers and industry to utilize the Internet. Cloud computing is very appealing to the business startups with low upfront, virtually and no maintenance cost. This computing paradigm offers new opportunities for developers and infrastructure providers at par. Until very recently, having virtually unlimited resources at very low affordable cost was just a dream, but cloud computing made it reality and there are many players in the market providing cloud services like Amazon, Microsoft, and Google (Hussain, Son, Eun, Kim, and Oh, 2012).

1.3.1 Cloud Services

Needless to say that cloud computing is becoming a well-known buzzword for the last decade. Generally cloud computing refers to both services accessed via, and delivered through, the vast universe of Internet, and the hardware and system software in remote datacenters that provide those services. The beauty of virtualization in cloud computing is attracting large businesses to migrate to cloud environments. The main concern of these corporations is the control over cloud. As far as the motivation for migration is concerned, it is important to realize that most of the issues are essentially old problems in new settings. For instance offshore outsourcing must guarantee certain security primitives such as data integrity, data security, privacy and etc (Hussain et al., 2012).

Cloud computing is also known as ‘utility computing’ which is based on ‘pay-as-you-go’ service (Armbrust et al., 2010). The scenario can be easily compared with our daily life, where we use gas and electricity in our homes as much as we need and at the end of the month we pay for exactly what we have used, neither more nor less. Cloud computing environment offers rich amount of resources. Examples are Amazon S3, Google Drive, and Microsoft SkyDrive. Besides storage, clouds also offer computation resources, such as Amazon Elastic

Compute Cloud (EC2), which can significantly reduce the cost of maintaining resources locally. Besides, online collaboration tools, such as Google Apps or versioning repositories for source code make it easy to develop applications online without purchasing licenses for different softwares. Along with the enterprises, home users can also take advantage of cloud computing, if the services are available at reasonable prices. The use of new sophisticated handheld devices is drastically increasing day by day. But still these devices are lacking far behind the traditional computers in computational power. Nevertheless, notebook computers are now transforming to tablets or a light netbook, which can take advantage of cloud services for intensive computations (Pearson and Benameur, 2010).

The core of the cloud services is comprised of three basic delivery models in the form of Layers (Hussain et al., 2012). The top layer is known as Software as a Service (SaaS). This layer delivers applications to consumers (either individual or enterprise) in a multitenant fashion. Usually the consumers use thin clients to access those services through Internet. The principle benefit to consumer is that, he/she does not have to pay the upfront cost for hardware or software licensing. The by far best example of this service suit is the Google Docs which is equivalent to Microsoft Office. Google provides the aforementioned service to its consumers for free.

Platform as a Service (PaaS) is the second type of service in the layered stack which refers to delivering the development environment as a service to the consumers instead of installing development tools/softwares on host computers. This makes the consumers capable of doing their development remotely by using only the services provided by the service provider. Normally this kind of service works well at enterprise level and the best example is Google App Engine.

At the bottom of the layered stack, cloud computing provides Infrastructure as a Service (IaaS). Instead of application or environment, in this paradigm, physical resources are delivered to consumers as a service. These resources include servers, connections, and related tools necessary to build an application environment from the scratch. Consumers have virtually unlimited resources according to their budget. They can rent processing, storage, networks, and other fundamental computing resources on which the consumer then deploy and run arbitrary cloud application softwares and system softwares. Amazon is providing such services on rent through its elastic computing called EC2.

1.4 Motivation

Recently, the number of vechiles on city roads is increasing and many problems are occurring, such as traffic congestion, the huge number of citizens getting killed in car accidents, fuel consumptions, emissions and etc. For example, according to the National Highway Traffic Safety Administration (NHTSA), there are about 43,000 people who are killed in car accidents each year in the United States (National Highway Traffic Safety Administration Website, 2003) and according to Road Safety in European Commission, there are about 35,000 people who are killed in car accidents each year in the European Union (Road Safety in European Commission Website, 2011).

The VANET security and privacy issues are very important to take place of the VANET's advantages. Hence, the focusing on the secure safety message is an interested because the attacks on these messages may lead to disaster results. In addition, the availability of the Internet access with high speed makes it is possible to provide all security services by the cloud environment to avoid the needed to fully connected V2V, V2I and I2I networks and allow vehicles to get their security services in different area.

1.5 Thesis Contribution

This thesis proposes VANET based on the cloud (V2Cloud), and designs a cloud security model framework, which is called VANET-Security as a Service (VSaaS), to manage the security services and provide a secure VANET communication between the different entities, e.g. vehicles and authorities, where this model (VSaaS) is hosted on a cloud. Our objectives include:

- Proposing VANET that depends on the cellular networks, which act as a gateway to the cloud to get the services which include security services.
- Proposing VANET-Security as a Service (VSaaS) model framework. VSaaS is responsible for:
    - Vehicles and authorities registration.
    - Key Management mechanisms, to generate keys for different entities and renew the keys when they become expired.
    - Authenticating the vehicles and their information messages, and authenticate the authorities that interacting with the VSaaS too.

    - Vehicle identity identification mechanism, to preserve the privacy and enable the traceability done only by the trusted authorities that have a permission to track vehicles.
    - Providing a security access list to manage the permissions among the different entities.
    - Providing a mechanism to revoke the misbehaved vehicle and the compromised authority.
    - Providing modules to process the Vehicle Information Messages (VIMs), which are sent by vehicles as coarse-information messages, and to construct fine-information messages, which called Traffic Information Messages (TIMs), they are disseminated to the vehicles based on their locations.
- Investigating the performance metrics measurements through the NS2, SUMO and the Trans simulations, to evaluate the security overhead of the secure Vehicle Information Messages (VIMs), which are sent by the vehicles to the cloud as a coarse-grained information. Moreover, we analyze our proposed model framework (VSaaS) against the security requirements in the VANET.

1.6 Limitations

In VANET, testing new protocols in real world is important but expensive because it needs a large number of vehicles equipped with special devices to participate in the experiment. This is difficultly and costly much time and money. Safety of a driver is important, but these types of experiments can expose their lives to dangers. To avoid the previous issues and satisfy the concept of *"Proof of Concepts"*, researchers depend on the simulations.

1.7 Thesis Structure

The rest of this thesis is organized as follows. Chapter 2 describes a literature review on VANET Security, Merging Cloud Computing with VANET and VANET-cloud Security. Chapter 3 describes the proposed security model framework in details. Chapter 4 introduces the simulation works. Chapter 5 presents the simulation's results that demonstrate the effectiveness of our model. Chapter 6 concludes our work and provides directions for future work.

Chapter 2
Literature Review

Chapter 2
Literature Review

This chapter includes the reviews of the previous works by others that are relevant to our research. We can classify the related previous works into three catagories:

1. VANET Security: which reviewing the researchers' efforts to secure VANET communication.
2. Merging Cloud Computing with VANET: which reviwing the impacts of the cloud computing on the VANET.
3. VANET-Cloud Security: which reviewing the researchers' efforts to secure VANET that used the cloud.

Finally, we summarize the related work to focous on our work scope.

2.1 VANET Security

Early papers proposed using pseudonyms to keep the privacy into consideration like (Raya, and Hubaux, 2007). Pseudonyms which are defined as, many short-lifetime certificates (private-public key pairs) installed on each vehicle by an authority, where these pseudonyms are used in one period and not be used again. This method protects the vehicle identity from being tracked by the unauthorized observers, but there is one major problem which is the link-ability of the pseudonyms. The attacker may identify the target vehicle by linking the previous pseudonym with the current one by the temporal or spatial locality. Also this approach has many other problems, for example, a large storage space is needed at each vehicle. Moreover, including the certificate in the safety message leads to larger message size and needs more computations to verify every certificate at the receiver side. In addition, the big number of certificates in the authority of all vehicles, causes a big overhead. For liability, the authority should store all these keys to identify the misbehaving vehicle. Moreover, the authority needs to search in a very huge number of keys and that costs a time.

The proposed approach in (Burmester, Magkos and Chrissikopoulos, 2008) was aimed to reduce the large number of pseudonyms which are preloaded on each vehicle. The approach reduced the number to a half, on the average. It depended on using two certificates: the encryption certificate and the signing certificate.

To solve the link-ability problem, some approaches like in (Buttyan, Holczer and Vajda, 2007) proposed a strategy called “hiding in crowd”. In this approach, the pseudonyms are updated regularly according to the spatial or temporal criteria. But, there are some situations that the link-ability is unavoidable in them. One of these situations is driving on a long road without junctions. In this case, the vehicle can be traced or linked to its group in spite of changing its pseudonyms.

Other approaches like in (Huang, Matsuura, Yamane and Sezaki ,2005; Sampigethaya, Huang, Matsuura, Poovendran and Sezaki, 2005) try to solve the locality problem by using a random silent period among the changing of the pseudonyms. In the silent period, vehicle does not transmit any message. The period duration should be random and short. It is hard to link between vehicles before and after the silent period. In this approach, the vehicles must change their pseudonyms in adjacent times, but it is not practical, because of the need of broadcasting the safety messages regularly.

Another solution in (Freudiger, Raya and Feleghhazi, 2007; Buttyán, Holczer and Vajda, 2007) proposed vehicles belong 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 the Identity-Based Cryptography (IBC), where the certificates are not needed for the authentication. IBC was proposed in 1984 by (Shamir, 1984). IBC differs from the public key infrastructure (PKI). In 2001, (Boneh and Franklin, 2001) introduced the first functional and efficient identity-based encryption scheme that was based on bilinear pairings property of the elliptic curve.

Authors (Kamat, Baliga, and Trappe, 2006 ; Kamat, Baliga, and Trappe, 2008) proposed an approach based on identity-based cryptography (IBC), which provides the authentication, non-repudiation and the 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 in some cases. Other approaches were proposed like in (Lai, Chang, and Lu, 2009 ; Sun, Zhang, Zhang and Fang, 2010), they try to avoid the disadvantages of the previous approach.

Another architecture is to use a group signature approach as in (Guo, Baugh and Wang, 2007 ; Lin, Sun, Ho and Shen, 2007). In this approach, vehicles are arranged into groups. Each group has a group manager. The manger is responsible of the signing vehicle messages. The identity of the vehicle can be detected only by the group manager. Another group-based approach is described in (Calandriello, Papadimitratos, Hubaux and Lioy, 2007). In this approach, the group manger signs the vehicles pseudonyms to reduce the certificate authority workload. Each vehicle produces its pseudonyms and signs its messages. But it is difficult to achieve that in a dynamic VANET, because of the size, membership revocation and the dynamic membership (new nodes enter the group and old nodes leave the group) that will increase the complexity and overheads.

Another architecture which does not depend on pseudonyms are described in (Zhang, Lin, Lu and Ho, 2008). This approach uses Hash-base Message Authentication Code (HMAC). Before a vehicle sends a message, it requests a symmetric key from the RSU to use it in the HMAC code. Then, the vehicle signs its message by the HMAC code. The receiver vehicle authenticates the message from the adjacent RSU. This approach offers anonymity but it depends highly on the RSU which may be not available.

Some papers do not use the pseudonyms or the groups to preserve the privacy as proposed in (Bayrak and Acarman, 2010). It proposes a shared private/public key which is given to all of the legitimate vehicles. This key is renewed regularly by an authority where each vehicle has its own public/private key to communicate with the authority.

2.2 Merging Cloud Computing with VANET

Among the existing works, authors (Olariu, Hristov and Yan, 2012) proposed a new concept called Vehicular Cloud (VC). VC used underutilized vehicle resources to form a cloud by aggregating vehicular computing resources. The authors considered that VC refers to a group of large autonomous vehicles included the computing, sensing, communication, and physical resources, where they could be coordinated and dynamically allocated to end users. It is worth to note that the proposed system did not take the advantage of the conventional cloud, and was only based on the vehicular resources. In contrast, VC resources cannot always be switched on, and often require the authorization of the vehicle's owner, which can be absent if the vehicle is in a steady state (e.g., vehicles in a parking lot).

Authors (Hussain et al., 2012) divided the VANET clouds into three major clouds: Vehicular Clouds (VCs), Vehicles using Clouds (VuCs) and Hybrid Clouds (HCs). The VC is subdivided into two categories: a static cloud which refers to stationary vehicles providing cloud services, and a dynamic cloud which is set up on the demand in an ad hoc manner. A VuC allows a VANET to connect to the traditional cloud with RSUs, whereas the HC is a combination of VC and VuC. Moreover, the vehicles can only interact with the traditional cloud through RSUs, which act as gateways. However, vehicles cannot be connected to the traditional cloud if the RSUs are not available, as in rural areas.

Authors (Mershad and Artail ,2013) addressed the problem of enabling the vehicles in the VANET to discover their needed services from the mobile cloud servers, which are moving nearby. The authors proposed a system called CROWN, which depends on the RSUs that act as cloud directories and interfaces. To achieve that, RSUs make their recorded data available to enable vehicles to discover the required cloud services within the area that covered by the RSU.

To provide the safety and non-safety services in the vehicular applications, authors (Baby, Sabareesh, Saravanaguru and Thangavelu, 2013) proposed the use of cloud computing services via RSUs. (Vehicular Cloud for Roadside) VCR scenarios architecture was proposed to allow vehicles to make benefits from the private and public vehicular cloud services. The previous efforts can be considered as help systems for vehicles, to access the conventional cloud through the RSUs via a cloud gateway, in order to find the requested cloud service without using any mobile computing resources.

A pure cloud formed by the vehicles which has been proposed in (Zingirian and Valenti, 2012). It is a new service paradigm called Sensor as a Service (SenaaS) for the vehicle communication platforms, it makes their components available, including vehicle sensors and devices, to third-party vehicle monitoring applications, as cloud computing resources called sensor-cloud service. This proposal lacks of the use of the traditional cloud to improve the computing capacity which is usually requested by vehicles.

To deal with the issue of the vehicles avoiding obstacles, a cloud-assisted system for autonomous driving was proposed in (Kumar, Gollakota and Katabi, 2012) and called Carcel. Carcel is a system that enables the cloud to collect information from the autonomous vehicle sensors as well as from the roadside infrastructure, to help vehicles avoiding obstacles, such as

pedestrians and other vehicles, which may not be directly detected by the sensors on the vehicle.

Authors (Lin, Shen and Weng, 2013) addressed the issue of seamless access to the Internet by making the use of cloud-based VANETs. In this study, the authors proposed a cloud-supported gateway model, which is called Gateway as a Service (GaaS), in order to provide an efficient gateway connectivity and to enhance the Internet usage experience for the vehicular networks.

2.3 VANET Cloud Seccurity

Authors (Rangarajan, Verma, Kannan, Sharma and Schoen, 2011) dealt with the cloud security issue for vehicular networks by proposing a new secure provisioning model called Vehicle-to Cloud (V2C). V2C is composed of a provisioning infrastructure, which links two levels, the automobile user and the infrastructure provider. In the proposed model, the authors integrated three security modules to enhance the security, an authentication module, an authorization and access control policies module and an assurance module. The authentication module manages the identities and authenticates the entities in V2C. The authorization and access control policies modules set the access control policies for every automobile user. To correlate management actions with the desired requirements, the assurance module is deployed throughout V2C. V2C focuses on the cloud services required by the automobile users, and is served via the traditional cloud. This proposal is not satisfy the privacy preservation and the other security requirements in the VANET.

GeoEncrypt (Geolock) in the VANETs has been proposed in (Yan and Olariu, 2009). The idea is to use the geographic location of a vehicle to generate a secret key. Messages are encrypted with the secret key, and the encoded texts are sent to the receiving vehicles. The receiving vehicles must be physically presented in a certain geographic region specified by the sender, to be able to decrypt the message. They are proposed as a future work to integrate this model into security methods, and the shape of the decryption region will be extended from the square shape to any shape in a professional manner.

Authors (Hussain, Abbas, Son and Oh, 2013), considered the concept of VuC framework and proposed another layer named TIaaS (Traffic Information as a Service) on the top of the Cloud Computing Stack. The service offers fine-grained traffic information for all the vehicles which are subscribed to TIaaS from the Cloud. The authors have proposed the Geolock-based encryption to provide the security, privacy and the conditional anonymity.

Authors (Mallisseryet, Pai, Pai and Smitha, 2014) have proposed the Cloud Enabled Secure Communication in the VANET as a method to classify the VANET's messages. In addition, the vehicles and the RSU use the cloud as the medium of storage. This proposed algorithm verifies the identity and authenticity of the vehicles and messages exchanged. The message exchanged and certificates used in VANET are encrypted by using the geolocation key of the RSU. The use of geolocation key provides a location confidentiality against vehicles outside. This method does not satisfy the authorization, privacy, non-repudiation, availability and the revocation.

The work in (Olariuet, Hristov and Yan, 2013) illustrates the power of the VC concept by enumerating a numerous application scenarios, for example, Remote Configuration and Car Performance Checking, Big traffic data analysis, Smart location-based advertisements and Vehicle Witnesses. The authors have emphasized more research challenges in the vehicular cloud including security issues.

Another work in (Yan, Wen, Olariu and Weigle, 2012) proposed that the cloud is associated with a number of grids. A city or traffic area is partitioned into grids. The grid size is predefined with two GPS coordinates. Each cell is associated with a virtual machine in the cloud. The virtual machine can dynamically requests resources from the cloud. Therefore, the traffic of the whole city can be mapped to the cloud. The customized security protocols can be configured and replaced in the VSecurity module.

Authors (Mishra, Panigrahy, Tripathy, Jena and Jena, 2011) proposed a protocol to ensure both the message authentication and the privacy preservation. The proposed scheme is based on a secure elliptic curve digital signature algorithm approach. Here, the authors have considered that the transport authority is sending all the vehicle registration details to the RSU. This can be considered as an invasion of privacy on the vehicular users.

Authors (Serna, Luna and Medina, 2008) proposed the basis of privacy mechanism that uses an authorization paradigm based on a Mandatory Access Control model, and a novel mechanism that propagates trust information based on a vehicles geolocation. With the change of the geographical location, the trust information is passed to a new regional CA. The authors have considered that the geographic location change happens only when a vehicle crosses the border of another country.

2.4 Summary

As we have shown in the previous related works, the security aspects in VANET communication have not been fully explored in one security model framework but the researchers proposed solutions to solve the security problems as individuals.

In spite of all the efforts which had been made in recent years in the field of VANET and its security, researchers still depend directly on the existence of the Road Side Units (RSUs), and the local Certified Authorities (CAs) to provide VANET services including the security services. According to our survey on the previous related works as explained in the next chapter, this dependency is facing many problems as,

- The availability issues: the RSUs which rarely exist in the highway roads and rural areas causes the services such as security services, which are available only in limited regions. Also, making these services available in different regions needs additional costs and efforts to deploy new RSUs, and connect both of the RSUs and the authorities into a single IP network.
- As to business viewpoint, RSUs deployment is costly.
- There is a need to a central management architecture to ensure the availability of the services when a vehicle moves from a region to another. Also, we need mechanisms to make a trust between RSUs.
- These RSUs are not efficient when the number of nodes (vehicles) is very large. It consumes channel bandwidth and effects network performance.
- Highly computations are needed in the vehicles to analyze the collected data and produce fine human readable information.
- Traditional VANET is not suitable for the cheaper vehicles, which lack of proper hardware and sensors to participate into VANET and make its benefits.

A number of authors pointed out in their works that the allocated bandwidth exceeds far more than the requirements for the VANET safety applications. Thus, the surplus bandwidth opens the doors for new opportunities along with the normal functionality of the VANET (Barberis, Gueli, Minh Tuan, Malnati, and Nassisi, 2011). This gave a motivation to Professor Olariu and his colleagues to envision a paradigm shift from the traditional VANET to the Vehicular Cloud Computing (VCC) by merging the VANET with the Cloud Computing (Olariu, Eltoweissy and Younis, 2011).

But till (February/2015), there was no solid architecture or general model for VCC as mentioned in (Hussain, Rezaeifar and Oh, 2015). Modern vehicles are equipped with permanent Internet by the 3G/4G cellular networks which make the cloud request available even when the RSUs do not exist. In addition to featuring like on-board computational, storage and sensing capabilities, which can be thought as a huge farm of computers, remain idle while the vehicles stay on the road. Moreover, as every vehicle which has Internet connections, it can automatically send the messages and the measurement to the cloud, which is controlled by authorities like police. It is worth to mention that the notation of Cloud Computing idea started from the fact of benefiting from it, by using it as an alternative of investing in infrastructure, business may find it useful to rent the infrastructure, and sometimes the needed software to run their applications. It decreases the number of RSUs as well as giving the vehicles an access to Internet where there is no coverage signal of the RSUs (Al Mamun, Anam, Onik and Esfar-E-Alam, 2012).

In the 12th Annual IEEE Consumer Communication and Networking Communication Conference at 2015, Authors (Mallissery, Pai, Ajam, Pai, and Mouzna, 2015) mentioned that the secure VANET cloud was a challenge task, and there is a need to a compatible security model which worked well with the different manufacturers in different countries to satisfy the security requirements in the VANET communication. The security and privacy challenges, which are faced by the standalone VANET and Cloud Computing, will remain unchanged even when the two technologies are merged to form VANET clouds (Hussain et al., 2012).

However, messages (especially safety-messages), which are sent by vehicles in VANET, should be authenticated because the false or altered messages may lead to bad situations like accidents. It should be aware of many challenges in the VANET security such as privacy, because the drivers want to protect their identities from the others to prevent unauthorized tracking, but at the same time, it (the privacy) is in a conflict with other security attributes like authentication, which makes the design of VANET security model needs extra efforts. Moreover, the desired security model for the VANET should include authentication, authorization, confidentiality, integrity, non-repudiation, revocation and privacy (Samara, Al-Salihy, and Sures, 2010).

Chapter 3
Methodology

VANET Security as a Service (VSaaS)

Chapter3
Methodology
VANET Security as a Service (VSaaS)

3.1 Introduction

Communication in VANET needs to be secure. Although the researchers worked on the VANET security issues as individuals, we propose VANET based on the cloud (V2Cloud), and design a cloud security model framework, which is called VANET-Security as a Service (VSaaS), to manage the security services and provide a secure VANET communication between the different entities, e.g. vehicles and authorities, where this model (VSaaS) is hosted on a cloud. Any VANET's security model or protocol should be satisfy the following requirements and attributes as mentioned in (Samara, Al-Salihy and Sures, 2010) :

1. **Authentication:** The identity of a vehicle should be verified to determine if it is a legitimate vehicle or not. Thus, the sender should be authenticated each message before sending it. This prevents intruders from sending malicious messages.

2. **Authorization:** Authorization establishes what each entity (vehicle, RSU and etc) is allowed to do in the system, e.g. what types of messages it can be send, information update rules and the protocols is allowed to execute.

3. **Data Integrity:** Message integrity is very important. If a received message was altered 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. In addition to, this requirement should detect the message repetition by an attacker.

4. **Non-repudiation:** A misbehaving vehicle may send incorrect information where a vehicle itself is legitimate and the message is consistent. This behavior may lead to bad situations like accidents. The sender should not deny that he sent that message, so it should add a liability to user for the messages which he send.

5. **Privacy:** Drivers want to protect their identities from others. This is a very critical requirement. However, the problem is that privacy conflicts with authentication and non-repudiation concepts. Hence, many researches try to solve that problem.

6. **Confidentiality:** is a vital attribute to keep the content of a message secret if need.

7. **Availability:** The system should be available all the time because the disconnection 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. Also, the availability includes that methods which ensure authorities are available and should be trust each other when a vehicle move from one region to others.

8. **Entity Revocation:** The ability to revoke vehicles or authorities is a very important. For example, when vehicle is engages in malicious activity, it must be revoked.

3.2 Notation Description

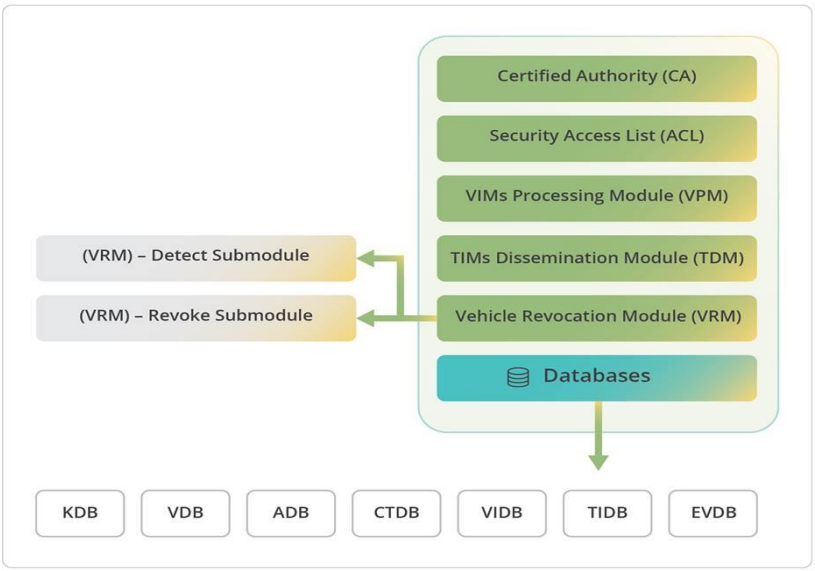
Notation used in this chapter are described in table 3.1

Table (3.1): Notation Description

Notation	Description
VID_N	Identity of vehicle *N*.
AID_M	Identity of authority*M*.
||	Concatenation symbol.
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*.
Pub_M	Public key of Authority *M*.
Prv_M	Private key of Authority *M*.
K_{PRIV}	Privacy shared key for all vehicles
K_D	Dissemenation shared key
K_{SM}	Shared key to encrypt messages between the authority and VSaaS
K_{TMP}	Temporary shared key used one time in authority registration process.
$Enc_{pub}(m, K)$	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 symmetric-key cipher.
$Dec_{sym}(m, K)$	Decrypting m with key K using symmetric-key cipher.
T	Timestamp
EVID	Encrypted vehicle identity value.
Sign(m, K)	Signing message m with key K
Verifysign (SIG, K)	Validating signature SIG with key K.
SP	Each authority has a local security process (SP)
SPID	Identity of each SP

3.3 Proposed VANET Security as a Service (VSaaS)

We propose VSaaS model framework to manage the security services in the VANET based on the cloud, and provide a secure VANET communication between the different entities, e.g. vehicles and authorities. This model framework consists of different modules as follow (See Figure 3.1):



Figure(3.1): Our ProposedVSaaSArchticture

1. Certified Authority (CA): this module is the main one in the VSaaS because it is responsible for:

- Vehicles and Authorities Registration.
- Key Management mechanisms, to generate keys for different entities and renew these keys when they become expired.
- Authenticate the vehicles and their information messages, and authenticate the authorities that interacting with the VSaaS.
- Vehicle identity identification mechanism to preserve the privacy, and enable the traceability done only by the trusted authorities that have a permission to track vehicles.

2. Security Access List (ACL): this module is responsible for:

- Allowing/denying the inter-actions that will be done between the different entities (vehicles, authorities, VSaaS modules).
- Allowing/denying the intra-actions that will be done between the modules within VSaaS.

3. VIMs Processing Module (VPM): this module is responsible for processing the Vehicle Information Messages (VIMs), which are sent by the vehicles as coarse-information messages and constructing fine-information messages, which called Traffic Information Messages (TIMs).

4. TIMs Dissemination Module (TDM): this module is responsible for disseminating the TIMs to the vehicles based on their locations.

5. Vehicle Revocation Module (VRM): this module is divided into two parts: detecting sub-module and revoking sub-module. The first one should be constructed by some algorithms to detect a misbehaved vehicle (this part is out of our scope). The second part is responsible for revoking a misbehaved vehicle when detected (this part is in our scope).

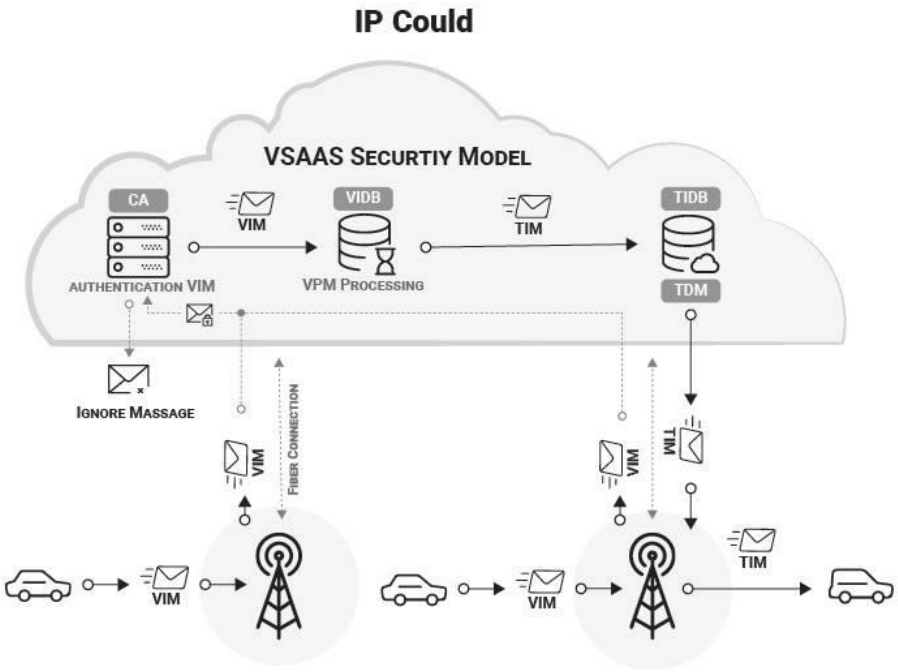
6. Databases: this module stores different types of data as:

- **Keys Database (KDB):** which is used to store different types of keys such as the CA public/private keys and the TDM's shared secret key. This database is managed by CA.
- **Vehicles Database (VDB):** which is used to store all the vehicles information and their keys. This database is managed by the CA.
- **Authorities Database (ADB):** which is used to store all the authorities' information and their keys. This database is managed by the CA.
- **Vehicle Information Messages Database (VIDB):** which is used to store the VIMs. This database is managed by the CA. VPM and VRM have their permission to access this database.
- **Traffic Information Messages Database (TIDB):** which is used to store the TIMs. This database is managed by VPM. TDM has its permission to access this database.
- **Cellular Towers Database (CTDB):** which is used to store all the cellular towers information, their coordinates and routes. This database is managed by system administrators. TDM has its permission to access this database.
- **Event Viewer Database (EVDB):** which is used to store all the events within the VSaaS framework model. System administrators and different modules have a permission to access this database especially for some reporting issues.

3.4 Vehcile Information Messages (VIMs) and Traffic Information Messages (TIMs)

In the traditional VANET, safety messages are the messages or the beacons which are broadcasted by vehicles, to warn the other vehicles about the traffic situations like congestions and collisions. These messages contain information like location, speed, direction and acceleration. These messages are sent into the forms of the V2V and V2I communications. Hence, these messages are divided into two types: periodic and event driven messages, which are sent when a hazardous situation occurs. Periodic messages are considered to be an important type of messages that supports a decision that has been taken in the safety applications. Periodic messages are broadcasted to surrounding vehicles, but it may lead to a wasted bandwidth consumption, especially in the dense environment, in addition to increasing the probability of a storm problem occurring.

In our work, we propose the VIMs, which are shaped in a form of V2Cloud communication, as an alternative of the safety messages, which are shaped in a form of V2V and V2I. VIMs include the current position, speed, direction, timestamp and the heading information, which are sent directly to the cloud infrastructure, and stored in the Vehicle Information Database (VIDB) if they were authenticated by the CA. Also, we propose the vehicles will send asynchronous VIMs when the change in parameters exceeds a certain percentage (needs more experiments to determine it), or when the hazardous conditions are occurred. Other Modules in the VSaaS may have a permission to access the VIDB, such as the VPM which is made to process the VIMs, and construct fine-grained information which called Traffic Information Messages (TIMs). TIMs will be stored in the Traffic Information Database (TIDB). TDM module disseminates the TIMs to the vehicles based on their location (See Figure 3.2).



Figure(3.2): VIM and TIM messages into V2Cloud communication and VSAAS Model

In addition, we propose a Message Type (MT) field in each VANET message sent in the V2Cloud between the different entities. MT is chosen to be a value of 32-bit length. MT is used to identify each message e.g. who are the sender and receiver of the message? And what's the aim of the message. Examples of messages in the proposed VSaaS are shown in table 3.2.

Table (3.2): Example of Messages in the ProposedVSaaS

MT	Sender	Receiver	Message Name	Description
1	Vehicle	CA	VehicleRequestCAKeyUpdate	Vehicle requests CA for new CA's public key
2	CA	Vehicle or ALL	CAKeyUpdateforVehicle	CA sends its CA's public key to specific vehicle or broadcast itto all
3	Vehicle	CA	RequestNewPrivacyKey	Vehicle requests CA for new symmetric privacy key
4	CA	Vehicle or ALL	PrivacyKeyUpdate	CA sends new symmetric privacy key to specific vehicle or broadcast it to all
5	Vehicle	CA	RequestNewDissKey	Vehicle requests CA for new symmetric Dissemination key
6	CA	Vehicle or ALL	DissKeyUpdate	CA sends new symmetric Dissemination key to specific vehicle or broadcast it to all
7	CA	Vehicle	KillVehicle	CA Sends Revoked Message to Vehicle N
8	Vehicle	CA	SecureVehicleInformationMessage	Vehicle sends secure VIMs to CA
9	TDM	Vehicles	SecureTrafficInformationMessage	TDM sends secure TIMs to Vehicles based on their location
10	Authority	CA	AuthorityRegister	Authority requests CA for registration
11	CA	Authority	OkRegister	CA sends secure Information to CA
12	Authority	CA	ACKAuthorityRegister	Authority sends ACK to CA for complete registration process
13	CA	Authority	killAuthority	CA Sends Revoked Message to Authority M
14	Authority	CA	AuthorityRequestCAKeyUpdate	Authority requests CA for new CA's public key
15	CA	Authority	CAKeyUpdateforAuthority	CA sends its CA's public key to specific authority
16	Authority	CA	AuthorityRequestSharedKeyUpdate	Authority requests CA for new shared key
17	CA	Authority	SharedKeyUpdateforAuthority	CA sends its shared key to specific authority
18	TA Authority	CA	Track	TA Authority sends Track message to track vehicle N
19	CA	TA Authority	StartTracking	CA informs TA to start tracking Vehicle N
20	TA Authority	CA	StopTracking	TA informs CA; the tracking Vehicle N mission was completed

3.5 Vehicle Entity in VSaaS

Before giving vehicle N a license to work, it should be registered in the Certified Authority (CA) by taking its physical Vehicle Identification Number (VIN) in some ways, to ensure that the VIN is true. VIN consists of 17 digits (Mercedes VIN Shopping Tips Website, n.d.) and was officially described in ISO standard 3779 in February 1977, and revised at last in 1983. CA extracts all the vehicle's information from this VIN. Also, the owner's information should be given. Then, CA generates the Vehicle Identification Number (VID_N) and the public/private keys (Pub_N, Prv_N) for the vehicle N. In our work, VID_N is chosen to be a value of 64-bit length. This length can present more than 18 billion of values. May be other values can be used. The 64-bit length ensures that, there are more than 18 billion attempts to guess the VID_N when the brute force attack presented. The public/private keys (Pub_N, Prv_N) have a long lifetime (a year for example). When a vehicle renews its license, the CA will generate and install new keys on the vehicle.

Each vehicle has a Tamper-Proof Device (TPD) installed by the manufacturer, to store all the secret information used in VANET. CA preinstalls the (Pub_N, Prv_N) and Pub_{CA} on each vehicle N's TDP in addition to the VID_N. Hence, CA has public/private keys (Pub_{CA}, Prv_{CA}). The public and private keys for vehicles and the CA are generated according to the public-key cipher algorithms (Rivest, Shamir and Adleman) RSA. Also, we consider the (Pub_{CA}, Prv_{CA}) have a medium lifetime (a month for example). When CA's public and private keys are renewed, CA broadcasts the **CAKeyUpdateforVehicle** message to all the vehicles which contain the new Pub_{CA}. Vehicles that did not receive the **CAKeyUpdateforVehicle** message according to different reasons, can send a **VehicleRequestCAKeyUpdate** to request the new Pub_{CA}.

For liability, vehicles' identities should be added to the vehicles' messages, but this requirement contradicts with the privacy. Therefore, vehicles' identities should be hidden (encrypted) from the others, only CA can identify the vehicles' identities. To solve it, CA generates a symmetric key called Privacy Key K_{PRIV} used to encrypt/decrypt the vehicles' identities. All the registered (trusted) vehicles have the same privacy key K_{PRIV}. This key has a medium lifetime (a month for example). The key size is selected to be 128-bit, which is a common size for the symmetric ciphers (Advanced Encryption Standard) AES. K_{PRIV} is preinstalled on the vehicle N's TDP when the vehicle N is registered with CA. It is worth to mention that the privacy key K_{PRIV} provides authentication and privacy. Authentication is achieved because only the registered and trusted vehicles have this privacy key K_{PRIV}. Using

the same privacy key K_{PRIV} by all the vehicles at the same time to authenticate the messages or a part of them, provides anonymity which achieved the privacy.

When the vehicle N sends a message or requests different keys from the CA, TDP will add Encrypted VID (EVID). EVID value is produced by concatenating the VID_N to the current reading from the tamper GPS (xy-coordinates) installed by the manufacturer, then encrypting the all with the privacy key K_{PRIV}. Show pseudocodes in table 3.3. EVID value is a portion of any message sent to the CA, where the entire message should be encrypted by the Pub_{CA}. As a result, only CA can decrypt the message by its private key Prv_{CA}. Concatenating the (xy-coordinates) with the VID_N before encryption, should ensure that the EVID value must be different in each message, and mitigates the linking between the two messages generated from the same vehicle. In addition, the reading from the tamper GPS (xy-coordinates) ensures that the message was sent from true location of vehicle and not from other place.

Table (3.3): Pseudocodes to encrypt/decrypt Vehicle Identity

Pseudocode: Encrypt Vehicle Identity
Input: VID_N
Output: Encrypted VID_N (EVID)
1. Read xy-coordinates from Tamper GPS
2. EVID = Enc_{sym} (VID_N|| xy, K_{PRIV})
3. Return EVID

Pseudocode: Decrypt Vehicle Identity
Input: Encrypted VIDN (EVID)
Output: VID_N
1. VID_N|| xy = Dec_{sym} (EVID, K_{PRIV})
2. Extract VID_N
3.Return VID_N

To send secure VIMs, vehicle N's TDP concatenates the message (m), the message type (MT), the time stamp (t) and the EVID together. Then encrypting the all by the CA's public key Pub_{CA} to form a secure VIM. Vehicles send secure VIMs to the CA to verify and authenticate them. CA receives the secure VIMs and decrypts them by its private key Prv_{CA}. CA validates the timestamp (t) and extracts the EVID to authenticate the Vehicle Identity VID_N. If (t) and (VID_N) are valid, CA extracts the message (m) and stores it in the VIDB. VPM processes VIMs to construct fine-grained information, which called Traffic Information Messages (TIMs), and stores them in the TIDB. TDM gets TIM and concatenates it to (t) and (MT), then encrypts the all by a symmetric shared key called Dissemination Key K_D to form the secure TIMs. After that, TDM determines the route of the secure TIMs based on the location of the cellular towers coordinates, which are stored in a Cellular Tower Database (CTDB) to

disseminate them. Vehicles that receive the secure TIMs, decrypt them by the Dissemination Key (K_D) and verify timestamp (t) to get the TIMs. Show pseudocodes in table 3.4.

Table (3.4): Pseudocodes to Send VIMs and Disseminate TIMs

Pseudocode: Sending Secure Vehicle Information Message (VIM)
Input: Vehicle Information Message m and VID_N
Output: secure VIM
1.Get current timestamp t
2. Set Message Type (MT)=8
3. EVID = Encrypt Vehicle Identity Pseudocode (VID_N)
4. M = m || t || MT || EVID
5. secure VIM = Enc_{pub} (M, Pub_{CA})
6.Return secure VIM

Pseudocode: CA Verifying Secured Vehicle Information Message (VIM)
Input: secure VIM
Output: Vehicle Information Message m and VID_N, or null
1. M = Dec_{prv} (secure VIM, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. VID_N = Decrypt Vehicle Identity Pseudocode (EVID)
6. If VID_N is false, then return null and stop
7. Extract m from M
8. Return m and VID_N
9. Store all information into VIDB.

Pseudocode: TDM Disseminating Secure Traffic Information Message (TIM)
Input: Traffic Information Message m
Output: secure TIM and route
1.Get current timestamp t
2. Set Message Type (MT)=9
4. M = m || t || MT
5. secure TIM = Enc_{sym} (M, K_D)
6. Determine the route for this TIM based on the location of cellular towers coordinates stored in CTDB.
7. Return secure TIM and route

Pseudocode: Vehicle Verifying Secured Traffic Information Message (TIM)
Input: secure TIM
Output: Traffic Information Message m or null
1. M = Dec_{sym} (secure TIM, K_D)
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
6. Extract m from M
7. Return m

When the vehicle N is registered with the CA, it pre installs the K_D on the vehicle N's TDP in addition to the previous secret information: (Pub_N, Prv_N), Pub_{CA}, VID_N and K_{PRIV}. This key has

a short lifetime (a day for example). The key size is selected to be 128-bit which the common size is for the symmetric ciphers AES. When the CA renews the K_D, it broadcasts the **DissKeyUpdate** message, which contains the new Dissemination Key (K_D), to all the vehicles. Vehicles that did not receive the **DissKeyUpdate** message because of the different reasons, can request the new Dissemination Key. In addition to that, CA will pass the new key to the DTM. Hence, when the CA sends the new KD, it is signed by the CA's private key to ensure the CA authentication, and encrypted by the key K_{PRIV}, to achieve the confidentiality and vehicles authentication.

All the messages in the VANET environment should be protected against the replay attack. This protection is achieved by adding the time information to the message which is called timestamp (t). When the vehicle or the CA receives a message, it will check the validity of its timestamp (t). TDP has an internal clock .TDP is responsible for adding timestamps to the message before sending them in addition to check the validity of timestamps in the received messages.

3.6 Vehicle Revocation

When a misbehaved vehicle is detected, it should be revoked by Vehicle Revocation Module (VRM). The algorithm which determines the misbehaved vehicle, is out of our work scope. In our work, we aimed to design a security model framework for the (V2Cloud).

In the previous works, where there is no general VANET security framework model, each vehicle should have a Revocation Key List, and check if the received messages were generated by the trusted vehicles or not. This was done by different approaches, but these approaches had many issues. For example, it needs a large storage space at each vehicle. Moreover, including more parameters or keys in the safety messages, leads to larger message size, and needs more computations to verify the messages at the receiver side. In addition, the need of searching in a very huge number of keys, costs more time (Raya, and Hubaux, 2007 ; Burmester, Magkos and Chrissikopoulos, 2008).

In our work, CA is receiving all vehicle messages and storing them in the VIDB. VRM is divided into two parts: detecting sub-module and revoking sub-module. The first one should be constructed by some algorithms to detect the misbehaved vehicle (this part is out of our scope). This sub-module has its permission to access the VIDB to check the behavior and messages of the vehicles, in order to detect any misbehaved vehicle. If it was detected, the second part, which is responsible for revoking the misbehaved vehicle, changes the vehicle

status to invalid, and create the revoked report (RR) that indicates to why the vehicle will be revoked. Then, CA will send a **killVehicle** message to that vehicle. This message includes RR which is signed by the CA's private key (Prv_{CA}), to ensure that the message is generated by the CA. After that, concatenate the (signed RR) with the Message Type (MT), timestamps (t), and encrypt all of them by the vehicle's public key Pub_N, to ensure that only the misbehaved vehicle N can decrypt this message by its private key Prv_N. When the misbehaved vehicle N receives the **killVehicle** message and decrypts it, the vehicle's TDP checks the validity of timestamps and verifies the signed revoked report part by the CA's public key Pub_{CA}. Finally, vehicle N's TDP will stop working and erase all the secret information. Show pseudocodes in table 3.5.

Table (3.5): Pseudocodes to Revoke Vehicle N

Pseudocode: CA Sending Revoked Message to Vehicle N
Input: VID_N and revoke report RR
Output: KillVehicle message
1. Get current timestamp
2. Set Message Type (MT)=7
3. SignRR = $Sign_{prv}$ (RR, Prv_{CA})
4. M = t || MT || SignRR
5. KillVehcile = Enc_{pub} (M, Pub_N)
6. Return KillVehcile

Pseudocode: Vehicle N Receiving Revoked Message
Input: KillVehicle
Output: N'TDP turn-off or null
1. M = Dec_{prv} (KillVehcile, Prv_N)
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$(SignRR, Pub_{CA}) = false, then return null and stop
6. Erase all keys and turn off TDP

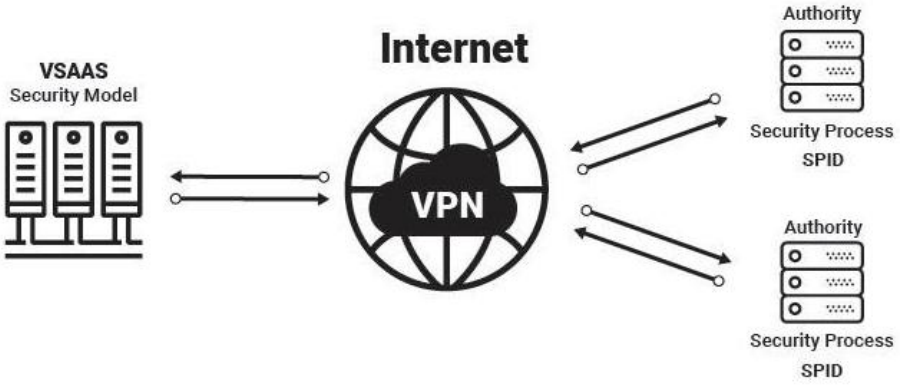
3.7Authority Entity in VSaaS

In our work, we classify the authorities which are registered with the CAs into three different types according to their permission:

1. **Traceability Authority (TA)** which has a permission to track the vehicle identity and location.

2. **Transportation and Traffic Monitor Authority (TTMA)** which has a permission to monitor and analyze the transport traffic, depending on the information recorded in the TIDB.
3. **Manufacturers** which have a permission to provide all the firmware updates, and check the vehicle performance remotely, depending on the sensor messages which are sent periodically by the vehicle itself. (This part is out of our work scope).

We propose a Virtual Private Network (VPN) mechanism to connect these authorities with the VSaaS. The governmental body, which manages the VSaaS, defines the authority M by giving the CA all the authority's information, such as (name, type, address, telephone number, contact person, email, IP address and permissions). The CA generates the Authority Identification Number (AID_M) which is chosen to be a value of 16-bit length. And by this length, we can present more than 65000 of the values. May be other values can be used. Also, CA generates the authority's public-private keys pair (Pub_M, Prv_M). The public and private keys are generated according to the public-key cipher algorithms RSA. Also, we consider the (Pub_M, Prv_M) has a long lifetime (a year for example). Finally, CA generates the secret shared key K_{SM} that will be used to exchange the information and messages between the authority M and the VSaaS modules such as CA. The key K_{SM}'s size is selected to be 128-bit, which is the common size for the symmetric ciphers AES. This key has a medium lifetime (a month for example). Each authority has a local security process (SP), which has a specific identifier SPID chosen to be a value of 16-bit length. And by this length, we can present more than 65000 of the values. May be other values can be used. The local SP is responsible for interacting with the VSaaS modules in a secure way. Hence, the current Pub_{CA} and Authority Identification Number (AID_M) have been given locally to the authority M when the site has been installed (See Figure 3.3).



Figure(3.3): Authority Entity Communicates with VSaaS

To register the authority, the new authority generates a temporary symmetric key K_{TMP} used only in the registration process (used for one time). The key size is selected to be 128-bit which the common size is for the symmetric ciphers AES. The new authority sends an **AuthorityRegister** message to the CA where the message consists of MT, t, AID_M and K_{TMP}. Then, encrypting the all by the CAs' public key Pub_{CA} to ensure that only the CA can decrypt this message. When CA received the **AuthorityRegister** message, CA decrypts it by its private key Prv_{CA}. Then, it validates the timestamp (t). After that, the CA extracts the AID_M and the temporary symmetric key K_{TMP} to validate them. Finally, the CA forms the **OkRegister** message that consists of MT, t and a signed part which consists of the Pub_M, Prv_M, SPID and the K_{SM} which also are signed by the Prv_{CA}, then encrypting the all by K_{TMP}. When the authority M receives the **OkRegister** message, it decrypts **OkRegister** message by the temporary symmetric key K_{TMP}, and verifies the signed portion by the Pub_{CA} to ensure that all the secret keys and information aregenerated by the CA. Then, authority M stores all the secret keys and information which are sent by the CA and it also erases the K_{TMP}. Now, authority M creates a local security process SP with SPID to interact with the CA in a secure way. Authority M' SP forms **ACKAuthorityRegister** message that consists of the MT and t. This message is encrypted by K_{SM}. Finally, the CA verifies the **ACKAuthorityRegister** message, and erases the K_{TMP} and be ready to exchange the messages with authority M. Show pseudocodes in table 3.6. Moreover, authorities can request a new CA's public key and a shared key K_{SM} when they are expired.

Table (3.6): Authority Registration Pseudocodes

Pseudocode: Authority Sending Registration Request
Input: Pub_{CA} and AID_M
Output: AuthorityRegister message
1.Authority generates temporary symmetric key K_{TMP}
2. Message Type =10
3. Get current timestamp t
4. M =MT || t || AID_M|| K_{TMP}
5. AuthorityRegister = Enc_{pub} (M, Pub_{CA})
6. Return AuthorityRegister

Pseudocode: CA Receiving Authority Registration Request and Sending Security Information to Authority
Input: AuthorityRegister
Output: OkRegister or null
1. M = Dec_{prv}(AuthorityRegister, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. CA Extracts AID_M and K_{TMP}.

6. CA Validates AID_M= false, then return null and stop
7. CA set Message Type =11
8. Get current timestamp t
9. SignokRegister= $Sign_{prv}$((Pub_M, Prv_M) || SPID || K_{SM} , Prv_{CA})
10. M = MT || t || SignokRegister
11. OKRegister = Enc_{sym} (M, K_{TMP}).
12. Return OKRegister to authority M

Pseudocode: Authority Receiving Security Information from CA and send back ACK

Input: okRegister
Output: ACKAuthorityRegister message or null
1. M = Dec_{sym} (okRegister, K_{TMP})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5.$VerifySign_{pub}$ (SignokRegister, Pub_{CA}) =false, then return null and stop
6. Extract and Store Security Information: (Pub_M, Prv_M) || SPID || K_{SM}
7. Erase temporary symmetric key K_{TMP}
8.Authority creates local security process with SPID
9. M' SP set Message Type =12
10. Get current timestamp t
11. M =MT || t
12. ACKAuthorityRegister = Enc_{sym} (M, K_{SM})
13. Return ACKAuthorityRegister

If any authority is compromised, CA will send a **killAuthority** message to this authority. This message includes revoked report (RR) that indicates to why the vehicle will be revoked which is signed by the CA's private key (Prv_{CA}), to ensure that the message is generated by the CA. After that, concatenate the (signed RR) with the Message Type (MT), timestamps (t), and encrypt all of them by the authority's public key Pub_M, to ensure that only the compromised authority M can decrypt this message by its private key Prv_M. When the compromised authority M receives the **killVehicle** message and decrypts it, the authority checks the validity of timestamps and verifies the signed revoked report part by the CA's public key Pub_{CA}. Finaly, the authority Mits local security process and erase all the keys. Show pseudocodes in table 3.7. The compromised authority needs to register again with the CA according to the governmental security rules.

Table (3.7): Pseudocodes to Revoke Authority M

Pseudocode: CA Sending Revoked Message to Authority M

Input: AID_M and revoke report RR
Output: KillAuthority message
1. Get current timestamp
2. Set Message Type (MT)=13

3. SignRR = $Sign_{prv}$(RR, Prv_{CA})
4. M = t || MT || SignRR
5. KillAuthority= Enc_{pub}(M, Pub_{M})
6. Return KillAuthority

Pseudocode: AuthorityM Receiving Revoked Message
Input: KillAuthority
Output: Destroy M' local security processor null
1. M = Dec_{prv} (KillVehcile, Prv_{M})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$(SignRR, Pub_{CA}) = false, then return null and stop
6. Erase all keys and Destroy M' local security process

Traceability Authority (TA) is an authority which has a permission to track the vehicle identity and location. Today, tracking vehicles depends on the Vehicle Identification Number (VID_N). Vehicle Information Messages (VIMs) are stored in the (VIDB) that is managed by CA. To track vehicle N, TA should have a permission to access the required fields from the VIDB's records such as VID_N, xy-coordinates, speed and etc. When TA has an order to track the vehicle N, which has the VID_N, TA's local security process SP will send a **Track message** to the CA, which consists of the MT, t, AID_M and the VID_N, then encrypting the all by the Pub_{CA}, to ensure that only the CA can decrypt this message. When the CA receives this message, it decrypts the message by the CA' private key and check the validity of the t, then, extracts the AID_M and the VID_N to verify them.

The CA informs the TA by the **startTracking** message to start tracking vehicle N. This message includes MT, t and VID_N encrypted by the TA's public key to ensure that only the TA can decrypt this message. When TA receives this message, its local security process starts to query and access all the needed information to track the vehicle N. Hence, the CA and authority TA' SP are using the shared secret key (K_{SM}) to encrypt all the traffic between them, to track the vehicle N through its VID_N. We proposed asymmetric key encryption because it is more efficient and faster than the public/private keys. When the TA completes the tracking order, it sends a **stopTracking** message to the CA. Show pseudocodes in table 3.8.

Table (3.8): Pseudocodes to Track Vehcile N by TA Authority

Pseudocode: TA Sending Request to CA to Track Vehicle N
Input: VID_N and AID_M
Output: Track message
1. Get current timestamp t
2. Message Type =18
3. M =MT || t || AID_M|| VID_N
4. Track = Enc_{pub}(M, Pub_{CA})

5. Return Track

Pseudocode: CA Receiving Track Vehicle N Request from TA

Input: Track
Output: startTracking or null
1. M = Dec_{prv} (Track, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. AID_M= false, then return null and stop
6. VID_N= false, then return null and stop
7. CA set Message Type =19
8. Get current timestamp t
9. M = MT || t || VID_N
10. startTracking = Enc_{pub}(M, Pub_{TA})
11. Return startTracking

Pseudocode: TA Receiving Start Tracking Vehicle N

Input: startTracking
Output: TA' local security process can access information in VIDB in secure way
1. M = Dec_{prv} (startTracking, Prv_{TA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. TA' local security process can access information in VIDB in secure way using shared secret key (K_{SM}).

Pseudocode: TA Sending Stop Tracking Vehicle N to CA

Input: VID_N
Output: stopTracking or null
1. Message Type =20
2. Get current timestamp t
3. M = MT || t || VID_N
4. StopTracking = Enc_{pub}(M, Pub_{CA})
5. Return StopTracking

Pseudocode: CA Receiving Stop Tracking Vehicle N

Input: stopTracking
Output:
1. M = Dec_{prv}(StopTracking, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. The mission was completed

Transportation and Traffic Monitor Authority (TTMA) is the authority which has a permission to monitor and analyze the transportation traffic by depending on the information recorded in the TIDB. Therefore, the TTMA has a permission to read the information from the TIDB through the authority local security process. These information may contain the VID_N if the authority has a traceability permission, and it contain encrypted VID_N if the authority has not

this permission. Moreover, the information forwarded from the CA to the TTMA should be encrypted by the secret shared key K_{SM}. The TTMA gets the secret shared key K_{SM} in the registration process.

3.8 Security Access List (ACL)

This module represents a set of permissions and rules to Allow/deny the inter-actions between the different entities (vehicles, authorities, VSaaS modules), and the intra-actions between the modules within the VSaaS. Our design of the VSaaS is modular. In the future, any type of the authorities, databases, new VSaaS's modules, can be easily and smoothly added to make specific tasks by defining its permissions, to interact with the different entities, databases and modules in the VSaaS. Table 3.9 shows a simple way to define a permission for each authority according to its functions. Also, Table 3.10 shows the simple shape of the entities and modules permission against the different databases.

Table (3.9): Example of defining permissions to each authority

AIDM	Traceability	Traffic Monitor	Manufacturer
0010101011110001	1	0	0
0010101011111111	0	1	0

Table (3.10): Examples of entities and modules permissions

Module/Database	KDB	VDB	ADB	VIDB	TIDB	CTBD
CA	Full	Full	Full	Full		Read
VRM		Read		Read		Read
VPM		Read		Read	Full	Read
TDM					Read	Read
TA Authority		Read		Read		Read
TTMA Authority					Read	Read

3.9 Pseudocodes to Update Keys

This section describes different pseudocodes to update and renew the following keys for vechiles: CA's public key (Pub_{CA}), privacy key (K_{PRIV}**)** which is used to encrypt/decrypt the vehicles' identities and dissemination key (K_D) which is used to disseminate TIMs. Also, this section describes different pseudocodes to updates and renews the following keys for authorities: CA's public key (Pub_{CA}) and shared secret key (K_{SM}).

3.9.1 Update CA's Public Key for Vehicles

CA has public/private keys (Pub_{CA}, Prv_{CA}). They are generated according to the public-key cipher algorithms RSA. Also, we consider the (Pub_{CA}, Prv_{CA}) have a medium lifetime (a month for example). When CA's public and private keys are renewed, CA broadcasts the **CAKeyUpdateforVehicle** message to all the vehicles which contain the new Pub_{CA}. Vehicles that did not receive the **CAKeyUpdateforVehicle** message according to different reasons, can send a**VehicleRequestCAKeyUpdate** to request the new Pub_{CA}. Pseudocodes are shown in table 3.11. The new CA's public key is signed by the old CA's private key to ensure that the new key is generated by the CA. **CAKeyUpdateforVehicle** is concatenated from the (signed new CA's Public Key) with the Message Type (MT), timestamps (t), and encrypt all of them by the privacy key K_{PRIV}, to ensure that only the trusted and registered vehciles can decrypt this message.When the vehicle N receives the **CAKeyUpdateforVehicle** message and decrypts it, the vehicle checks the validity of timestamps and verifies the signed new CA's public key by the old CA's public key. Finally, the vehicle extracts new CA's public key, stores it and erases the old one.

Table (3.11): Pseudocodes to distribute new CA's Public Key

Pseudocode: Vehicles Sending Request for New CA's Public Key
Input: VID_N
Output: VehicleRequestCAKeyUpdate message
1.Get current timestamp t
2. Set Message Type (MT)=1
3. EVID = Encrypt Vehicle Identity Pseudocode (VID_N)
4. M = t || MT || EVID
5. VehicleRequestCAKeyUpdate= Enc_{pub} (M, Pub_{CA})
6.Return VehicleRequestCAKeyUpdate

Pseudocode: CA Receivingthe CA's Public Key Request from Vehicle N
Input: VehicleRequestCAKeyUpdate
Output: call pseudocode for CAKeyUpdateforVehicle or null
1. M = Dec_{prv} (VehicleRequestCAKeyUpdate, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. VID_N = Decrypt Vehicle Identity Pseudocode (EVID)
6. If VID_N is false, then return null and stop
7. call pseudocode for CAKeyUpdateforVehicle

Pseudocode: CA Sending new CA's Public Key to Vehicle
Input:CA's New Public Key (Pub_{CA})
Output: CAKeyUpdateforVehicle
1.Get current timestamp t

2. Set Message Type (MT)=2
3. SPub =$Sign_{prv}$ (new Pub_{CA}, oldPrv_{CA})
4. M = t || MT || SPub
5. CAKeyUpdateforVehicle = Enc_{sym} (M, K_{PRIV})
6. Return CAKeyUpdateforVehicle

Pseudocode: Vehicle Receiving the new CA's Public Key
Input: CAKeyUpdateforVehicle
Output: CA's New Public Key Pub_{CA} or null
1. M = Dec_{sym} (CAKeyUpdateforVehicle, K_{PRIV})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$ (SPub, oldPub_{CA}) = false, then return null and stop
6. Return newPub_{CA}, store it and erase the old one

3.9.2 Update Privacy Key (K_{Priv})

CA generates the privacy key (K_{PRIV}) that used to encrypt/decrypt the vehicles' identities. It is generated according to the symmetric cipher algorithms AES. Also, we consider this key has a medium lifetime (a month for example). K_{PRIV} is preinstalled on the vehicle N's TDP when the vehicle N is registered with CA. When privacy key (K_{PRIV}) is renewed, CA broadcasts the **PrivacyKeyUpdate** message to all the vehicles which contain the new K_{PRIV}. Vehicles that did not receive the **PrivacyKeyUpdate** message according to different reasons, can send a **VehicleRequestNewPrivacyKey** to request the new K_{PRIV}. Pseudocodes are shown in table 3.12. The new privacy key is signed by the current CA's private key to ensure that the new privacy key is generated by the CA. **PrivacyKeyUpdate** is concatenated from the (signed new privacy key) with the Message Type (MT), timestamps (t), and encrypt all of them by the old privacy keyK_{PRIV}, to ensure that only the trusted and registered vehciles can decrypt this message. When the vehicle N receives the **PrivacyKeyUpdate** message and decrypts it, the vehicle checks the validity of timestamps and verifies the signed new privacy key by the current CA's public key. Finally, the vehicle extracts new privacy key K_{PRIV}, stores it and erases the old one.

Table (3.12): Pseudocodes to distribute new privacy key K_{PRIV}

Pseudocode: Vehicles Sending Request for New Privacy Key K_{PRIV}
Input: VID_N
Output: VehicleRequestPrivacyKeyUpdate message
1.Get current timestamp t
2. Set Message Type (MT)=3
3. EVID = Encrypt Vehicle Identity Pseudocode (VID_N)
4. M = t || MT || EVID
5. VehicleRequestPrivacyKeyUpdate = Enc_{pub} (M, Pub_{CA})

6.Return VehicleRequestPrivacyKeyUpdate

Pseudocode: CA Receiving the Privacy Key Request from Vehicle N

Input: VehicleRequestPrivacyKeyUpdate
Output: call pseudocode for PrivacyKeyUpdate or null
1. M = Dec_{prv} (VehicleRequestPrivacyKeyUpdate, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. VID_N = Decrypt Vehicle Identity Pseudocode (EVID)
6. If VID_N is false, then return null and stop
7. call pseudocode for PrivacyKeyUpdate

Pseudocode: CA Sending new Privacy Key to Vehicle

Input: New Privacy Key (K_{PRIV})
Output: PrivacyKeyUpdate
1.Get current timestamp t
2. Set Message Type (MT)= 4
3. SPub =$Sign_{prv}$ (new K_{PRIV}, Prv_{CA})
4. M = t || MT || SPub
5. PrivacyKeyUpdate= Enc_{sym} (M, old K_{PRIV})
6. Return PrivacyKeyUpdate

Pseudocode: Vehicle Receiving the new Privacy Key

Input: PrivacyKeyUpdate
Output: New Privacy Key K_{PRIV}
1. M = Dec_{sym} (PrivacyKeyUpdate, old K_{PRIV})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$ (SPub, Pub_{CA}) = false, then return null and stop
6. Return newK_{PRIV}, store it and erase the old one

3.9.3 Update Dissemination Key (KD)

CA generates the dissemination key (K_D) that used to disseminate TIMs to the vehciles based on their location. It is generated according to the symmetric cipher algorithms AES. Also, we consider this key has a short lifetime (a day for example). K_D is preinstalled on the vehicle N's TDP when the vehicle N is registered with CA. When dissemination key (K_D) is renewed, CA broadcasts the **DissKeyUpdate** message to all the vehicles which contain the new K_D. Vehicles that did not receive the **DissKeyUpdate** message according to different reasons, can send a **VehicleRequestDissKeyUpdate** to request the new K_D. Pseudocodes are shown in table 3.13. The new dissemination key is signed by the current CA's private key to ensure that the new dissemination key is generated by the CA. **DissKeyUpdate** is concatenated from the (signed new dissemination key) with the Message Type (MT), timestamps (t), and encrypt all of them by the current privacy key K_{PRIV}, to ensure that only the trusted and registered vehciles can decrypt this message. When the vehicle N receives the **DissKeyUpdate** message

and decrypts it, the vehicle checks the validity of timestamps and verifies the signed new dissemination key by the current CA's public key. Finally, the vehicle extracts new dissemination key K_D, stores it and erases the old one**.**

Table (3.13): Pseudocodes to distribute new dissemination key K_D

Pseudocode: Vehicles Sending Request for New Dissemination Key K_D
Input: VID_N
Output: VehicleRequestDissKeyUpdate message
1.Get current timestamp t
2. Set Message Type (MT)=5
3. EVID = Encrypt Vehicle Identity Pseudocode (VID_N)
4. M = t || MT || EVID
5. VehicleRequestPrivacyKeyUpdate = Enc_{pub} (M, Pub_{CA})
6.Return VehicleRequestPrivacyKeyUpdate

Pseudocode: CA Receiving the Dissemination Key Request from Vehicle N
Input: VehicleRequestPrivacyKeyUpdate
Output: call pseudocode for DissKeyUpdate or null
1. M = Dec_{prv} (VehicleRequestPrivacyKeyUpdate, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. VID_N = Decrypt Vehicle Identity Pseudocode (EVID)
6. If VID_N is false, then return null and stop
7. call pseudocode for DissKeyUpdate

Pseudocode: CA Sending new Dissemination Key to Vehicle
Input: New Dissemination Key ($K_{D)}$
Output: DissKeyUpdate
1.Get current timestamp t
2. Set Message Type (MT)=6
3. SPub =$Sign_{prv}$ (new K_D, Prv_{CA})
4. M = t || MT || SPub
5. DissKeyUpdate= Enc_{sym} (M, K_{PRIV})
6. Return DissKeyUpdate

Pseudocode: Vehicle Receiving the new Dissemination Key
Input: DissKeyUpdate
Output: New Dissemination Key K_D
1. M = Dec_{sym} (PrivacyKeyUpdate, K_{PRIV})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$ (SPub, Pub_{CA}) = false, then return null and stop
6. Return new K_D, store it and erase the old one

3.9.4 Update CA's Public Key for Authorities

When CA's public and private keys are renewed, each authority will request CA by **AuthorityRequestCAKeyUpdate** message to get new new CA's Public Key. CA responds **CAKeyUpdateforAuthority** message which contain the new Pub_{CA}. Pseudocodes are shown in table 3.14. The new CA's public key is signed by the old CA's private key to ensure that the new key is generated by the CA. **CAKeyUpdateforAuthority** is concatenated from the (signed new CA's Public Key) with the Message Type (MT), timestamps (t), and encrypt all of them by the authority M' public key Pub_M, to ensure that only authority M can decrypt this message. When the authority M receives the **CAKeyUpdateforAuthority** message and decrypts it, the authority checks the validity of timestamps and verifies the signed new CA's public key by the old CA's public key. Finally, the authority extracts new CA's public key, stores it and erases the old one.

Table (3.14): Pseudocodes to send new CA's Public Key to Authority M

Pseudocode: Authorities Sending Request for New CA's Public Key
Input: AID_M
Output: AuthorityRequestCAKeyUpdate message
1.Get current timestamp t
2. Set Message Type (MT)=14
3. M = t || MT || AID_M
4. AuthorityRequestCAKeyUpdate = Enc_{pub} (M, Pub_{CA})
5.Return AuthorityRequestCAKeyUpdate

Pseudocode: CA Receiving the CA's Public Key Request from Authority M
Input: AuthorityRequestCAKeyUpdate
Output: call pseudocode for CAKeyUpdateforAuthority or null
1. M = Dec_{prv} (AuthorityRequestCAKeyUpdate, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. AID_M=false, then return null and stop
6. call pseudocode for CAKeyUpdateforAuthority

Pseudocode: CA Sending new CA's Public Key to Authority
Input: CA's New Public Key (Pub_{CA})
Output: CAKeyUpdateforAuthority
1.Get current timestamp t
2. Set Message Type (MT)=2
3. SPub =$Sign_{prv}$ (new Pub_{CA}, oldPrv_{CA})
4. M = t || MT || SPub
5. CAKeyUpdateforVehicle = Enc_{pub} (M, Pub_M)
6. Return CAKeyUpdateforAuthority

Pseudocode: Authority Receiving the new CA's Public Key
Input: CAKeyUpdateforAuthority
Output: CA's New Public Key Pub_{CA} or null
1. $M = Dec_{prv}$ (CAKeyUpdateforAuthority, Prv_M)
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$ (SPub, $oldPub_{CA}$) = false, then return null and stop
6. Return $newPub_{CA}$, store it and erase the old one

3.9.5 Update Secret Shared Key (K_{SM}) for Authorities

CA generates the secret shared key K_{SM} that will be used to exchange the information and messages between the authority M and the VSaaS modules such as CA. The key K_{SM}'s size is selected to be 128-bit, which is the common size for the symmetric ciphers AES. This key has a medium lifetime (a month for example).When K_{SM} is renewed, the authority M will request CA by **AuthorityRequestSharedKeyUpdate** message to get new Key (K_{SM}). CA responds **SharedKeyUpdateforAuthority** message which contain the new K_{SM}. Pseudocodes are shown in table 3.15. The new secret shared key (K_{SM}) is signed by the current CA's private key to ensure that the new key is generated by the CA. **SharedKeyUpdateforAuthority** is concatenated from the (signed new secret shared key (K_{SM})) with the Message Type (MT), timestamps (t), and encrypt all of them by the authority M' public key Pub_M, to ensure that only authority M can decrypt this message. When the authority M receives the **SharedKeyUpdateforAuthority** message and decrypts it, the authority checks the validity of timestamps and verifies the signed new secret shared key by the CA's public key. Finally, the authority extracts new new secret shared key, stores it and erases the old one.

Table (3.15): Pseudocodes to send new Secret Shared Key to Authority M

Pseudocode: Authorities Sending Request for New Secret Shared Key
Input: AID_M
Output: AuthorityRequestSharedKeyUpdate message
1.Get current timestamp t
2. Set Message Type (MT)=16
3. M = t || MT || AID_M
4. AuthorityRequestSharedKeyUpdate= Enc_{pub} (M, Pub_{CA})
5.Return AuthorityRequestSharedKeyUpdate

Pseudocode: CA Receiving the Secret Shared Key Request from Authority M
Input: AuthorityRequestSharedKeyUpdate
Output: call pseducode for SharedKeyUpdateforAuthority

or null
1. M = Dec_{prv} (AuthorityRequestSharedKeyUpdate, Prv_{CA})
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. AID_M=false, then return null and stop
6. call pseudocode for SharedKeyUpdateforAuthority

Pseudocode: CA Sending new Secret Shared Key to Authority

Input:Secret Shared Key (K_{SM})
Output: SharedKeyUpdateforAuthority message
1.Get current timestamp t
2. Set Message Type (MT)=17
3. SPub =$Sign_{prv}$ (new K_{SM}, Prv_{CA})
4. M = t || MT || SPub
5. CAKeyUpdateforVehicle = Enc_{pub} (M, Pub_M)
6. Return SharedKeyUpdateforAuthority

Pseudocode: Authority Receiving the new Secret Shared Key

Input: SharedKeyUpdateforAuthorit
Output: New Secret Shared Key (K_{SM}) or null
1. M = Dec_{prv} (SharedKeyUpdateforAuthority, Prv_M)
2. Extract MT from M
3. Extract timestamp t from M
4. If t is invalid, then return null and stop
5. $VerifySign_{pub}$ (SPub, Pub_{CA}) = false, then return null and stop
6. Return new K_{SM}, store it and erase the old one

Chapter 4
Simulation Works

Chapter4
Simulation Works

Testing new protocols, scenarios and wireless technology schemes is complex, high expensive and cannot be accomplished in large testbed, in addition, the testing new technologies for transportation in the real world is very dangerous. Simulation plays an important role to find out the beneficial and effective technologies before implementation. This chapter will introduce the simulation of VANET and presents different required tools which used in our simulation's implementation.

4.1 Introduction

For the VANET simulation, the software developers and the researchers together developed several programs in order to allow the studies and evaluations of numerous application and protocols in VANET. The important features of VANETs includes vehicles, which can move very fast. The considered network is highly dynamic which means that the topology of the network is continuously changing, while the position and density of the nodes is changing. VANET simulation requires two types of simulation components: Network and Mobility. In most cases the network and mobility simulator are separated. There are several simulators available that can be used for VANETs simulation. This study has classified existing VANET simulation software into three different categories: (a) Network simulators, (b) Traffic/Mobility simulators, (c) Software to integrate between (a) and (b) or software which can simulate both mobility and network (VANETs simulator). (Figure 4.1) represents the classification of VANET simulators (Yan, Ibrahim and Weigle, 2009).

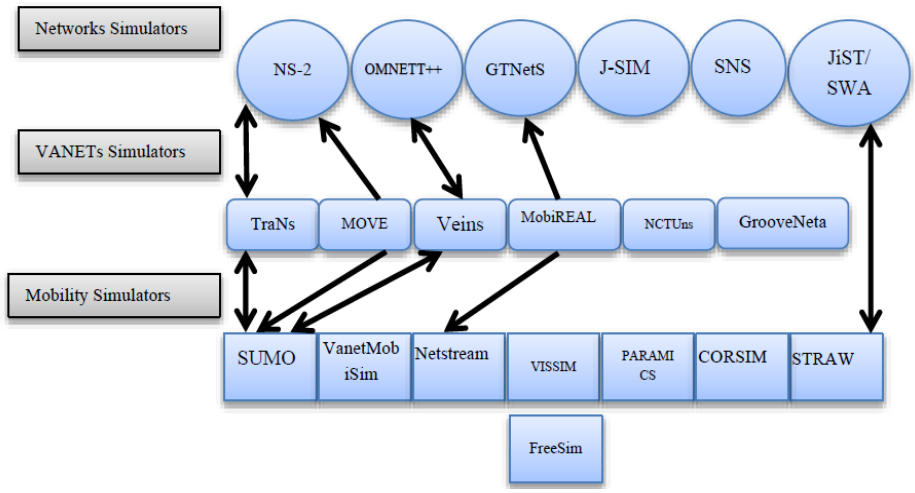


Figure (4.1): Classfication of VANET Simulators

4.1.1 VANET Mobility Generators (Traffic Simulation)

Creating a realistic mobility model for the simulation of VANETs is important. Vehicular traffic typically moves in relatively predictable ways along a set path. These movements are governed by how the road network is laid out. The placement of lanes and traffic features, e.g. traffic signs, turning lanes or traffic lights combined with both a source, destination and other vehicles decides how a vehicle will move in the real world. To get accurate results for how VANET technologies will work it is important to model these movements with a high degree of accuracy.

Initial work on Mobile Ad Hoc Networks (MANETs) often used random node movements. In essence nodes would choose random directions to move in and periodically change direction. This practice was initially carried over into VANET research. Of course it is nothing like vehicle traffic in the real world. Studies have shown that random node movements are a poor substitute for a mobility model and should not be used (Yoon, Liu and Noble, 2003).

A second approach that was taken for a mobility model was the use of real world mobility traces. Obtained by tracking the location of real world vehicles using GPS or other technologies they mimic the real world exactly. Nodes within the simulation are then moved according to these traces exactly. While they do an excellent job of simulating mobility as it occurs in the real world they are of limited flexibility. Changing parameters, such as traffic density, is not feasible for large scale simulation. A better approach is the use of a dedicated traffic simulator. There are a wealth of traffic simulators available. So, in order to have realistic and acceptable simulation of VANET, mobility generator is required. Examples are SUMO, Netstream, VISSIM and STRAW (Yoon, et al., 2003).

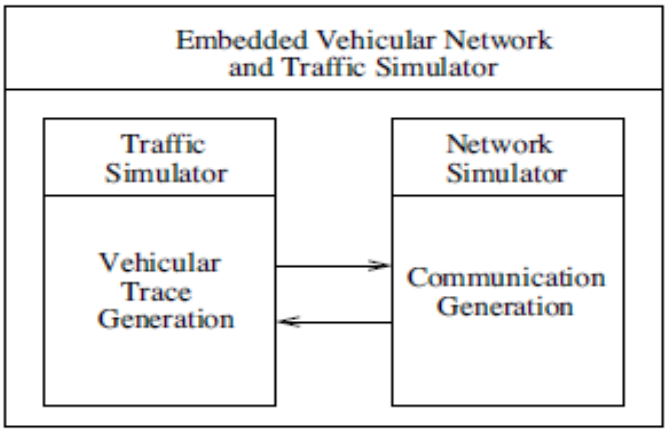
4.1.2 Network Simulation

Study on behavior of networks under several different condition are possible using the network simulators. Researchers are able to adjust the simulation to get results with required specification. The advance network simulator are relatively cheap and fast compare to time and cost which are required for set an test bed includes several computer in network, data links and routers. Therefore the network simulator helps user to simulate several scenario which are have difficulty to implement or has a high cost in real world especially for VANET. Network simulation is very useful in order to tests novel network standards or for proposing the novel modification of the existing protocols in a very reproducible and well-ordered manner. Examples are NS-2, NS-3, OMNET++, JiST/SWANS, and GTNetS.

4.1.3 VANET Simulation

As explained previously, in order to simulate a VANET application, two different simulations are required, Mobility simulator and network simulator. Up to now, these two issues in VANET simulation are decoupled. However, the problems for VANET simulation is that how to integrate these two simulators. A simple solution to aim this goals is that to perform the movement model in the network simulation. This type of simulation is called one way communicating, which is just only have the network and mobility simulation separately and the network simulator cannot affect the traffic and mobility simulator. So, the communications have no effects in the vehicles moving.

In the other side, VANET simulators are providing two-way communication (see Figure 4.2), and mostly include two simulators (mobility and network) that could make a connection between mobility and network simulator. These type of simulation are more useful for traffic information those are have assumption of that feedbacks from the networks simulator should have effect on the cars mobility. For this kind of simulation, at first traffics are generated in traffic simulator and then the traffic are feeds into network simulation, and simulation are going to run. Network simulation can have effect on mobility of cars after simulation started.This kind of simulation are knows as VANET simulator or an integrated framework.



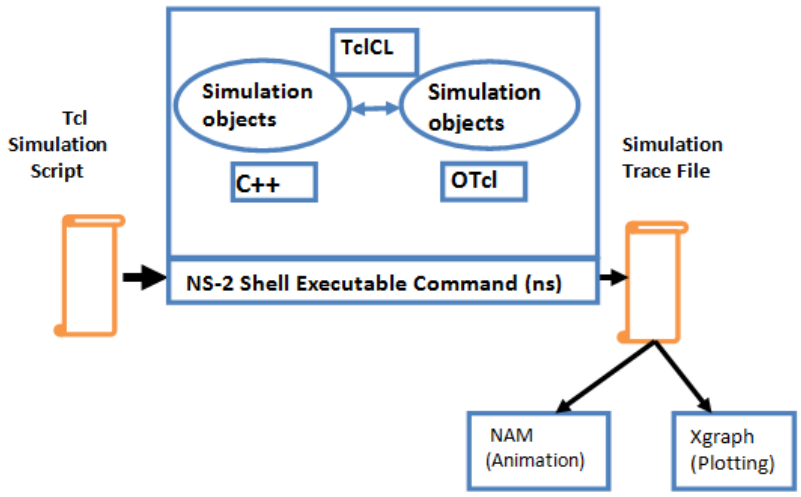
Figure(4.2): Connection between Network and Traffic Simulators

4.2 Simulators

NS-2 ver. 2.35 network simulator, Sumo ver. 0.10 is traffic simulator and Trans ver. 1.2 VANET simulator are chosen here in this thesis to test Vehcile Information Messages VIMs performance in VSaaS. All of these working under Linux Centos 6.6 and the kernal version is 2.6.32. Brief descriptions of these simulators are provided in the following sub sections.

4.2.1 NS-2 (Network) Simulator

The NS-2 (VINT Project Website, 2011) simulator was originally created as part of the Defence Advanced Research Projects Agency (DARPA) sponsored Virtual Inter-Network Testbed (VINT) project at the University of California. It has since been extended and improved with a large community of users and developers. The core kernel is written in C++ but utilises a number of Tcl scripts for the particulars of wired and wireless networks (including some details of satellite and older technologies). The NS-2 simulation scenario scripts are written in TcL (See Figure 4.3). The use of TcL does simplify the creation of scenario scripts, and with deeper knowledge of the simulation system, direct C++ programs can be written.



Figure(4.3): The Component of NS-2

Cuurenly, NS-2 is one of the best environments that have been developed to simulate the real network for the wire and wireless networks. It is open source event-driven simulatorbased in object oriented objects and emerged to support the area of research and computer communication network. NS-2 includes modules for supporting various types of network components such as multicast routing protocols, transport layer protocols, and application. However, it is available on several platforms such as FreeBSD, Linux, SunOS and Solaris. NS-2 also builds and runs under Windows with Cygwin. Simple scenarios should run on any reasonable machine; however, very large scenarios benefit from large amounts of memory and fast CPU's.

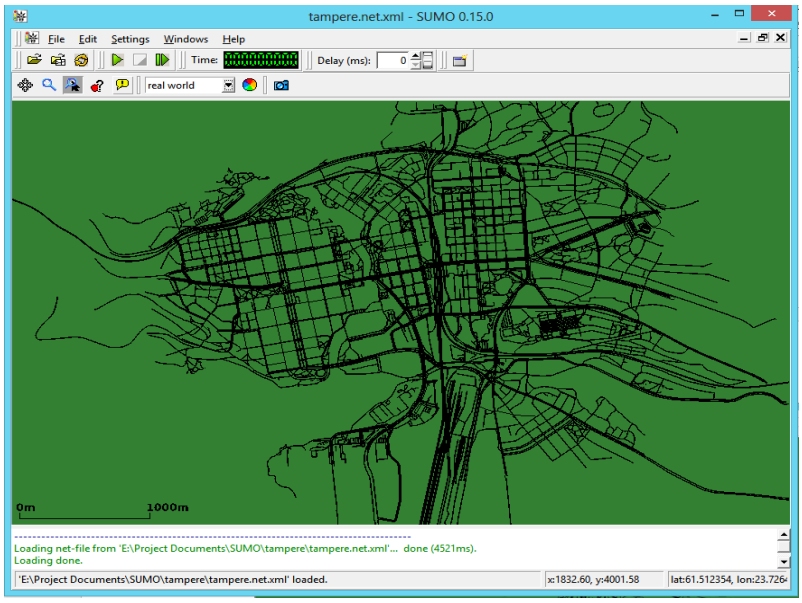
4.2.2 SUMO (Traffic/Mobility) Simulator

SUMO or Simulation of Urban Mobility (Behrisch, Bieker, Erdmann and Krajzewicz, 2011) is a very powerful traffic simulator which is open source as well. SUMO has been used in a wide variety of VANET projects (Sommer, German and Dressler, 2011). The road network, vehicle types and vehicle routes are all highly configurable and allow for customized simulations. Furthermore, Traffic Control Interface (TraCI) allows SUMO to communicate bi-directionally with any network simulator implementing TraCI. This allows the results of traffic simulator to affect the network simulator and vice versa (Krau, 1997).

SUMO road networks are defined by a network file. In the network file lanes are defined as edges in a directed graph with vertices taking the form of connections between lanes. Individual lanes have attributes such as speed limits or turning restrictions. Connections between lanes can simply indicate a change in direction or can be complex multilane intersections with traffic lights or priority traffic direction. While quite complex there is a suit of included tools for generating SUMO road networks. Simple geometric road networks can be generated using the NETGEN utility. To model real life road networks map data from a variety of sources can be imported using the NETCONVERT utility. One such source is the Open Street Map (OSM) project. It provides a Google Maps like interface to viewing community generated map data (See Figure 4.4). It is also possible to download the underlying map data to convert using NETCONVERT. These tools help to provide a way to generate realistic road networks. Vehicle traffic is defined by a route file. Again there is a suite of tools to generate routes.

4.2.3 Trans (VANET/Integrator) Simulator

TraNS (Traffic and Network Simulation Environment) is a GUI tool that integrates traffic and network simulators (SUMO and ns2) to generate realistic simulations of Vehicular Ad hoc NETworks (VANETs). TraNS allows the information exchanged in a VANET to influence the vehicle behavior in the mobility model. For example, when a vehicle broadcasts information reporting an accident, some of the neighboring vehicles may slow down (TraNS Offical Website, 2012).



Figure(4.4): Example of a MAP in the SUMO Simulator

4.3 Crypotographic Algorithms

Our proposed framework needs to use some cryptographic algorithms. They are:

1. Symmetric-key cryptographic algorithm
2. Public-key cryptographic algorithm.

4.3.1 Symmetric-key Cryptographic Algorithm

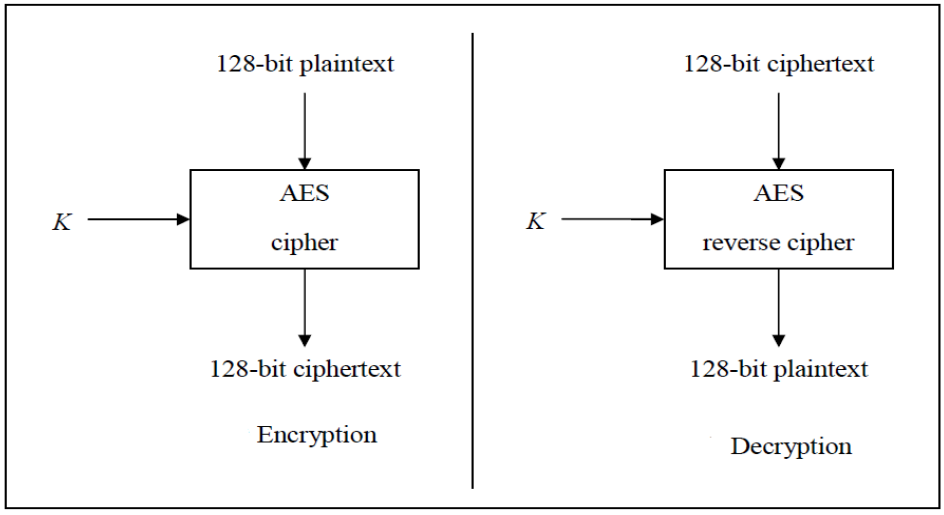
This proposed framework uses 128-bit keys for symmetric-key cryptographic operations. It needs to encrypt blocks of 128-bit length. We choose AES-128 because it is standard, easy to implement and popular for this purpose. In symmetric-key cryptographic, the same key used for encryption and decryption operations (Advanced Encryption Standard Website, 2001).

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 which stands for Federal Information Processing Standard. 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 4.5) (Advanced Encryption Standard Website, 2001).

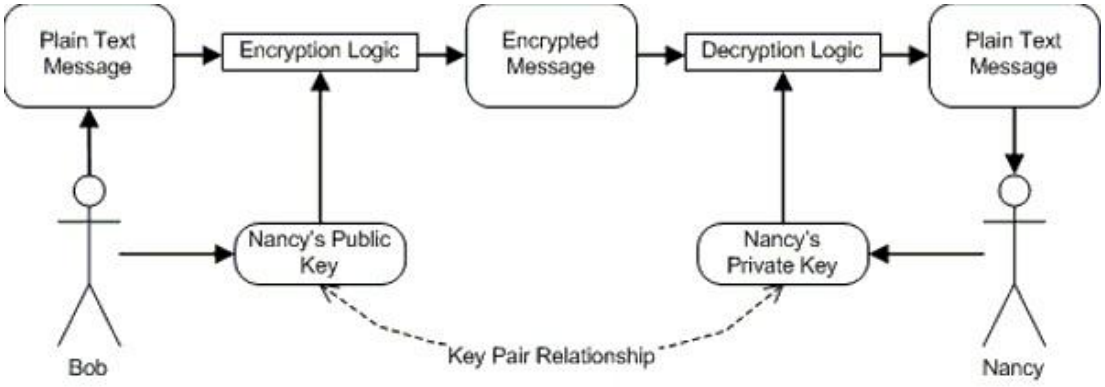
4.3.2 Public-key Cryptographic Algorithm

Public-key cryptography is called also asymmetric-key cryptography. Public-key cryptography uses two separated keys: private key and public key. Private Key is kept secret

whereas public key is published. For encryption, the public key used for encryption and private key used for decryption, and vice versa for signing process. The most common public-key cryptographic 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 (See Figure 4.6) (Kleinjung et al., 2010). We choose here RSA-2048 bit as a public-key cryptographic algorithm because it is very secure, standard, easy to implement and popular for this purpose.



Figure(4.5): AES Encryption/Decryption



Figure(4.6): RSA Encryption/Decryption

4.4 The Performance Analysis of the Secure Vehcile Information Messages (VIMs) in our Proposed VSaaS

This section evaluates and analyzes the performance of the secure Vehicle Information Messages (VIMs), which are sent by the vehicles to the Certified Authority (CA) then to the storage where these components are hosted on the cloud. For a secure communication in the VANET, the security requirements should be satisfied. We need to ensure that our proposed messages "Vehicle Information Messages (VIMs)" are effective and reliable. As a part of the security requirements, it is essential to meet certain performance requirements, which guarantees, that the VANET will probably work its function without any fail. Thus, the impact of the security model or protocols will be analyzed. In our work, the main security service is the (CA), which is responsible for the cryptographic and the authentication of the secure Vehicle Information Messages (VIMs), which are sent by the vehicles. The existing of the CA reveals two additional factors, that should be taken into consideration. They are: the security overhead in the message size and the time taken for the encryption/decryption operations. As a result (Raya and Hubaux, 2005):-

Secure VIM size = Standard VANET Safety Message Size + Security Overhead Size (4.1)

Time Overhead = Encryption Time + Transmitting Time (delay) + Decryption Time (4.2)

4.4.1 Performance Matrices

This work investigates the throughput, end to end delay and the message delivery rate as in (Khairnar and Kotecha, 2013; Khasa and King, 2016; Rohal, P., Dahiya and Dahiya, 2013) to evaluate the performance of our security model (VSaaS) against the Vehicle Information Messages (VIMs), and answer the important question: Is the public key cryptography (CA service) fit?

1. Throughput

Throughput is the number of the packets passing through the network during a certain time. It counts the total number of packets that have been successfully delivered to the desired node. The throughput increases as the node density increases. It is measured in bits per second (bit/s or bps). Throughput can be represented mathematically as in the equation below:

$$\text{Throughput} = \frac{no.of\ \ delivered\ packet * packet\ size * 8}{total\ simulation\ time} \quad (4.3)$$

2. End-to-End Delay

End-to-end delay is defined as the time taken for a packet to be transmitted across a network from the source to the destination. It is calculated by taking the average time for the data packet that arrive to the destination. It also includes the delay caused by the route discovery process and the queue in the data packet transmission. Only the data packets that are successfully delivered to the destination are counted. Furthermore, if the value of the delay is low, it means that the performance of the protocol is better. It is measured in second. The following equation is used to calculate the average end-to-end delay,

$$T_{E2E} = \frac{\Sigma(T\_R - T\_S)}{n} \quad (4.4)$$

T_{E2E} is the average End-to-End Delay, T_R is the time of received packets at the destination node, T_S is the time of sent packets from the source node, and n is the number of packets.

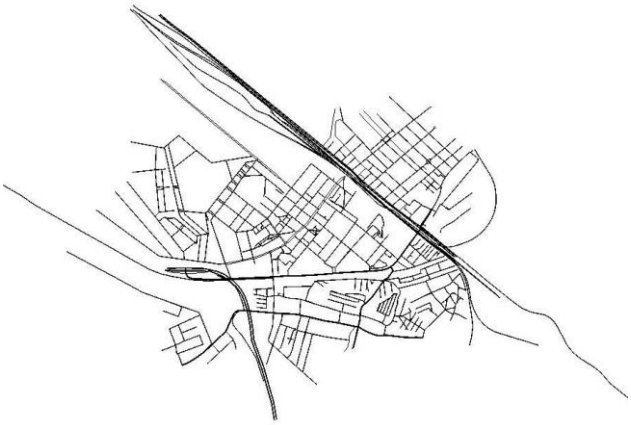
3. Message Delivery Rate

Message delivery rate is the sum of the successful received messages by all the nodes in the network per second. It is measured in messages per second. The following equation is used to calculate the message delivery rate,

$$\text{Message Delivery Rate} = \frac{no.of\ delivered\ packet}{total\ simulation\ time} \quad (4.5)$$

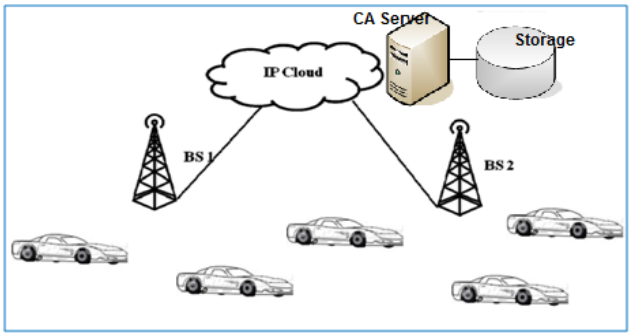
4.5 Simulation Setup

Our simulation work considers the vehicles moving in a part of Cologne city, which has a region size of 12594m x 6208m (See Figure 4.7). This area has been covered by appropriate number of gateways, that linked the vehicles to the cloud, where the CA and the storage are hosted. The simulation time has been set to 300 seconds. The Maximum Transmission Unit (MTU) has been set to 1500 bytes. We take into consideration the cloud delay, which is approximately 30 milliseconds as mentioned in (Al Mamun et al., 2012). And, the cloud backbone bandwidth has been set to 100 Mbps. The mobility model of the vehicles includes the speed, accelerator and the positions, which are retrieved from the map using the SUMO and the Trans simulators. Moreover, we configured ns2 to support the roaming among the gateways.



Figure(4.7): A Part of Cologne City Map

The aim of this simulation work is to evaluate the performance of our securitymodel framework (VSaaS) against the secure Vehicle Information Messages (VIMs), which are sent by the vehicles to the CA, then to the storage where the CA and the storage are hosted in the cloud, (See Figure 4.8). Then, answer the important question: Is the public key cryptography (CA service) fit? To evaluate this security overhead, we investigate the performance metrics measurements.



Figure(4.8): Sample Figure of the Simulated Topology

Because of the existing of the CA, we should take into consideration its overhead factors, which are: the security overhead in the message size, and the time taken for the encryption/decryption operations.

4.5.1 The Size Overhead

We set the normal size of the Vehicle Information Messages (VIMs) to 200 bytes including the header, timestamp, message type (MT) value and etc, and according to the standard, the typical size of the safety messages in the VANET is between 100 and 200 bytes without the security size overhead as mentioned in (Xu, Mak, Ko and Sengupta, 2004; Yang, Liu, Zhao and Vaidya, 2004). Where the security overhead in the message size of the secure VIMs is resulted because of the encryption operation, which has been done by using the CA's public

key (we choose RSA-2048bit which expands the normal message by 56 byte). So, the size of our proposed secure message (VIM) becomes 256 bytes as described in the equation (4.1).

4.5.2 Benchmarks

The simulation of our proposed protocol needs to use a speed (time) benchmark for the selected cryptographic algorithms. In (Cryptography Benchmarks Website, 2009), 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. The 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. The 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. The operating system is Linux.

Table 4.1 shows the time needed by RSA-2048 for the encryption and decryption operations on the selected machines.

Table (4.1): 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

The Pentium 4 benchmark result was choosen for encryption operation in the vehicle side because of the CPUs installed on the vehicles have lower performance than those used in desktop computers. And, the AMD benchmark result was choosen for decryption operation in the CA side because of the CPUs installed on the servers have higher performance than those used in desktop computers. Thus as a result, the total time used to encrypt and decrypt every secure VIM is calculated as 0.22 (encryption timein the vehicle side) + 2.90 (decryption timein the CA side) = 3.12 Milliseconds.

4.5.3 Simulation Scenarios

Scenario 1, the Simulation is executed for different number of vehicles: 25, 50, 75, 100, 125 and 150 with a normal message size of 200 bytes (without security), where the message rate is

0.3 second. Moreover, the simulation is executed again for different number of vehicles: 25, 50, 75, 100, 125 and 150 with security overhead (CA effects) where the message size becomes 256 bytes and take into consideration the encryption/decryption time overhead, and the message rate is also 0.3 second.

Scenario 2, the Simulation is executed for fixed number of vehicles which is 50 vehicles with a normal message size of 200 bytes (without security) where the message rate is varied: 0.1, 0.2. 0.3, 0.4, 0.5 and 0.6 second. Moreover, the simulation is executed again for fixed number of vehicles which is 50 vehicles with security overhead (CA effects), where the message size becomes 256 bytes, and take into consideration the encryption/decryption time overhead, also the message rate is varied: 0.1, 0.2. 0.3, 0.4, 0.5 and 0.6 second.

In the next chapter, we will present our results together with critical discussion.

Chapter 5
Results and Discussion

Chapter 5
Results and Discussion

5.1 Simulation Results

5.1.1 Throughput Computational Cost

Throughput is the main measurement in the performance matrices. We need to make some computational works in order to inform us if the security overhead is acceptable or not before the starting with the simulation implementation. We propose the using of the CA (Public key cryptographic) to support the security in the VANET, it is important to accept its overhead in the vehicular context. Theoretically, according to the numerical upper bounds, the throughput can be calculated by using the following equation [66]:

$$\text{Throughput } (kbps) = \frac{N \times R \times M \times 8}{1024} \quad (5.1)$$

N is the number of vehicles, R is the messaging rate (message per second per vehicle) and M is the total message size (bytes).

Table 5.1 gives us the theoretical calculated throughput values from equation 5.1 for the secure VIM, when its size is 256 bytes, the message rate is 0.3 second and the number of vehicles is varied 25, 50, 75, 100, 125 and 150.

Table (5.1): No. of vehicles vs. throughput for secure VIM

No. of vehicles	**25**	**50**	**75**	**100**	**125**	**150**
Throughput (Kbps)	162.5	325	487.5	650	812.5	975

And, Table 5.2 gives us the theoretical calculated throughput values from equation 5.1 for the secure VIM when its size is 256 bytes, the number of vehicles is 50 vehicles and the message rate is varied 0.1,0.2, 0.3, 0.4, 0.5 and 0.6 second.

Table (5.2): Message rate vs. throughput for secure VIM

Message Rate	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**
Throughput (Kbps)	977	488	325	244	195	163

5.1.2 Simulation Results: Scenario 1

1. Throughput

Figure 5.1 shows the system throughput of the normal messages and the secure messages sent by the vehicles. Normally, the throughput increases linearly with the increase in the number of the vehicles, because the increasing in vehicles' number increases the number of the sent packets, which is resulted in the increasing number of the delivered packets. The delivered packets is the main factor in the throughput equation (4.3).

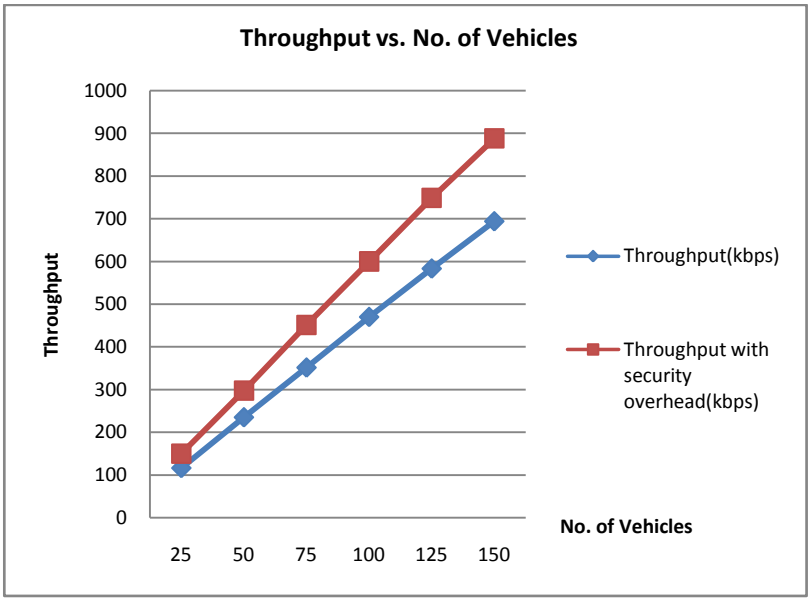


Figure (5.1): Throughput vs. No. of Vehicles for both normal and secure messages

Also, the effect of the CA in the throughput is shown in the (Figure 5.1), the throughput of the secure messages is more than the throughput of the normal messages, according to the security overhead in the message size that increases the throughput. But, this effect is acceptable because the infrastructure's throughput capacity can afford this overhead, and as shown in (Figure 5.1), the throughput did not exceed 1 Mbps, even when the 150 vehicles sent secure messages to the CA at the same time. It is worth to mention that, the actual throughput in the scenario with security model is better than without security model because the system has the abiity to resist and drop the malicious messages and the messages that generated from the untrusted vehicles.

Finally, the throughput values of the simulation results are agreed with the computational works in table 5.1, because all the throughput values which got from the simulation are below the numerical upper bounds.

2. End-to-end Delay

(Figure 5.2) shows the end-to-end delay of the normal and secure messages sent by the vehicles. That delay is not be considered when the number of the vehicles increases, because of the low contention on the medium.

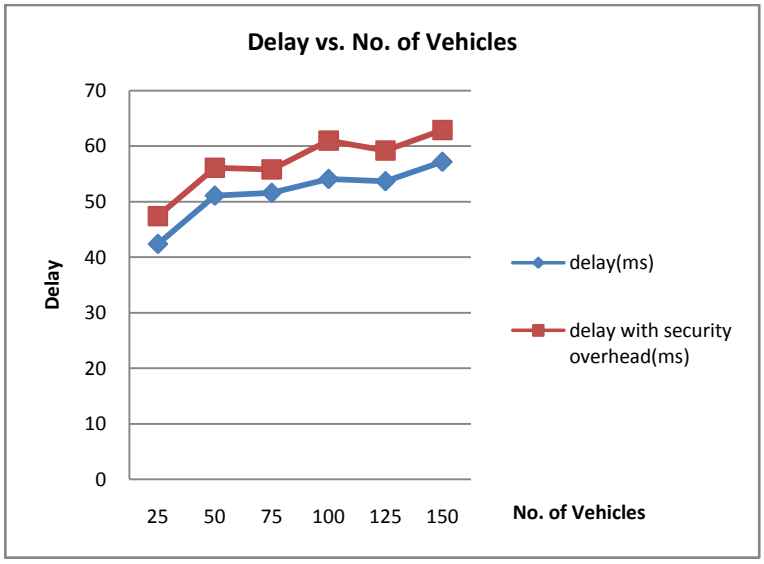


Figure (5.2): Delay vs. No. of Vehicles for both normal and secure messages

Also, there is no considerable effects of the CA in the delay as shown in the (Figure 5.2). This is because the infrastructure can afford the security overhead in the message size and the cryptographic operations time overhead, according to the low contention on the medium and the high transmission rate that minimizes the effects of security overheads. Thus, the CA and the cryptographic operations do not critically affect the delay.

In addition to, the delay values are between 42 ms and 63 ms; which are acceptable and good results in the cloud environment.

3. Message Delivery Rate

(Figure 5.3) shows the message delivery rate of the normal messages and the secure messages sent by vehicles. Normally, the message delivery rate increases linearly as the number of vehicles increases, because the increase in the vehicles' number increases the number of the sent packets, which is resulted in the increasing number of the delivered packets. The delivered packets is the main factor in the message delivery rate equation (4.5).

Also, there is no considerable effects of the CA in the message delivery rate as shown in the (Figure 5.3). This is because the infrastructure can afford the security overhead according to

the low contention on the medium and the high transmission rate that minimizes the effects of the security overhead in the message size. Thus, the CA and the cryptographic operations do not critically affect the message delivery rate.

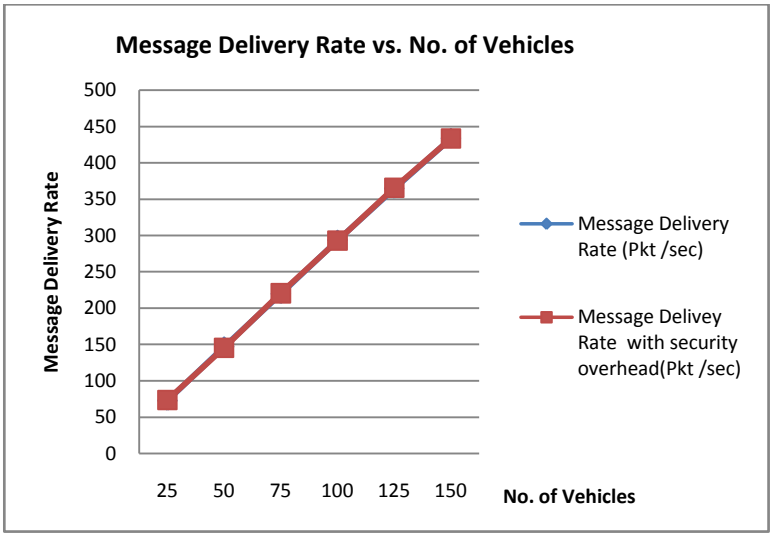


Figure (5.3): Message Delivery Rate vs. No. of Vehicles for both normal and secure messages

5.1.3 Simulation Results: Scenario 2

1. Throughput

(Figure 5.4) shows the system throughput of the normal messages and the secure messages sent by the vehicles. Normally, the throughput decreases as the message rate value increases, because the increasing in the message rate value means decreasing in the number of the sent packet per second, which is resulted in the decreasing number of the delivered packets. The delivered packets is the main factor in the throughput equation (4.3).

Also, the effect of the CA in the throughput is shown in the (Figure 5.4), the throughput of the secure messages is more than the throughput of the normal messages, according to the security overhead which increases the message size that increases the throughput. But, this effect is acceptable because the infrastructure's throughput capacity can afford this overhead. And as shown in (Figure 5.4), the throughput does not exceed 1 Mbps, even when the message rate of the secure message is set to maximum (10 messages/vehicle/second). It is worth to mention that, the actual throughput in the scenario with security model is better than without security model because the system has the abiity to resist and drop the malicious messages and the messages that generated from the untrusted vehicles.

Finally, the throughput values of the simulation results are agreed with the computational works in table 5.2, because all the throughput values which got from the simulation are below the numerical upper bounds.

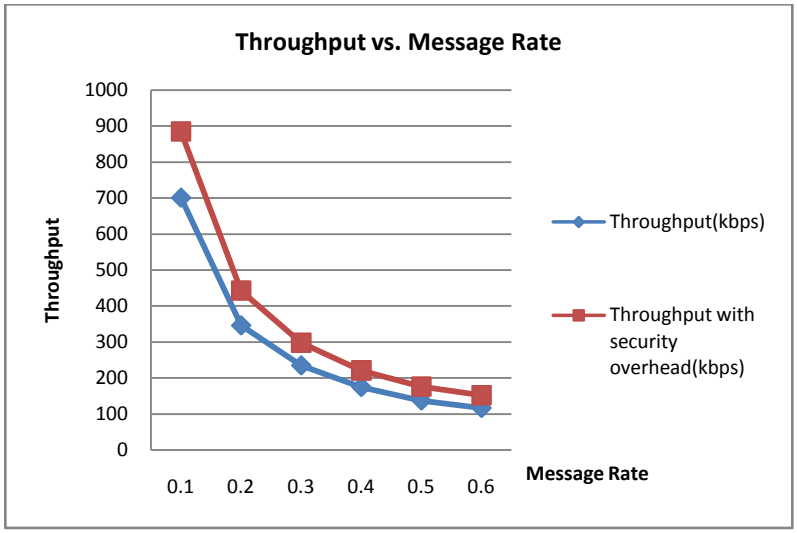


Figure (5.4): Throughput vs. Message Rate for both normal and secure messages

2. End-to-end Delay

(Figure 5.5) shows the end-to-end delay of the normal messages and the secure messages sent by vehicles. That delay is not be considered when the message rate varied from 0.1 to 0.6 seconds, because the infrastructure can afford this variation for both normal and secure messages according to the low contention on the medium and the high transmission rate that minimizes the effects of the variation in the message rate.

Also, there is no considerable effects of the CA in the delay as shown in the (Figure 5.5). This is because the infrastructure can afford the security overhead in the message size and the cryptographic operations time overhead according to the low contention on the medium and the high transmission rate, that minimizes the effects of these security overheads. Thus, the CA and the cryptographic operations do not critically affect the delay.

In addition to, the delay values are between 49 ms and 57 ms, which are acceptable and good results in the cloud environment.

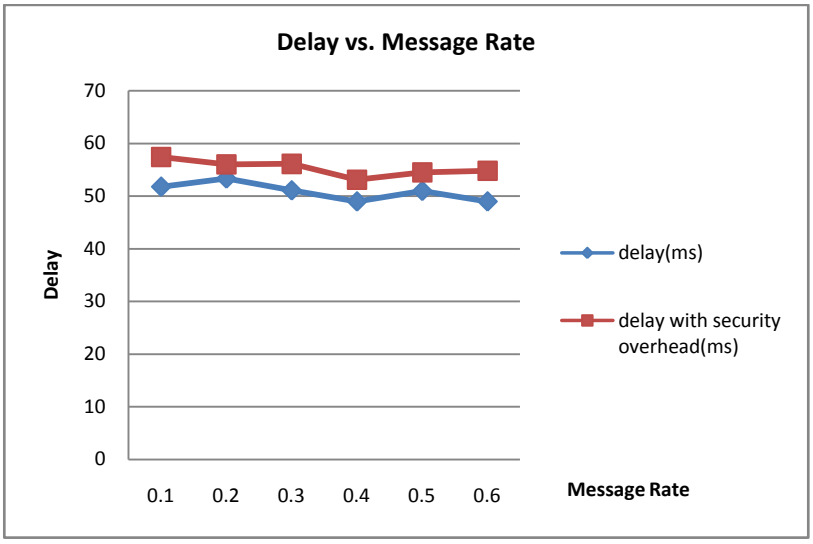


Figure (5.5): Delay vs. Message Rate for both normal and secure messages

3. Message Delivery Rate

(Figure 5.6) shows the message delivery rate of the normal messages and the secure messages sent by the vehicles. Normally, the message delivery rate decreases as the message rate value increases, because the increase in the message rate value means a decrease in the number of the sent packet, which is resulted in the decreasing number of the delivered packets. The delivered packets is the main factor in the message delivery rate equation (4.5).

Also, there is no considerable effects of the CA in the message delivery rate as shown in the (Figure 5.6). This is because the infrastructure can afford the security overhead according to the low contention on the medium and the high transmission rate that minimizes the effects of the security overhead in the message size. Thus, the CA and the cryptographic operations do not critically affect the message delivery rate.

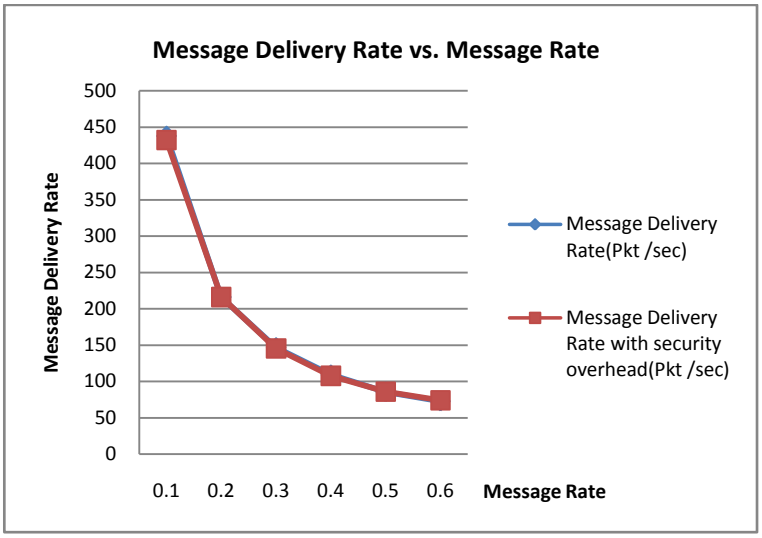


Figure (5.6): Message Delivery Rate vs. Message Rate for both normal and secure messages

5.1.4 Results Discussion

The security overhead (CA effects) of the secure Vehicle Information Messages (VIMs), in our proposed model which is based on the cloud, is acceptable. The impact of security overhead appears in the throughput because of the security overhead in the message size has an increasing in the throughput, but this effect is acceptable because of the infrastructure's throughput capacity can afford this overhead and the throughput is not exceeds 1 Mbps. It is worth to mention that, the actual throughput in the scenario with security model is better than without security model because the system has the abiity to resist and drop the malicious messages and the messages that generated from the untrusted vehicles.

In our proposed model ,there is no considerable effects of the CA in the delay and message delivery rate because of the low contention on the medium and the high transmission rate that minimizes the effects of security overheads.

Also, the throughput and message delivery rate are increased as the number of vehicles increased and as the message rate increased according to increased number of delivered packets.

Finally, the delay is not be considered when the number of vehicles increased or the message rate increased because of the low contention on the medium and the high transmission rate that minimizes the effect of the increasing in sent packet against the delay.

5.2 Security Analysis

Firstly, our proposed model framework (VSaaS) will be discussed against the security requirements in the VANET. After that, some related security issues also will be discussed.

5.2.1 VSaaSAgainst Security Requirements in VANET

1. **Identification and Authentication:** the CA, which is a part of the VSaaS model, generates an identifier to every vehicle, which is called Vehicle Identification Number (VID), before giving a license to the work and registers this VID with CA itself. Thus, it should be understood that, the CA can identity and verify the vehicle by its VID to determine if it is a legitimate vehicle or not, where this VID should be added to the vehicles' messages in a secure way. This Identification prevents the intruders from sending false messages. It is not possible to track the VID of the vehicle only through the authorities that have a traceable permission. Also, CA generates an identifier to every authority, which is called Authority Identification Number (AID). Thus, it should be understood that, the CA can identity and verify an authority by its AID, to determine if it is a legitimate authority or not. This Identification prevents the intruders from cooperating with the VSaaS model.

2. **Privacy and Anonymity:** For liability, vehicles' identities (VIDs) should be added to the vehicles' messages, but this requirement contradicts with the privacy. Therefore, vehicles' identities should be hidden (encrypted) from the others, only the CA can identify the vehicles' identities. To solve it, the CA generates a symmetric key which is called the privacy key K_{PRIV}, it is used to encrypt/decrypt the vehicles' identities (VIDs). The VID is concatenating with the current reading (xy-coordinates) which is taken from the tamper GPS, then encrypting the all with the privacy key K_{PRIV} to produce the EVID, which is added to each message as an alternative of the clear VID. It is worth to mention that, the privacy key K_{PRIV} provides authentication and privacy. Authentication is achieved because only the registered and trusted vehicles have this privacy key K_{PRIV}, where it is used to encrypt/decrypt the vehicles' identities (VIDs). Using the same privacy key K_{PRIV} by all the vehicles at the same time, provides anonymity which achieves the privacy. And, the concatenating xy-coordinates to the VID every time before encryption, ensures that the EVID value is different for every message, and mitigates the linking between the two messages generated from the same vehicle. Also, the EVID is a part of the vehicles' messages, where the whole message is encrypted by the CA's public key. Thus, only CA can decrypt the whole message by the CA's private key to get the EVID.

3. **Confidentiality:** all the messages sent by the vehicles and authorities are encrypted by the CA's public key. Thus, only CA can decrypt the messages by the CA's private key. In addition, the CA generates the secret shared key K_{SM} that will be used to exchange the information and messages between the authority and the VSaaS modules. This keeps the content of messages secret.

4. **Authorization:** the VSaaS provides the authorization through proposing a security access list (ACL), to manage the permissions. The ACL represents a set of permissions and rules to Allow/deny the inter-actions between the different entities (vehicles, authorities, VSaaS modules) and the intra-actions between the modules within the VSaaS. Our design of VSaaS is modular. It is easy to add new types of authorities, databases and VSaaS's modules by defining their permissions.

5. **Availability:** it is essential for the part of security availability to meet certain performance requirements, which guarantees the VANET will work its function probably without any fail. This work simulated and evaluated the secure Vehicle Information Messages (VIMs) with the security overhead (CA effects).The performance of the secure Vehicle Information Messages (VIMs) is acceptable. The impact of the security overhead appears in the throughput because the security overhead in the message size has an increasing in the throughput, but this effect is acceptable because the infrastructure's throughput capacity can afford this overhead. There is no considerable effects of the CA in the message delivery rate and end-to-end delay because the low contention on the medium and the high transmission rate, minimize the effects of the security overheads. But, the availability of system like that, is something that cannot be fully guaranteed. The primary vulnerability, which lies in the different types of wireless technologies, is considered as jamming attacks. Also, the DoS attacks can be realized by sending too many messages to the specific destination, therefore, there won't be enough time to process the valid messages. Detection and prevention of the DoS also require mechanisms, hardware and software to satisfy the concept of the intrusion detection and prevention.

6. **Non-Reputation:** Non-Repudiation is achieved in our work because of the following reasons:
    - The VSaaS is resistant against the masquerade attack.
    - Vehicles cannot cheat about their positions and related parameters because a secure positioning solution is used in the messages.

- The vehicle cannot deny having a sent message, because it includes the vehicle's identity concatenated to its real xy coordination, and encrypts the all by the privacy key K_{PRIV}.
- The vehicle cannot claim that the message was replayed because the timestamp is included in each message.

7. **Entity Revocation:** the VSaaS provides mechanisms to revoke the vehicles and authorities when they are engaged in a malicious activity. But, the methodology to determine a malicious activity is out of our scope work.

5.2.2 More in Security

Messages are provided by the timestamps to guarantee the message freshness and provide protection against the reply attacks. Only the authorities, that have a traceable permission, can track a vehicle through its VID. It is worth to mention, the VID is chosen to be a value of 64 bit length. This length ensures that there are 18 billion attempts to guess the VID when the brute force attack presented.

All mentioned keys in the VSaaS framework model are changed frequently in a way to keep the content of messages secret, and prevent any attempts to uncover these keys. Moreover, the VSaaS provides mechanisms to change the keys if any compromising happens.

It is not possible to send a false location, because the algorithm of the sending secure VIM reads the (xy-coordinates) from the tamper GPS, which is build-in on the vehicles, and concatenates it to the VID in order to produce the EVID that is a part of the messages sent by the vehicles. Moreover, each vehicle has a tamper-proof device (TPD) installed by the manufacturer, to store all the secret information used in the VANET. It is fabricated such as no one can reveal or compromise its information. TPD should erase all the secret information if it is removed from the vehicle. This is providing a physical security to the TPD.

The integrity mechanisms do not mentioned to in our work because of the encrypting of the whole message was proposed. Thus, it is meaningless to take into consideration any integrity mechanisms with encrypting of the whole message. To send secure VIMs, a security level was assumed to be equivalent at least to RSA 2048, which is supposed to survive until 2030.

Chapter 6
Conclusion and Future Works

Chapter6
Conclusion and Future Works

6.1 Conclusion

This thesis highlighted a number of previous related works which proposed the VANET security, merging the Cloud Computing with the VANET and the VANET-cloud security. Also, it proposed VANET based on the cloud (V2Cloud) and the design of a security model framework that is hosted on the cloud to manage the security services, and provide a secure VANET communication between the different entities e.g. vehicles and authorities. This security model framework is called VANET Security as a Service (VSaaS).

The throughput, end-to-end delay and the message delivery rate was investigated through the NS2, SUMO and the Trans simulations, to evaluate the security overhead of the secure Vehicle Information Messages (VIMs). The impact of the security overhead appeared on the throughput because the security overhead in the message size has an increasing in the throughput, but this effect is acceptable because the infrastructure's throughput capacity can afford this overhead. It is worth to mention that, the actual throughput in the scenario with security model is better than without security model because the system has the abiity to resist and drop the malicious messages and the messages that generated from the untrusted vehicles.

There is no considerable effects of the CA in the message delivery rate and end-to-end delay, because the low contention on the medium and the high transmission rate minimizes the effects of the security overheads. Moreover, our proposed model framework (VSaaS) was discussed against the security requirements in VANET.

The VSaaS framework model is secure, efficient, and modular, managed by cloud and fulfills the security requirements.

6.2 Future Works

In the Future, we would like to work on the uncompleted portions, for example, the detecting submodule in the Vehicle Revocation Module (VRM) which is responsible for detecting a misbehaved vehicle. And, the Manufacturers authority which have a permission to provide all the firmware updates, and check the vehicle performance remotely.

Also, we would like to evaluate all type of messages and all modules of VSaaS on larger roadmaps with more vehicles using varying mobility models and investigate the performance matrices. Maybe, it is possible to measure the effects of different security techniques e.g. Hash-based Message Authentication Code (HMAC) and others.

This work opens the door for the researchers who interested in the cloud computing to identify, decribe and design the cloud environment needed to host this security model framework.

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

The Reference List

The Reference List

Accidents Data Statistics. (2011). *Road Safety in European Commission*. Retrieved December 12, 2015, from: http://ec.europa.eu/transport/road_safety/index_en.htm

Advanced Encryption Standard (AES). (2001, November). *NIST, FIPS PUB 197*. Retrieved January 4, 2016, from: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Al Mamun, M., Anam, K., Onik, M., & Esfar-E-Alam A. (2012, October 24-26). *Deployment of Cloud Computing into VANET to Create Ad Hoc Cloud Network Architecture*. Paper presented at the World Congress on Engineering and Computer Science, San Fanscisco, USA.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., et al. (2010). A View of Cloud Computing. *Communication of the ACM, 53(4)*, 50-58.

Baby, D., Sabareesh, R. D., Saravanaguru, R. A. K., & Thangavelu, A. (2013). *VCR: Vehicular Cloud for Road Side Scenarios*. Lecture in Advances in Computer and Information Technology, Springer, Berlin / Heidelberg, Germeny.

Barberis, C., Gueli, E., Minh Tuan, L., Malnati, G., & Nassisi, A. (2011, January 9-12). *A customizable Visualization Framework for VANET Application Design and Development*. Paper presented at 2011IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, USA.

Bayrak, A. O., & Acarman, T. (2010, June 21-24). *A Secure and privacy protecting protocol for VANET*. Paper presented at Intelligent Vehicles Symposium (IV), San Diego, USA.

Behrisch, M., Bieker, L., Erdmann, J., & Krajzewicz, D. (2011, October 23-29). *SUMO-Simulation of Urban MObility: An Overview*. Paper presented at the 3th International Conference on Advances in System Simulation, Barcelona, Spain.

Boneh, D., & Franklin, M. (2001, August). *Identity-Based Encryption from the Weil Pairings*. Paper presented at Annual International Cryptology Conference, Berlin / Heidelberg, Germeny.

Burmester, M., Magkos, E., & Chrissikopoulos V. (2008, October 12-14). *Strengthening Privacy Protection in VANETs*. Paper presented at 2008 IEEE International Conference Networking and Communications, Avignon, France.

Buttyán, L., Holczer, T., & Vajda I. (2007, July). *Effectiveness of Changing Pseudonyms to Provide Location Privacy in VANETs*. Lecture presented at European Workshop on Security in Ad-hoc and Sensor Networks, Springer, Berlin / Heidelberg, Germeny.

Calandriello, G., Papadimitratos, P., Hubaux, J. P., & Lioy A. (2007, September). *Efficient and Robust Pseudonymous Authentication in VANET*. Paper presented at the 4th ACM International Workshop on Vehicular Ad Hoc Networks, ACM.

Crypto++ 5.6.0 Benchmarks. (2009, March). *Cryptography Benchmarks*. Retrieved April 28, 2016, from: http://www.cryptopp.com/benchmarks.html

Festag, A., Noecker, G., Strassberger , M., Lübke, A., Bochow, B., et al. (2008, March). *NoW - Network on Wheels: Project Objectives, Technology and Achievements*. Paper presented at the 6th International Workshop on Intelligent Transportation (WIT 2008), Hamburg, Germany.

Fiore, M., Haerri, J., Filali, F., & Bonnet C. (2007, March). *Vehicular Mobility Simulation for VANETs*. Paper presented at the 40th Annual Simulation Symposium (ANSS 2007), Norfolk, Virginia.

Freudiger, G., Raya, M., & Feleghhazi, M. (2007, August 14-17). *Mix Zones for Location Privacy in Vehicular Networks*. Paper presented at the First International Workshop on Wireless Networking for Intelligent Transportation Systems (WiNITS), Vancouver, Canada.

Guo, J., Baugh, J. P., & Wang, S. (2007, May). *A Group Signature based Secure and Privacy-Preserving Vehicular Communication Framework*. Paper presented at 2007 Mobile Networking for Vehicular Environments, Anchorage, AK.

Hartenstein, H., & Laberteaux, K. P. (2008). A Tutorial Survey on Vehicular Ad Hoc Networks. *IEEE Communications Magazine, 46(6),* 164–171.

Huang, L., Matsuura, K., Yamane, H., & Sezaki K. (2005, March 13-17). *Enhancing Wireless Location Privacy Using Silent Period*. Paper presented at 2005 IEEE Wireless Communications and Networking Conference (WCNC), New Orleans, USA.

Hubaux, J. P., Capkun, S., & Luo, J. (2004). The Security and Privacy of Smart Vehicles, *IEEE Security and Privacy Magazine, 2(3),* 49–55.

Hussain, R., Abbas, F., Son, J., & Oh, H. (2013, May 13-16). "*TIaaS: Secure Cloud-Assisted Traffic Information Dissemination in Vehicular Ad Hoc Networks*". Paper presented at the 13th IEEE/ACM International Symposium on Cluster Computing and the Grid, Delft, Netherlands.

Hussain, R., Rezaeifar, Z., & Oh, H. (2015). A Paradigm Shift from Vehicular Ad Hoc Networks to VANET-Based Clouds .*Wireless Personal Communications: An International Journal Springer, 83(2),* 1131-1158.

Hussain, R., Son, J., Eun, H., Kim, S., & Oh, H. (2012, December 3-6). *Rethinking Vehicular Communication: Merging VANET with Cloud Computing*. Paper presented at 4th IEEE International Conference on Cloud Computing Technology and Science, Taipei, Taiwan.

Kamat, P., Baliga, A. & Trappe, W. (2008). Secure, Pseudonymous, and Auditable Communication in Vehicular Ad Hoc Networks. *Security and Communication Networks, 1(3),* 233 – 244.

Kamat, P., Baliga, A., & Trappe, W. (2006, September). *An Identity-Based Security Framework for VANETs*. Paper presented at the International Conference on Mobile Computing and Networking, Los Angeles, USA.

Kargl, F., Papadimitratos, P., Buttyan, L., Muter, M., Schoch, E., et al. (2008). Secure Vehicular Communication Systems: Implementation, Performance, and Research Challenges. *IEEE Communications Magazine, 46(11),* 110–118.

Khairnar, V. D., & Pradhan, S. N. (2013, April). Comparative Study of Simulation for Vehicular Ad-Hoc Network. *International Journal of Computer Applications, 4(10),* 75-87.

Khairnar, V., & Kotecha, K. (2013, October). Simulation-Based Performance Evaluation of Routing Protocols in Vehicular Ad-hoc Network. *International Journal of Scientific and Research Publications, 3(10),* 36-45

Khasa, Y., & King, P. (2016, March). Performance Evaluation of Routing Protocols in MANET. *International Journal of Computer Science Engineering and Technology (IJCSET), 6(3)* , 109-112.

Kleinjung, T., Aoki, K., Franke, J., Lenstra, A. K., Thomé, E., et al. (2010). *Factorization of a 768-bit RSA modulus*. Paper presented at the Annual Cryptology Conference, Springer, Berlin / Heidelberg, Germeny.

Krau S. (1997). *Microscopic Traffic Simulation: Robustness of a Simple Approach Tech. Rep.* Germany: DLR.

Kumar, S., Gollakota, S., & Katabi, D. (2012, August 13-17). *A Cloud-Assisted Design for Autonomous Driving*. Paper presented at the first edition of the MCC workshop on Mobile Cloud Computing. Helsinki, Finland.

Lai, C., Chang, H., & Lu, C.C. (2009, October 20-22). *A secure anonymous key mechanism for privacy protection in VANET*. Paper presented at the 9th International Conference on Intelligent Transport Systems Telecommunications, Lille, France.

Lin, T. W., Shen, J. M., & Weng, H. C. (2013). Cloud-Supported Seamless Internet Access in Intelligent Transportation Systems. *Wireless Personal Communication,* 72(4), 1-26.

Lin, X., Sun, X., Ho, P.-H., & Shen., X. (2007). GSIS: A Secure and Privacy Preserving Protocol for Vehicular Communications. *IEEE Transaction on Vehicular Technology, 56(6), 3442-3456.*

Mallissery, S., Pai, M. M., Ajam, N., Pai, R. M., & Mouzna, J. (2015, January 9-12). *Transport and Traffic Rule Violation Monitoring Service in ITS: A Secured VANET Cloud Application.* Paper presented at 12th Annual IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, USA.

Mallissery, S., Pai, M. M., Pai, R. M, & Smitha. A. (2014, November). *Cloud Enabled Secure Communication in Vehicular Ad-hoc Networks*. Paper presented at IEEE International Conference on Connected Vehicles and Expo (ICCVE) , Vienna, Austria.

Mercedes VIN Shopping Tips. (n.d.). *Vehicle Identification Number (VIN)*. Retrieved February 1, 2016, from: http://www.autohausaz.com/mercedes-auto-arts/mercedes-vehicle-identification-numbers.html

Mershad, K., & Artail, H. (2013). Finding a STAR in a Vehicular Cloud. *IEEE Intelligent Transportation Systems, 5(2),* 55–68.

Mishra, B., Panigrahy, S. K., Tripathy, T. C., Jena D., & Jena S. K. (2011, December 11-14). *A Secure and Efficient Message Authentication Protocol for VANETs With Privacy Preservation.* Paper Presented at the 2011 World Congress on Information and Communication Technologies, Mumbai, India.

Olariu, S., Eltoweissy, M., & Younis, M. (2011). Towards Autonomous Vehicular Clouds. *ICST Transactions on Mobile Communications and Applications, 11(7),* 1-11.

Olariu, S., Hristov, T., & Yan, G. (2012). *The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds: The Cutting Edge Direction*. USA: Wiley.

Olariu, S., Hristov, T., & Yan, G. (2013). *The Next Paradigm Shift: From Vehicular Networks to Vehicular Clouds: The Cutting Edge Direction*. (2nd ed.). USA: Wiley-IEEE Press.

Papadimitratos, P., Buttyan, L., Holczer, T., Schoch, E., Freudiger, J., et al. (2008). Secure Vehicular Communication Systems: Design and Architecture. *IEEE Communications Magazine, 46(11),* 100–109.

Pearson, S., & Benameur, A. (2010, November). *V2C: Privacy, security and trust issues arising from cloud computing*. Paper presented at the IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), USA.

Plobl, K., & Federrath, H. (2008). A Privacy Aware and Efficient Security Infrastructure for Vehicular Ad Hoc Networks. *Computer Standards & Interfaces, 30(6),* 390–397.

Public Health and the Epidemic of Motor Vehicle Crashes. (2003, November). *National Highway Traffic Safety Administration*. Retrieved December 12, 2015, from: http://www.nhtsa.gov

Rangarajan, S., Verma, M., Kannan, A., Sharma, A., & Schoen, I. (2012, August). *V2C: A Secure Vehicle to Cloud Framework for Virtualized and On-Demand Service Provisioning*. Paper presented at the International Conference on Advances in Computing, Communications and Informatics, India.

Raya, M., & Hubaux, J. P. (2005). *The Security of Vehicular Ad Hoc Networks*. Paper presented at the 3rd ACM workshop on Security of ad hoc and sensor networks, New York, USA,

Raya, M., & Hubaux, J. P. (2007). Securing Vehicular Ad Hoc Networks. *Journal of Computer Security, 15(1),* 39–68.

Rohal, P., Dahiya, R. & Dahiya, P. (2013, March). Study and Analysis of Throughput, Delay and Packet Delivery Ratio in MANET for Topology Based Routing Protocols (AODV,

DSR and DSDV). *International Journal for Advance Research in Engineering and Technology (IJARET), 1(2),* 54-58.

Samara, G., Al-Salihy, W., & Sures, R. (2010, December 9-11). *Security Analysis of Vehicular Ad Hoc Networks (VANET)*. Paper presented at the 2th IEEE International Conference. on Network Applications, Protocols and Services, Bangalore,India.

Samara, G., Al-Salihy, W., & Sures, R. (2010, May 11-13). *Security Issues and Challenges of Vehicular Ad Hoc Networks (VANET)*. Paper presented at IEEE 4th International Conference on New Trends in Information Science and Service Science (NISS), Gyeongju, Korea.

Samara, G., Al-Salihy, W., & Sures, R. (2010, May 11-13). *Security issues and challenges of Vehicular Ad Hoc Networks (VANET)*. Paper presented at the 4th IEEE International Conference on New Trends in Information Science and Service Science (NISS), Gyeongju, Korea.

Sampigethaya, K., Huang, L., Matsuura, K., Poovendran, R., & Sezaki, K. (2005). *Caravan: Providing Location Privacy for VANET*. Paper presented at the 3rd Embedded Security in Cars Workshop (Escar).

Serna, J., Luna, J., & Medina, M. (2008, September). *Geolocation-Based Trust for VANETs Privacy*. Paper presented at the 4th International Conference on Information Assurance and Security, Korea.

Shamir, A. (1984, August). *Identity-Based Cryptosystems and Signature Schemes*. Paper presented at Workshop on the Theory and Application of Cryptographic Techniques, Berlin / Heidelberg, Germeny.

Sommer, C., German, R., & Dressler, F. (2011, January). Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis, *IEEE Transactions on Mobile Computing, 10(1),* 3-15.

Sun, J., Zhang, C., Zhang, Y., & Fang, Y. (2010). An Identity-Based Security System for User Privacy in Vehicular Ad Hoc Networks. *Parallel and Distributed Systems, IEEE Transactions, 21(9),* 1227 – 1239.

The NS Manual, Formerly NS Notes and Documentation. (2011, November). *The VINT Project.* Retrieved March 1, 2016, from: http://www.isi.edu/nsnam/ns/doc/

Traffic and Network Simulation Environment. (2012). *TraNS Offical Website*. Retrieved March 15, 2016, from: http://lca.epfl.ch/projects/trans/

Xu, Q., Mak, T., Ko, J., & Sengupta, R. (2004). *Vehicle-to-Vehicle Safety Messaging in DSRC*. Paper presented at the first ACM workshop on Vehicular ad hoc networks, New York, USA.

Yan, G., & Olariu, S. (2009, October). *An efficient Geographic Location-Based Security Mechanism for Vehicular Ad Hoc Networks*. Paper presented at IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, China.

Yan, G., Ibrahim, K., & Weigle, M. C. (2009). *Vehicular Network Simulators, In Vehicular Networks: From Theory to Practice*. UK: Crc Press.

Yan, G., Wen, D., Olariu, S., & Weigle, M. (2012). Security Challenges in Vehicular Cloud Computing. *IEEE International Transactions on Intelligent Transportation Systems, 14(1),* 284-294.

Yang, X., Liu, J., Zhao, F., & Vaidya N. (2004, August 22-26). *A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning*. Paper presented at the first Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous 2004), Boston, Massachusetts, USA.

Yoon, J., Liu, M., & Noble, B. (2003, March). *Random Waypoint Considered Harmful*. Paper presented at the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, San Francisco, USA.

Zhang, C., Lin, X., Lu, R., & Ho, P. H. (2008, May 19-23). *RAISE: an Efficient RSU-Aided Message Authentication Scheme in Vehicular Communication Networks*. Paper presented at 2008 IEEE International Conference on Communications (ICC), Beijing, China.

Zingirian, N., & Valenti, C. (2012, June 3-7). *Sensor Clouds for Intelligent Truck Monitoring*. Paper presented at Intelligent Vehicles Symposium, Alcala de Henares, Spain.