Security management in vehicular ad hoc networks

Tayeb Diab

University of Haute-Alsace

Ecole Doctorale (269) Mathematics, information science and engineering

Laboratory (IRIMAS) Institut de Recherche en Informatique, Mathématiques, Automatique et Signal

Speciality: Computer sciences

PHD THESIS

A Dissertation Submitted in Fulfilment of the Requirements for the Degree of DOCTOR at University of HAUTE ALSACE
23/01/2020

by

Tayeb DIAB

Security management in vehicular ad hoc networks

In front of the jury composed of:

M.Ahmed SERHROUCHNI Prof, Telecom Paris Tech, Paris	**Rapporteur**
M. François SPIES Prof, University of Franche-Comté, Besançon	**Rapporteur**
M. Jaafer GABER HDR, UTBM, Belfort	**Examiner**
M. Frédéric DROUHIN MCF, University of Haute-Alsace, Colmar	**Examiner**
M. Pascal LORENZ Prof, University of Haute-Alsace, Colmar	**Supervisor**
M. Marc GILG HDR, University of Haute-Alsace, Colmar	**Co-Supervisor**

For my father, my mother, my brothers, my sisters
and my dear deceased brother **Mustapha**.

Acknowledgments

I would like to express my gratitude to Professor Pascal Lorenz, the director of my thesis for his effort and patience. A very special thank goes to Dr Marc Gilg and Dr Frédéric Drouhin for their expertise, understanding, knowledge, valuable advice, help and never-ending support during this thesis.

I appreciate my supervisors effort to guide me during this thesis without which this work would have not been possible.

My respect, gratitude and my deep love for my father, my mother, my brothers, my sisters and my dear deceased brother Mustapha. They make it easy for me and encouraged me to achieve my goals in my life.

Finally, I would like to thank my friends for accepting nothing less than excellence from me.

Publications

T. Diab, M. Gilg, and P. Lorenz. A secure communication model using lightweight Diffie-Hellman method in vehicular ad hoc networks. International Journal of Security and Networks, 14(2):61–77, 2019.

T. Diab, M. Gilg, F. Drouhin and P. Lorenz "Anonymizing communication in VANETs by applying I2P mechanisms", IEEE Globecom'19, Waikoloa, USA, 2019.

Abstract

Vehicular networks (VANet) are particular ad hoc mobile networks where high speed and high vehicle mobility are critical constraints for this type of networks. Thus, some applications related to road safety may be vulnerable to different attacks in the network.

In this thesis, we are interested in protecting the privacy of drivers. To do this, the Invisible Internet (I2P) project is used and adapted to the VANet context. I2P is an anonymous network, in which applications can exchange messages anonymously and securely. The operating principle of I2P is based on creating encrypted tunnels between the communicating nodes, which reinforces the anonymity and therefore the security of the communication.

The problem studied in this thesis is to establish and maintain encrypted tunnels in VANet while providing anonymity and facing different critical characteristics of VANet.

Résumé

Les réseaux véhiculaires (VANet) sont des réseaux mobiles ad hoc particuliers où la vitesse élevée et la forte mobilité des véhicules constituent des contraintes critiques dans ce type de réseaux. Ainsi, certaines applications liées à la sécurité routière peuvent être vulnérables à différentes attaques dans le réseau.

Cette thèse s'intéresse à l'anonymat de la communication dans les VANets. Notre objectif est de trouver des algorithmes permettant de sécuriser cette communication et la rendre anonyme.

Pour ce faire, le protocol "the Invisible Internet Project" (I2P) est utilisé et adapté au contexte des VANets. I2P est un réseau anonyme dont les applications peuvent échanger les messages de manière anonyme et sécurisé. Le principe de fonctionnement d'I2P est basé sur la création des tunnels chiffrés entre les nœuds communicants, ce qui permet de renforcer l'anonymat et donc la sécurité de la communication.

La problématique étudiée est d'établir et maintenir des tunnels chiffrés dans un réseau VANet tout en garantissant l'anonymat et en faisant face aux différentes caractéristiques critiques des VANets.

Contents

List of Figures

List of Algorithms

List of Tables

INTRODUCTION

Today, Vehicular ad hoc networks (VANet) became an interesting area of research, where new fields appeared and need more studies to find solutions and provide more services. The VANet represents a specific mobile ad hoc network (MANET) [60], where the nodes are vehicles and roadside units (RSUs).

Intelligent transport systems aim to enhance the road safety and minimize the number of accidents. Besides, they provide other services and features allowing good travel conditions and better vehicle traffic, without congestion and with a minimum pollution level. However, the critical characteristics of VANet such as the high speed, mobility of vehicles, the rapid topology change and the variety of communication environments may complicate the development of new algorithms and applications.

VANet infrastructures provide different types of information depending on the context of the application being executed. An information message may concern accidents, climate conditions, traffic jams, location of service facilities, etc. The exchange of these messages in the network must be secure and fast, especially for applications related to the traffic safety that can be critical and vulnerable to different intrusions. Therefore, it requires the implementation of robust, efficient and secure protocols for a good level of quality of service and security.

Evolution of VANet last years opens up a vast area of research in several fields. Especially, in the security domain, various studies have been performed to propose new approaches and mechanisms of security. In VANet, attackers can disrupt communication by generating malicious activities such as messages forgery, preventing legitimate vehicles from accessing network services and eavesdropping communication to launch attacks later. In VANet, attacks can be classified into several types according to several parameters like the number of attackers and the type of malicious activity occurred. Sybil attack, Denial of service, Blackhole, Wormhole, Eavesdropping, False position information, Man In The Middle attacks are among the most known attacks in VANet [71].

VANets as wireless and mobile networks require the implementation of algorithms and methods with a high level of robustness and security to face attacks and deal with critical characteristics of the network. The mobility of vehicles in VANet can help the attacker during the attack to move and

change the position, which makes it difficult to locate it. However, the mobility of the attacked vehicles, can make it difficult for the attacker to accomplish its attack.

Communication security involves using protocols providing several security services such as availability, integrity, confidentiality, authenticity, non-repudiation and anonymity (in some cases). Safety applications in VANet are the most critical category. It provides security of individuals in such cases of accidents, collision and road condition warnings, which requires the availability of different network services all the time. Besides, at the level of vehicle authority services like communication between policy, civil protection or gendarmerie vehicles must be secure, which is the same requirement for the driving improvement application category. This refers to authenticate the users and prevent the VANet against suspected entities to ensure their legitimacy. In addition to provide confidentiality and data integrity of the exchanged messages, which ensure that the designated receiver has access to the data while outside entities have not.

Anonymizing communication is an important issue to improve security and to deal with intrusions. The anonymity of the communication highlights another aspect of security, which consists of hiding real identities of the sender and recipient nodes. In ad hoc vehicular networks, the anonymity concept is used in different applications such as the military or the civilian sector, in which an intermediate node (possibly malicious) must not know neither the sender identity nor the recipient identity. In parallel with security, robust and efficient methods must be used to obtain a minimum delay time and a good level of reliability to address critical characteristics of VANet and to achieve the expected results.

In this thesis, we aim to provide a secure and anonymous communication in VANet, in which we propose a secure model inspired by the Invisible Internet Project (I2P), taking into account the critical constraints of this category of networks. We choose I2P [20] as a reference model thanks to its high level of security and anonymity of the communication on the Internet.

I2P is an anonymous subnet, where applications can exchange messages anonymously and securely. The operating principle of I2P is based on using encrypted tunnels between nodes, which reinforces the anonymity and thus the security of the communication within the network. Besides, I2P uses four levels of encryption: I2CP (I2P Control Protocol) encryption, Garlic encryption, tunnel encryption, and transport encryption.

I2P is designed to the internet, which has different characteristics compared to VANet. We treat this difference by adapting security mechanisms and algorithms used in I2P to be suitable for VANet and cope with its different characteristics. To do so, we divide the work into two parts (two contributions):

1- Apply the I2P transport encryption level in VANet using Diffie-Hellman method [79] and signature mechanisms.

2- Create and maintain encrypted tunnels and implement the garlic and tunnel encryption algorithms in VANet.

The first work is an initiation to adapt the I2P protocol to VANet, in which we try to apply the I2P transport encryption mechanism in this category of networks. We use the Diffie-Hellman method to securely share the secret key and achieve the integrity, confidentiality and non-repudiation services. In this contribution, we deal with two attacks: the Man-In-The-Middle (MITM) attack and the Replay attack using a signature mechanism to authenticate the participating nodes. These attacks can be launched in different ways and at different times. In the attempt to prevent it, a signature mechanism and a communication model is used to detect the attack at the launch time.

We continue this work by adapting some mechanisms and algorithms of I2P in VANet in the second contribution. This contribution can be highlighted in two major points:

1- We create tunnels, implement the garlic and tunnel encryption layers of I2P, which represents the first version of the proposed protocol.

2- We develop the second version of the protocol by adding a tunnel maintenance algorithm to maintain the existence of the created tunnels in the network.

We show the effectiveness and security of the proposed model by analyzing the different cases of anonymity. We have launched the simulations using the NS3 network simulator [64]. The results show a low Packet Delivery Ratio (PDR) in our protocol compared to the Ad-hoc On-Demand Distance Vector (AODV) protocol [46] (in some cases). The overhead is significantly increased due to the security algorithms implemented within the protocol.

The on-demand process for discovering routes in AODV can be time-consuming and delays the communication, while in our protocol, paths can be created during the tunnel creation and maintenance processes, which relatively reduces the end-to-end delay.

This thesis is organized into five chapters. In the first chapter, we present a general overview of vehicular ad hoc networks. We define general concepts and cover VANet characteristics. We classify applications in four classes, present entities and architectures of communication and different types of environments existed in VANet. Besides, we cover famous mobility models and routing protocols and show some standards and technologies used in wireless vehicular communication. In the end, we present in general the most challenges and issues that can be discussed in different works in VANet.

The second chapter concerns security in VANet. It defines general concepts about security services and QoS (Quality of Service) parameters.

Besides, different attacks known in this category of networks are classified and some proposed solutions for some of these attacks are presented.

In chapter 3, we detail the anonymity service by presenting I2P, Freenet and Tor as the most known protocols providing security and anonymity on the Internet. After conducting the experimentation, I2P has shown to be the most suitable for securing and anonymizing the communication in VANet. Then, we present the I2P protocol in detail and explain the general concept and its different protocol stack layers. We explain its operation principle regarding the exchange of messages using tunnels, processing messages inside and between tunnels and tunnel creation process. We present different I2P encryption levels. Then, we show the difference in characteristics between VANet and I2P network and we explain our adaptation of I2P within VANet according to the different contributions that we have proposed.

Chapter 4 represents the first contribution of our thesis. We propose a novel approach of security to face several well-known attacks. We design a model of communication that combines digital signature and message authentication mechanisms to securely generate the secret key. Therfore, achieve integrity, confidentiality, session key security and non-repudiation. In this model, we implement the first encryption layer of I2P called "transport encryption" as an initiation in our work. Finally, we show the security of our model by analyzing different cases of attacks.

In chapter 5, we present the second contribution, in which we propose a model of security to ensure anonymity in the vehicular ad-hoc network. This model is inspired from the Invisible Internet Project (I2P), in which we continue our previous work by adapting some of the I2P mechanisms and algorithms in VANet. The I2P protocol is adapted to respond to several requirements of VANet. The proposed model is based on tunnels and encryption algorithms that use digital signatures and authentication mechanisms. Here, we create tunnels and maintain their existence. Moreover, the tunnel and garlic encryption layers are implemented in VANet. We show the effectiveness and the security of our proposed model by analyzing different cases of anonymity and showing the performance results. It is important to point out that the simulations have been performed using the NS3 platform.

We end this thesis by presenting a conclusion and some prospects.

Chapter 1

OVERVIEW ON VEHICULAR AD HOC NETWORKS

1.1 Introduction

Recent developments in telephone and computer communications technologies have resulted in new solutions addressing many network problems. With this evolution, many types of networks have appeared in different domains, each of which has its operating context citing for example wired networks, wireless, with or without infrastructure, etc.

Mobile wireless networks or MANETs are ad hoc networks constituting a set of mobile units communicating via a radio medium and require neither fixed infrastructure nor centralized administration [37]. This type of networks has seen a great evolution in terms of hardware and software tools. Thus, new particular types of MANETs appeared in the same context with differences in some parameters (speed, direction, etc.) for example; sensor networks, wireless vehicular networks, etc.

During the last decades, the number of road accidents has exceeded the limits. Therefore, incredible results of the number of dead and wounded. Among the main reasons; high speed of vehicles, driving under the influence of alcohol, fatigue, drowsiness, using the phone while driving, etc. In 2007, 110 people died and more than 4,600 wounded with a cost of more than 438 million euros a day were caused by road accidents in the European Union. Similarly in 2007, in the United States, these accidents killed 102 people and 7900 wounded at a cost of more than 630 million dollars daily [65]. Furthermore, traffic congestion has become a problem in some countries, mainly in developed countries. This is due to its harmful effects in terms of air pollution, fuel consumption and therefore greenhouse gas (GHG) emissions as well as time lost by users in transport.

Making solutions for the road accidents and traffic congestion effects, studies are necessary. Many governments, car manufacturers and the industrial consortium have set the reduction of road accidents as a major

priority [65]. The idea was to make road networks smart to enable communication between nodes via wireless communications and to make adaptive decisions in different critical situations, which gives the birth of ITS (intelligent transport systems). ITS are transport systems that use the new information and communication technologies (NTICs) to improve road safety, traffic efficiency, road user comfort and reduce the negative environmental impact caused by gas emissions and pollution generated by road traffic. ITS makes transportation more automated, which gives rise to a new class of networks; wireless vehicular networks or vehicular ad hoc networks (VANet).

The potential applications in this type of networks aim to help drivers by providing traffic information, meteorological, near accidents and so on to make driving safer and passenger travel more comfortable using convivial applications such as network games, internet access, discussion group in a traffic jam, etc.

This chapter presents a general idea about the vehicular ad hoc networks. It presents the different types of applications, communication architectures, deployment environments and characteristics of VANet. Besides, it cites some models of vehicle mobility and access technologies used in communication before presenting the standards of vehicular communication. Finally, it presents some challenges and issues that face the development of this type of network.

1.2 Definitions and generalities on VANet

VANet as a new category of mobile ad hoc networks, they have different characteristics and use specific applications adapted to their requirements. As below, this section presents a general view about this category of networks, details its characteristics and classify its applications.

1.2.1 Definition of VANet

VANet defines a specific type of mobile ad hoc networks (MANET). Nodes within the network can be categorized in two types: mobile nodes which are vehicles and fixed nodes called RSU (Road Side Units) located in different environments (urban, rural or highway) and in critical locations such as slippery roads, service stations, intersections, places with a dangerous climate, etc. (figure 1-1) [42]

The data exchange between nodes in the network is done in a wireless transmission space and in a multi-hop way. The vehicles use vehicle-to-vehicle communication (V2V) (or inter-vehicular communication IVC) to communicate with each other and with the RSUs via vehicle-to-roadside communications (V2I). Each vehicle acts as receiver, transmitter and router to allow the exchange of data in the network.

In MANETs, movement of nodes is arbitrary, however in VANet, vehicles move according to restrictions in directions and speed. Vehicles traffic is based on mobility models where roads, intersections, buildings, etc. are predefined according to the vehicular environment.

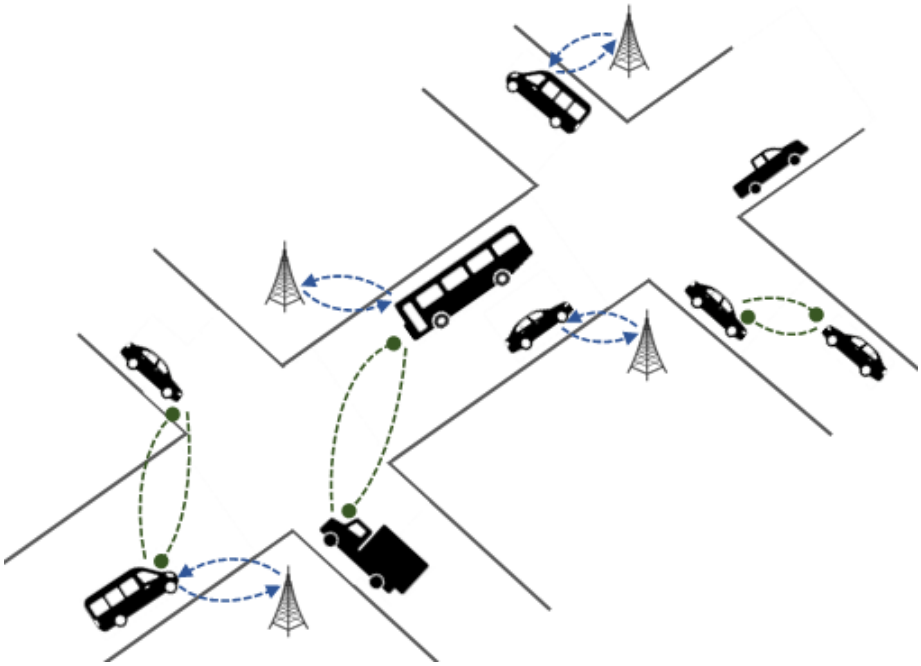


Figure 1-1: Vehicular ad hoc network

1.2.2 VANet characteristics

This section reviews the main characteristics of vehicular ad hoc networks:

1.2.2.1 Mobility

mobility of vehicles represents a critical characteristic that influences the behaviour of communication within the network. VANets have specific mobility compared to several know mobile networks. This section clarifies the particularity of VANet considering the mobility of nodes.

a. High mobility

Unlike conventional wireless networks, in VANet, the speed of vehicles is very high comparing to mobile nodes in other networks, and therefore the communication time between nodes will be shorter, which poses significant problems of radio propagation instability. The lifetime of links on a motorway is around 50 seconds for vehicles going in the same direction, but less than 5 seconds to the vehicles moving in the opposite directions [65].

In this context, we can talk about mobile ipv6 [53] which uses mechanisms supporting the mobility of nodes. Unlike the IP protocol version

4, IPv6 integrates mobility management through extensions on the IPv6 header. It has specific data structures namely association table and list of association updates, used to maintain the temporary addresses of mobile nodes.

In the case of IPv4 mobility, a mobile node can always be reached by its main address (the mother address), whether it is attached to its original network (mother network) or whether it is distant from it. In contrast, ipv6 uses the notions of neighbor discovery and auto-configuration, as well as it uses options included in the header of the IPv6 packet, which are defined specifically for mobility support in IPv6. This allows a user to obtain a temporary IP address corresponding to the main address assigned by the home agent. The registration phase with the home agent is done once this temporary address has been obtained, then they are stored in the association table.

b. Mobility models

In VANet, vehicles follow specific directions according to the roads and the environment characteristics like buildings, junctions, traffic regulations, etc. This implies that vehicles follow regular and limited mobility patterns according to these characteristics. Using mobility models and limitation of vehicles directions provide significant results close to reality. Three mobility models can be distinguished according to the environments of the vehicular networks: highway, rural and urban models.

c. Predictable mobility

The availability of certain information such as the average speed of vehicles and the path of roads can lead to predicting the next position of the vehicle, in which vehicle displacement models can play the main role in this prediction [65].

1.2.2.2 Energy

Energy is a major constraint in traditional mobile networks and this affects the computing capacity and the quality of applications. However, in VANet, the communication entities have an efficient power system that provides sufficient energy capacity to power the communication platform. Even when the vehicle engine is stopped (power system shutdown), the embedded platform can use the battery device [65].

The absence of energy constraint in these types of networks leads to benefit from massive and important calculation capabilities. Nevertheless, the appearance of electric vehicles and the addition of new sensors, network interfaces, applications, etc. vehicle energy management becomes more complex, which involve developing energy-saving solutions that can be used soon.

1.2.2.3 Velocity of vehicles

The velocity of a node is the rate of change of its position versus time, its value is from 0 km/h in traffic jam situations to more than 200 km/h in highways [42]. Vehicle velocity is an important factor in vehicular network applications. High vehicles velocities make the duration of the communication between two nodes very short. Therefore, the routing process, for example, will be run multiple times to find the path to the destination. The frequent execution of these processes involves a considerable delay during the communication, thus the loss of packets when using expired paths. However, when vehicles move with low velocities (in traffic jam situations for example), the network will be dense which causes connection problems between nodes such as interference between signals.

1.2.2.4 Density of vehicles

In vehicular ad hoc network environments, the number of nodes varies according to the situation of vehicles. In case of congestion due to accidents, for example, the transmission range of a node may exceed 200 nodes. In this case, the density is maximum and therefore network problems may arise such as overhead, interference, choices of long paths, etc. However, in rural environments or at a time of low traffic, the number of nodes decreases and the network can have a low density, which influences the communication due to the lack of intermediate nodes. As a solution, several copies of the message must be saved and retransmitted later by the same transmitter. Nevertheless, this solution causes a significant delay in dissemination of the packets [42].

1.2.2.5 Heterogeneity of nodes

Heterogeneity of nodes is one of the properties of vehicular ad hoc networks. In this context, the term heterogeneity means the diversity of nodes according to two different aspects: structural and functional [42]. The structural aspect concerns the structure of the node, in which two types of nodes can be distinguished: vehicles and RSUs. For the functional aspect, nodes can be categorized according to the types of applications used, for example, the control applications installed on authority vehicles or RSUs, ad hoc and emergency applications executed by privet vehicles or authorities, maintenance and warning applications made by emergency vehicles, etc.

1.2.2.6 Communication environments

VANets impose high environmental diversity, in which a high number of obstacles, buildings, trees can exist, which influences the communication between vehicles and with RSUs. The mobility of vehicles allows them to move from one environment to another (urban to a highway for example)

with different characteristics (buildings, junctions, traffic regulations, etc.) which lead to models of complex wave propagation.

1.2.2.7 Geolocation

The geospatial positioning of vehicles and roads in an autonomous and precise way is an important and necessary service for the good execution of certain applications in vehicular networks. Several existing positioning systems like GNSS and DGPS can be used in VANet as tools to accomplish the different services required for an application [65].

GNSS (Global Navigation Satellite System) is a satellite positioning system more attractive than location systems based on radar, lidars, ultrasonic sensors, cameras, etc. It proposes a global clock and a system of terrestrial coordinates common between the applications distributed on the vehicles. GPS (Global Positioning System) is an example of GNSS where the current accuracy of GPS receivers integrated with vehicles is of the order of 10-15 meters which is useful for guidance. DGPS (Differential GPS) is an evolution of GPS technology to improve accuracy. The best precision obtained is about one meter. Another technique that allows the best accuracy is real-time kinematics (RTK). This technique promises a precision of the order of one centimetre. Satellite positioning techniques pose the problem of signal loss when vehicles pass through tunnels and dense forests for examples. One solution to this problem is to merge GNSS and vehicle inertial sensors.

1.2.2.8 Communication Models

Communication models represent the way of delivering messages from the source to the destination [65]. Point to point communication is one of these models, in which two nodes can communicate with each other. Broadcast, for example, can be used to send messages to all the nodes in the network.

Another type of models mostly used in VANet is geocast, in which a node sends messages to a defined geographical area. Platoon or the "train of vehicles" is an example of an application using broadcast communication. Its principle is that a set of vehicles are organized in a convoy. A vehicle is elected as a convoy leader who broadcasts information (speed changes for example) and the vehicles of the convoy follow it.

1.2.2.9 Network topology and connectivity

Given the mobility of the nodes, the vehicle can join and leave the network in a short time, which changes the topology and causes network partitioning frequently [65]. Connectivity of nodes in vehicular networks is one of the key parameters that must be taken into consideration by the new proposed solutions.

1.2.2.10 Frequent exchange of information

As an ad hoc network and the presence of different types of nodes such as vehicles and RSUs, VANets are considered as networks with high-frequency exchange. Nodes exchange information frequently to maintain the existence of the network and during the running of applications [50].

1.2.3 VANet applications

Applications in VANet are realized for the benefit of passengers and transport authorities by providing, for example, highway management, weather management of the road, collision prevention and safety, etc. A consortium of manufacturers (General Motors, Chrysler Daimler, Toyota, Nissan, Volkswagen, Ford, BMW) has established a report 75 applications [65].

Applications deployed in vehicular ad hoc networks can be classified into four classes:

1.2.3.1 Traffic safety applications

Decreasing the number of accidents and thus the number of wounded and dead is one of the main goals behind developing and studying vehicular communications. These applications help to improve the driver's vision by providing driving assistance where he can anticipate and act to make driving safer. Security messages are transmitted between the different nodes in the network (vehicles and/or RSUs), they can carry information about the state of the vehicle or the traffic. The messages describe the state of the vehicle concerning the status of the brake, the traffic lights, etc. The traffic status can be described according to the velocity of the vehicles, acceleration, number of pedestrians, braking or collision warnings, road condition warnings (ice, obstacle), etc [65][42].

This type of application helps the driver to make good decisions to avoid certain critical situations of death or injury to passengers. For example, safety systems can inform the driver about unexpected actions to be careful like passing the red light by a vehicle or crossing the road by a pedestrian. In case of an accident (when the airbag is triggered for example), a message is sent to a nearby rescue centre to intervene as soon as possible.

1.2.3.2 Vehicle Authority Services

Ad-hoc vehicular networks are also operated by transport authorities such as police, gendarmerie, emergency recovery units, etc. Authority vehicles contribute to vehicle safety and emergency improvement by issuing other warning messages to other vehicles to inform emergency vehicles using virtual sirens, or to facilitate the passage of authority vehicles in case of a traffic jam for example.

Besides, another type of authority service: traffic monitoring. Nodes detect and transmit information to the authority centres using surveillance applications such as stolen vehicle tracking, vehicle safety inspection, electronic license plate verification, conduit permit verification, etc [42]. Such applications must comply with security requirements and require a discussion of the legal aspect of vehicular communications.

1.2.3.3 Driving improvement applications

This type of applications provides information on the local and global environment of vehicles to optimize road traffic and prevent congestion [42]. Road traffic information may be exchanged locally between vehicles and/or RSUs to improve driving, for example, the dissemination of weather information to suggest beneficial actions such as using the air conditioner in areas of congestion or pollution, fog lamp lighting in case of fog, etc.

Global Traffic Information can be sent by a remote node to all other nodes in the entire network. This information concerns the overall state of the network. Drivers can use this type of information to avoid certain critical situations, for example, in case of congestion, information indicating this situation will be issued to long-distance vehicles to allow them to choose other faster routes.

In addition, other comfort applications that make travelling in good conditions are included. These applications provide a source of information regarding the location of fuel stations, weather, etc.

1.2.3.4 Business and entertainment applications

This class of application provides drivers and passengers with commercial services and entertainment via the Internet or private networks [42]. This category offered a lot of services in different areas such as offers of restaurants, location of vehicles for the tourist purpose, etc. Telematic services like highway tolling, automatic payment at service stations, interactive multimedia services (downloading videos, online games, etc.), point-to-point communication between two drivers travelling together, internet access, exchange of messages and sharing of data (videos, network games ...), etc [65]. In this type of applications, commercial services must guarantee all conventional business requirements such as transaction security and confidentiality, secure payment and so on [42].

1.3 Architectures and components of vehicular ad hoc networks

1.3.1 The communication entities

Vehicular ad hoc networks can include four communicating entities organized according to communication architectures [65]:

1.3.1.1 Personal equipment

It represents any equipment used by the user inside the vehicle, in which it can interact with the vehicle such as a phone, laptop, autonomous GPS, etc. As an example, by activating the Bluetooth interface of the mobile phone, it is possible to use the microphones integrated into the vehicle to use it as a telephone via the human-machine interface (HMI) of the vehicle [65].

1.3.1.2 The On-Board Unit (OBU)

Intelligent vehicles are equipped with a central platform that has wired and wireless interfaces and devices as represents figure 1-2 [65]. This platform is connected to a set of processors, in addition to a wireless On-Board Unit (OBU) for communication with other vehicles and RSUs. This interface constitutes heterogeneous digital devices (processor (CPU), random access memory, etc.) performing calculation applications [42]. The OBU can record, compute, locate and send messages on a network interface using equipment forming a DSRC (Dedicated Short Range Communication) system [65]. Thus, the global positioning system (GPS) integrated with the vehicle that facilitates the location service.

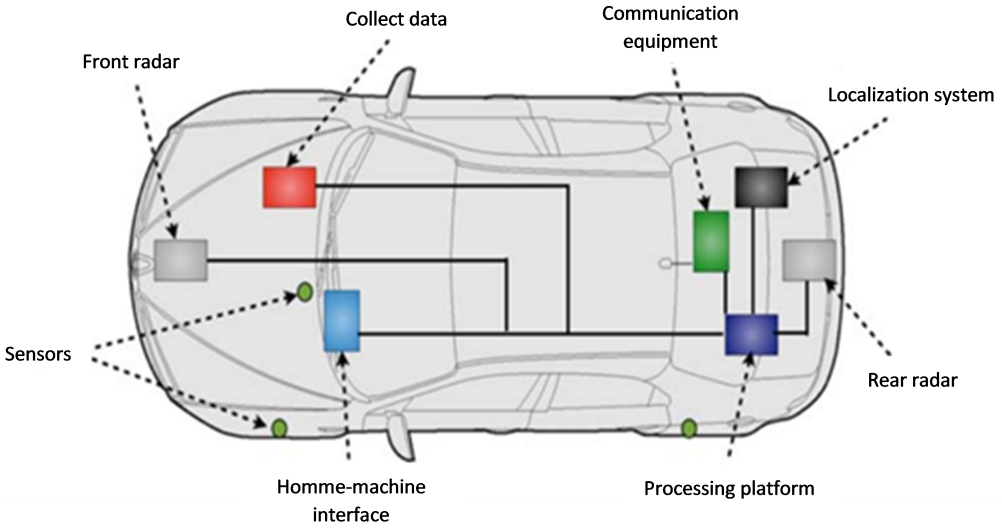


Figure 1-2: Example of a smart vehicle

1.3.1.3 Road-Side Unit (RSU)

Road-side equipment is stationary nodes located along the road allowing vehicles to connect to the global network [65]. They can be used to inform nearby vehicles by broadcasting information concerning traffic conditions, meteorological or specific to the road for example (maximum speed, overtaking authorization, etc.). Public bodies and highway operators can use RSUs according to their applications. During the communication, RSUs can also act as base stations for relaying information sent by the vehicles.

1.3.1.4 The central equipment

This equipment is located on the "server" side [65]. It is transparent to the user. This central equipment may be a storage server, an entry point to a wired network (Internet), transaction server (electronic toll for example), etc.

1.3.2 Communication architectures

The old traffic management systems are based on a centralized architecture. Their principle is to implement a set of cameras and sensors on the road to collect information on the traffic status of vehicles, and any information collected will be sent to a central entity for processing, which makes the process of transfer and processing messages slower (in the order of minutes) than the time required in certain situations. Besides, these systems require a large investment dedicated to the installation of sensors and cameras and their maintenance. The transmission delay in these systems and the cost of installation of equipment on the roads (especially on a large scale) represents a real obstacle to investment in this type of systems [65].

With the rapid development of wireless telecommunications technologies, new localization and message collection systems based on a decentralized architecture have appeared. Here, the vehicular ad hoc network is considered as an application of mobile ad hoc networks. Its architecture is based on a distributed and autonomous system where vehicles and RSUs communicate with each other to relay messages without the need for a central entity. In recent years, this raises a real interest among the scientific community, auto manufacturers and telecom operators [65].

Three types of communication architectures can be distinguished in vehicular networks: V2V, V2I and hybrid.

1.3.2.1 Vehicle-to-Infrastructure Communication (V2I)

This architecture is based on two types of nodes: wireless mobile nodes constituting the vehicles and fixed infrastructure nodes (RSUs) connecting the vehicles with global networks such as the Internet (figure 1-3). A vehicle can connect to an RSU if it is within its coverage area [42]. When

an obstacle exists, for example, the vehicle sends a message to the road service, this communication is unidirectional (from the OBU to RSU). When an RSU sends information to vehicles in its area is an example of I2V communication. Generally, communication V2I means both V2I and I2V.

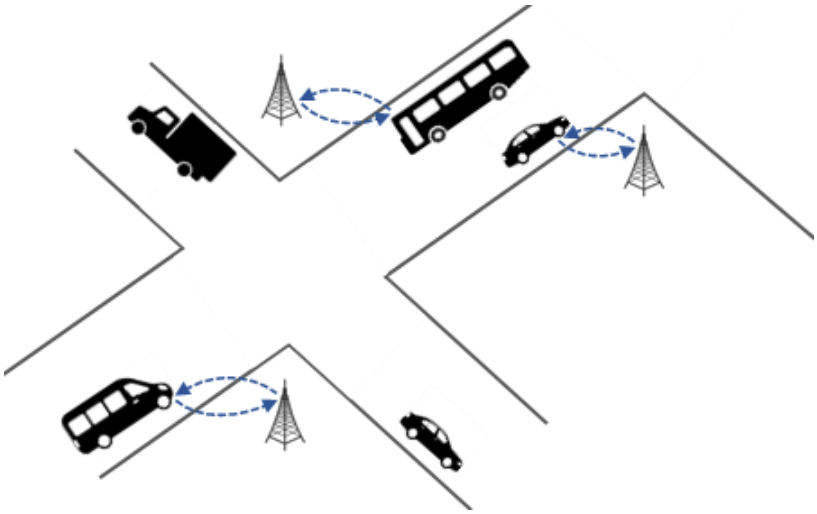


Figure 1-3: Vehicle-to-Infrastructure Architecture

1.3.2.2 Vehicle-to-Vehicle Communication (V2V)

Deployment of RSUs has some difficulties due to certain restrictions such as high cost of implementation, geographical limitations (rural environments, islands, mountains for example), etc [42]. These conditions push vehicles to connect directly with each other without using RSUs, which gives birth to the V2V architecture (figure 1-4).

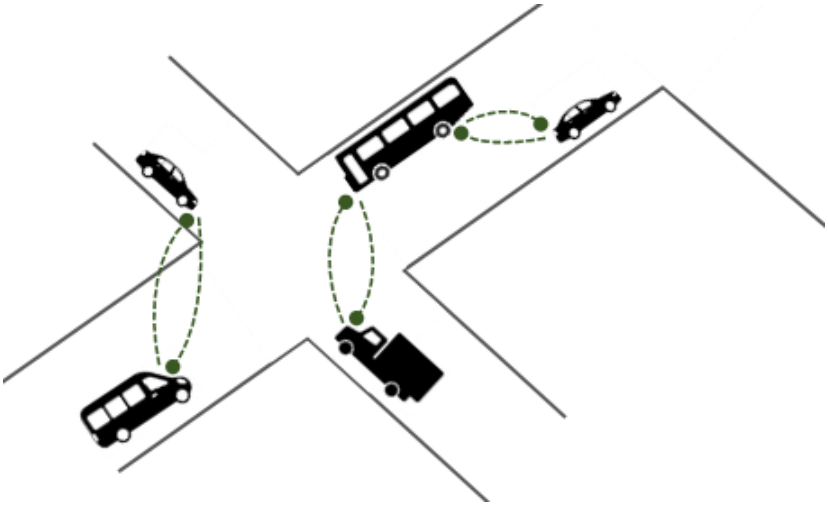


Figure 1-4: Vehicle-to-Vehicle architecture

The vehicle-to-vehicle communication architecture V2V or inter-vehicular communication (IVC) is identical to MANET architectures where the mobile nodes form the communication network without the need for centralized coordination. This architecture is based only on the communication of vehicle OBUs (cars, trucks, emergency vehicles, etc.) and without the presence of RSUs. The vehicles transmit the data packets in a multi-hop way [42]. This type of architecture is required when the RSU is unavailable in the road. In this case, the vehicles communicate with each other to transmit the information to the destination. Generally, this mode of communication is used in scenarios of broadcasting alert messages (emergency brake, slow down, collision, alert of an emergency vehicle approaching, warning of violation of traffic lights, etc.) or for cooperative driving.

The inter-vehicular communications can reduce delay, minimize the cost of transmitting messages, etc. However, the lack of control and assistance of the authorities required for certain sensitive applications such as vehicular safety, traffic zone monitoring, driving assistance, etc. makes the communication less secure. These types of applications cannot be provided by ordinary vehicles contrary to RSUs that can be controlled and managed by authorities in fixed centres such as police stations, gendarmeries, meteorological offices, etc.

1.3.2.3 Hybrid communication

The V2I architecture is based on the communication of vehicles with RSUs according to their coverage areas. However, these ranges are limited which makes the connection with vehicles impossible in many cases. In this case, the use of vehicles as a transmission relay makes it possible to extend these ranges. Access to infrastructures (RSUs) can improve network performance, in which can solve routing problems when transmitting over long distances in V2V architectures [65].

In this case, a combination of the two types of communication V2I and V2V allows to take advantage of both architectures to achieve an interesting hybrid architecture, flexible, low cost compared to V2I and V2V and more suitable for different VANet environments (highways, urban and rural roads). In the literature, the hybrid architecture is called VANet (figure 1-5) [42].

According to this architecture, the vehicular network is composed of two parts: an infrastructure part which represents a set of fixed nodes (RSUs) and a non-infrastructure part which consists of mobile nodes (Vehicles). Vehicles act as routers and/or end nodes, they send and receive data packets to and from other nodes in ad hoc and multi-hop way and/or via fixed nodes (RSUs).

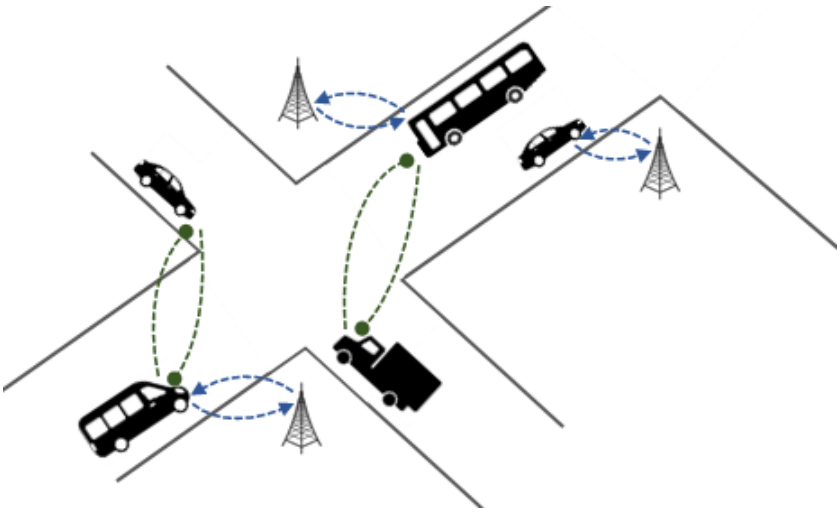


Figure 1-5: Hybrid architecture

1.3.3 Deployment environments

In the road networks, vehicle movement environments can be differentiated by their locations (urban, suburban, rural, mountainous, etc.) and by their means (highway, departmental, national, or municipal roads, etc) [65]. In what follows, a classification of these environments according to the speed and traffic density of vehicles is proposed, in which three environments can be distinguished: urban, highway and rural environments.

1.3.3.1 Urban environment

An urban environment is a road network formed by intersections and breakpoints (traffic lights, stop, give way, etc.). Its mobility model is complex where the density of vehicles is important and vehicles speed is reduced (less than 60 km/h). The presence of buildings causes disturbance waves circulating in the network. The existence of important infrastructures (eg panels) in urban environments makes the installation of wireless network equipment (RSU) very easy.

1.3.3.2 Highway environment

Unlike urban areas, a large diversity of vehicles (truck, car, etc.) use highways with high speed (<130 in France) and form less dense networks [65]. The highway consists of two main roads with two different directions. Each road is composed of many lanes (usually more than two lanes). The highway environment seems less disturbing for radio waves thanks to the absence of obstacles like buildings. However, the high speed of vehicles causes the problem of Doppler spreading, thus, the trucks remain disturbing obstacles for communication and some solutions such as the vision of

the digital cameras.

1.3.3.3 Rural environment

The rural environment consists of less organized roads dedicated to the low speed of movement. In these environments, vehicles form a low-density network.

1.4 Mobility of vehicles and routing of messages in VANet

1.4.1 Mobility models

The movement of vehicles in a highway, urban or rural environments is described by movement patterns known in the literature under "mobility models". The simulation of a vehicular network considers realistic scenarios describing a set of rules that defines the movement pattern of vehicles taking into account different parameters such as velocity, displacement, network density, etc. These scenarios can be modelled using mobility models [42].

Mobility models can be classified into categories according to the movement generated by the vehicle.

1.4.1.1 Random-based mobility patterns

In these models, vehicles movement progresses randomly over time. The vehicles move to any destination and with a random speed between 0 and its maximum value. These models do not represent all vehicle movement scenarios citing for example presence of barriers or geographical restrictions such as buildings, streets, corners, etc. Thus, certain mobility characteristics of vehicles are not shown in this type of models. The speed is changed randomly in time, which is contradictory to the movement of vehicles in a reality where the speed is increased and decreased gradually over time.

1.4.1.2 Geographic map-based mobility models

Geographical restrictions in vehicular environments push research in this area to propose other models to more accurately represent the movements of vehicles. These models are based on rural or urban maps that represent a set of roads, streets, obstacles, buildings, turns, etc.

The simulation of vehicle movement in this class of models is very simple and close to reality because of the use of geographical restrictions. However, other traffic parameters are not taken into consideration. These parameters are useful for improving and predicting vehicle movement such

as the density of the network to distinguish between certain vehicular scenarios, for example, hours of traffic congestion and hours of less traffic.

1.4.1.3 Group-based mobility models

This type of models is based on the collaboration of nodes with the same objective of traffic in a network such as the same destination. In this context, the vehicular network is divided into groups of vehicles, each with its behaviour.

1.4.1.4 Mobility models based on prediction

Prediction-based mobility models are a new category of mobility patterns where vehicles correlate their previous speeds and locations to predict their future movements. Besides, the change of speed and direction is closer to reality, and some vehicle movement parameters such as network density are determined by formulas and prediction rules according to the traffic flow.

1.4.2 Routing protocols in VANet

Routing protocols are used to find communication paths to enable the exchange of data between communicating nodes. In a VANet environment, this process becomes more complex. The fast topology change involves a considerable transfer delay and lost data packets when changing selected paths during the routing process. Many VANet routing protocols are developed to meet different constraints.

Conventional routing protocols in VANet can be classified into three categories: topology-based protocols, geo-based protocols, and cluster-based protocols [42].

1.4.2.1 Topology-based protocols

Initially, this category of protocols has been proposed to MANET networks. Thereafter, the appearance of vehicular networks with some common properties such as mobility, decentralized control, etc. has led researchers to adapt this class of protocols to this new type of networks. However, there are some different features between the two types of networks, for example, the mobility of nodes in VANet is high and requires mobility patterns.

This type of routing protocols is based on using end-to-end paths between the source and destination nodes by selecting topological links between network nodes. Routing protocols in this category can be divided into three sub-categories: proactive, reactive and hybrid.

a. Reactive routing protocols

For this category of protocols, the path to the destination node is established when sending the request. No route maintenance is performed between the nodes when no request sent in the network. Before sending a request, a route discovery process is started immediately to find the path to the destination node. Among the best-known protocols in this category: AODV (Ad hoc On-demand Distance Vector), ACB (Prediction Based Routing), MURU (Multi-hop Routing Protocol for Urban VANet), etc.

AODV (Ad hoc On-demand Distance Vector) [54], the contributions in this thesis are based on the AODV protocol. This protocol is widely used in VANet after it has been proposed to MANET networks. Section 5.4.1.1 in chapter 5 describes the operation principle of this protocol.

b. Proactive routing protocol

Proactive protocols maintain updated information on the entire network all the time. In these protocols, routes are established between all the nodes in the network even if they are not used. Routes are updated using propagated packets in the network periodically without considering network load, data transfer rate, network size, etc [86].

OLSR (Optimized Link State Routing) and RBVT-P (Road-Based Vehicle Traffic Routing Routing) are more well-known protocols in this routing category.

OLSR (Optimized Link State Routing) [44] is a proactive link-state protocol for unicast routing. It was proposed to MANET networks and then successfully adapted to VANet [74]. OLSR uses a new concept of MPR nodes (Multi-Point Relays) that correspond to a subset of its one-hop neighbours. These nodes are chosen to cover all its neighbourhood with two hops.

c. Hybrid routing protocols

Hybrid routing protocols are combinations of reactive and proactive protocols [74]. They aim to improve routing efficiency and scalability by focusing on the benefits of both types of protocols. They can reduce the overhead generated during reactive or proactive routing, minimize the delay caused by reactive protocols when sending data by performing the route discovery process more efficiently [74].

This type of protocol is intended for vehicular networks with a limited number of nodes and low mobility. ZRP and HARP are the most well-known protocols in this type of routing.

ZRP (Zone Routing Protocol) [74], in this protocol, the network is divided into zones. It combines between the intra-zone proactive routing protocol (IARP) inside the zone and the inner-zone reactive routing protocol (IERP) between the different zones.

1.4.2.2 Geographical routing protocols

Geographical routing is a new form of addressing adapted to vehicular networks (on a large scale). It is based on two main elements: location service and geographic transfer process. The location service determines the position of the node and of the destination (using geolocation devices such as GPS) to calculate the path to that destination through the intermediate nodes [38].

In this type of protocol, the nodes maintain no routing tables for the remote nodes (more than one hop) and do not exchange any information about the link state with neighbouring nodes except their positions [72]. To send a request, the node must know the destination position to add it to the header of the packet to allow the intermediate nodes to know its location [38].

Four types of geographic routing protocols can be distinguished [38]: geo-unicast, geo-multicast (geo-anycast), geocast (geo-broadcast) and temporal geocast. The first three types are standardized by the European Telecommunications Standardization Institute (ETSI) [7]. For these categories, addressing is defined by the combination of the destination area and the node identifier. For temporal geocast, the time information is used.

Geo-unicast: the information is sent to a specific node in a destination area when the information arrives at that area.

Geo-anycast: the information is destined for any node in the destination area when the message arrives.

Geocast: The information is intended for all the nodes located in the destination zone at the moment the message arrives there.

Time Geocast: Information is destined to all nodes in the destination area when the message arrives for a certain period.

Research in this area is very active and many protocols are developed to support this type of routing, for example, GPSR (Greedy Perimeter Stateless Routing), GPCR (Greedy Perimeter Coordinator Routing), CAR (Connectivity Aware Routing), etc.

GPSR (Greedy Perimeter Stateless Routing) is a well-known unicast geographic routing protocol [38]. It is more suitable for highway environments, where nodes are uniformly distributed [72]. The routing process in this protocol is based on the position and the address of each node in the network.

1.4.2.3 Routing protocols based on clustering

Routing protocols based on clustering are more appropriate to the network topology formed by clusters, the vehicular network is an example where

(in several scenarios) it is divided into groups of nodes close to each other forming clusters [72].

A node in a cluster can have one of three roles: cluster head (cluster leader), gateway, or member [74]. Each cluster has its cluster head, and the nodes connected to more than one cluster are gateways, while other nodes are members in the cluster. The cluster head maintains information about gateways and members, and inside the cluster, the nodes communicate with each other through direct links (intra-cluster communication). The packet transmission process between nodes is similar to AODV protocol, except that relaying data or control packets is dedicated only to the cluster head and gateways.

In this type of protocols, the configuration of clusters and the choice of cluster head represent a big challenge [72]. The rapid change in network topology, cluster creation and maintenance involve high delay and overhead in the network. Further, some protocols require the existence of RSUs to be able to configure and maintain the clusters in the network [74]. One of the most known clustering protocols in vehicular ad hoc networks is COIN as defined below.

COIN (Clustering for open IVC network) assumes that each node in the network is equipped with GPS equipment. In this protocol, the network is divided into clusters according to the cluster composition process which is based on node mobility, driver behaviour and distance between vehicles [72].

1.5 Standards and technologies of wireless vehicular communication

1.5.1 Access Technologies

In VANet, two types of information exchange systems can be distinguished: intra-vehicular systems composed of internal sensors inside the vehicle to distribute the information within the vehicle, and extra-vehicular systems which aim to exchange information between the vehicle and its environment [65].

1.5.1.1 Intra-vehicular communication systems

Intra-vehicular communication systems or advanced driver assistance systems are composed of sensors, a computing platform and wired network CAN (Control Area Network) or wireless network (Bluetooth, WiFi). Each manufacturer can build its system without ensuring interoperability with vehicles of other competitor brands. ADAS systems (Advanced Driver Assistance System) use two kinds of sensors or information sources: Proprioceptive and exteroceptive sensors [65].

a. Proprioceptive sensors

These sensors provide internal vehicle information such as odometry speed, acceleration, engine condition, brake status, etc. This type of information allows knowing the status and capacity of the vehicle that helps to predict some possible risks and, therefore, find solutions to reduce these risks. For example, the ABS (Anti-lock Braking System) or ASR (Acceleration Slip Regulation) systems use this type of information.

b. Exteroceptive sensors

The exteroceptive sensors perceive the vehicle's navigation environment and provide information about the vehicle itself and the surrounding objects based on their perception of the driving environment. In this context, monocular or stereoscopic vision, laser or radar telemetry, ultrasound, etc. are among the techniques used to provide acknowledge about local and short-range environment of the vehicle.

1.5.1.2 Extra-vehicular communication systems

Extra-vehicular systems are divided into three subsystems according to their use: telecommunication systems, radio broadcasting systems and extra-vehicular networks.

a. Telecommunication systems

Telecommunication systems or mobile cellular networks are dominant in the field of mobile communications. They are particularly used for user comfort applications (Internet on board, videoconferencing, other paid services). Their architecture is based on base stations that control access to support and manage the roaming process. Examples of dominant telecommunication standards in Europe (according to the European Telecommunications Standards Institute (ETSI) [7]): GSM (2G), GPRS (2.5G) and UMTS (3G).

GSM/GPRS (Global System for Mobile communication/General Packett Radio Service) are low-speed radio systems (maximum theoretical rate is 171.2 kbit/s), while transmission of road safety information requires real-time communication with low delay and high data reliability, which cannot be assured by these systems.

UMTS (Universal Mobile Telecommunication System). Data transmission in these systems can theoretically reach transfer rates of 1.92 Mbit/s and 128 kbit/s for mobile equipment with high speed. UMTS is more suited to road traffic safety applications (which generate a large amount of data) than GSM/GPRS. However, these systems provide no guarantee of delay when high-speed vehicles in certain environments such as highways.

b. Radio broadcasting systems

In these systems, the communication is unidirectional, in which the base stations send information to the users that receive the same information at the same time. This section introduces standards for mobile broadcasting: RDS/TMC in addition to the DAB/DMB and DVB-T/DVB-H normalized by the ETSI [7].

RDS/TMC (Radio Data System/Traffic Message Channel). RDS is a radio data broadcasting system responsible for sending transported information in addition to the normal audio signal by frequency modulation through a sub-carrier FM (allows a rate of 1.2 kbit/s). TMC is a European standard for broadcasting digital data on navigation systems. TMC is used nowadays by certain radio stations to alert users of peripheries and highways in France.

DAB/DMB (Digital Audio Broadcasting/Digital Multimedia Broadcasting). DMB is an evolution of DAB, developed and standardized by the ETSI in 2005. DMB allows broadcasting digital radio with multimedia content and mobile TV on small devices such as mobile phones. The DMB standard can support traffic applications (allows a rate of 2,4 Mbit/s). However, it cannot respond to certain requirements of road safety applications such as high speed, low latency, etc.

DVB-T/DVB-H (Digital Video Broadcasting-Terrestrial / Handheld). DVB is a broadcast technology for digital television. DVB-T is a system that transmits voice and video over a compressed MPEG stream. DVB-H is an optimized version of DVB-T. It is suitable for mobile reception and adds to DVB-T a temporal redundancy and high protection of the transmitted flux [65]. DVB-T/DVB-H has a latency of 6s, which is too important for critical contexts of road safety applications. For that, it is used only for comfort applications using I2V communications.

c. Extra-vehicular networks

They propose direct exchanges of information between the entities. They are used for V2V communications and road safety applications [65].

Infrared (IR) is a line-of-sight network through sensors where both communicators must be close to each other. VANet uses this technology in inter-vehicular communications in the context of very short-range in point-to-point. CarTALK and PATH projects are IR applications used for V2V communications in VANet.

WiMAX (Worldwide Interoperability for Microwave Access) is a wireless metropolitan network based on the IEEE 802.16 standard (allows a rate of 70 Mbit/s over a radius of 50km). Mobile WiMAX is the mobile version of this technology. It is adapted to communications of vehicles with moderate speeds with medium and long-range. It meets the needs of

real-time applications such as voice over IP (VoIP) and video on demand.

WiFi (Wireless Fidelity) technology allows an omnidirectional radio coverage of 400 meters with a theoretical speeds of 300 Mbit/s (802.11n) that seems sufficient to maintain a multi-hop connectivity in the highway or urban environments. But, it needs to be improved to be applied in VANet networks. Many research studies have shown that WiFi is not well adapted to vehicular networks.

DSRC (Dedicated Short Range Communication) is a set of technologies dedicated to vehicular communications. It has evolved from IEEE 802.11a to IEEE 802.11p or WAVE (Wireless Access for Vehicular Environments) which is particularly suitable for medium-range and time-sensitive applications. It has good reliability with an error rate of 10-6 to 160 km/h [65]. The DSRC transmit messages of high priority like critical messages related to road safety. WAVE has more features adapted to mobility (such as time of the establishment of shorter connections) that allow sending information to vehicles with high speed.

Table 1.5.1.2 [65] shows characteristics of access technologies in vehicular ad hoc networks. It describes the access technologies according to throughput, maximum range, mobility ability, real-time traffic support, latency and transmission mode.

1.5.2 Vehicular wireless communication standards

1.5.2.1 IEEE (Institute of Electrical and Electronics Engineers) standards

IEEE has extended the 802.11 protocol family by adding the 802.11p protocol. This protocol modifies the physical and MAC layers to adapt them to vehicle networks accordingly to the DSRC band. Besides, a set of protocols for the 1609 family (known as WAVE) has been defined by IEEE to allow wireless access in vehicle networks. The 1609 family is structured in four components (1609.1 to 1609.4) defining the architecture, the communication model, the management structure, the security and the physical access [65].

As shown in figure 1-6 [65], 802.11p and WAVE specify a complete protocol stack. The DSRC/WAVE model uses two batteries; one for road safety applications and one more "classic" for other categories of applications.

	GSM, GPRS	**UMTS**	**RDS/ TMC**	**DAB, DMB**	**Infrared**	**WiMAX**	**WiFi**	**DSRC**
throughput	DL: 60-80, UL: 20-40	DL: 384, UL: 384	RDS: 0.730, UL: n/a	DL: 2400, UL: n/a	DL/UL: 6000	DL/UL: 4500-22000	DL/UL: 54000	DL/UL: 3000-27000
maximum range	35 km	20km (according to the cell)	20km	n/a	1-100 m	50 km	400 m	1000 m
mobility ability	High 300 km/h	High 500 km/h	High 300 km/h	high 150 km/h	Medium 70 km/h	Medium 70 km/h	high 250 km/h	high 300 km/h
real-time traffic support	No	Yes (according to the distance to BS)	No	No	Yes (8 levels of priorities)	Yes (according to the distance to BS)	Yes (EDCA)	Yes
latency	500-700	200-300	10 min	<100	10	50	according to the implementation	<5
transmission mode	V2I	V2I	V2I		V2V, V2I	V2I	V2V, V2I	V2V, V2I

Table 1.1: Characteristics of access technologies in VANet

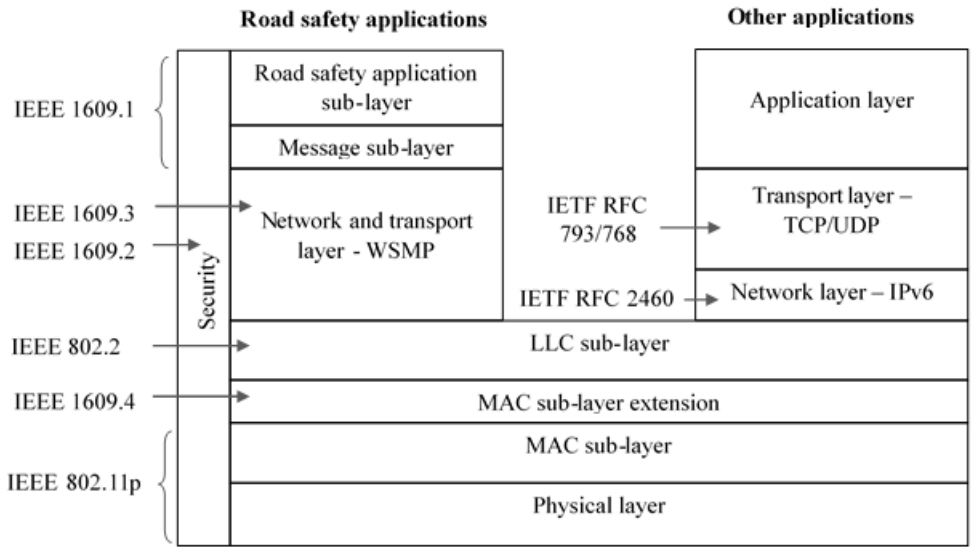


Figure 1-6: DSRC/WAVE model: IEEE 1609

a. IEEE 1609.1

This protocol defines the message formats and data storage mode used by the application layer. It describes 3 components of this layer: the Resource Manager Applications (RMA) which is a remote entity that uses the RM to communicate with the RCP. The Resource Manager (RM) which is a resource manager that relays the message from the RMA to the RCP. Finally, the Resource Command Processor (RCP) that executes the commands given by the RMA.

b. IEEE 1609.2

The IEEE 1609.2 protocol specifies methods for securing management and application messages for the DSRC/WAVE system. It describes the procedures that the vehicle must perform to ensure authenticity, confidentiality, integrity or non-repudiation. Depending on the security services deployed, the message format is different. For example, a transaction message is signed and encrypted while the alert message is only signed.

c. IEEE 1609.3

This protocol defines the WAVE Short Message (WSM) and the associated WAVE Short Message Protocol (WSMP) of exchange to provide network and transport layer functionality for road safety applications. The WSMP is used by road safety applications such as local danger warning (LDW) due to its need for low latency. Furthermore, this protocol defines the WAVE Service Advertisement (WSA) message which is used to announce the availability of DSRC services at a given location, for example, the announcement of the presence of a traffic information service offered by an RSU.

d. IEEE 1609.4 and IEEE 802.11p

The IEEE 802.11p standard defines the physical layer of the DSRC system. The DSRC technology is defined in the frequency band (5.850 GHz - 5.925 GHz). This bandwidth is segmented into 7 channels of 10 MHz each, 1 control channel (CCH) and 6 service channels (SCH), each of which can offer rates from 6 to 27 Mbit/s [65]. The IEEE 1609.4 standard defines the organization, scheduling and the use of these different channels. Its purpose is to define a mechanism allowing several devices (multi-channels) to agree on the same channel at the same time to be able to communicate.

1.5.2.2 ETSI (European Telecommunications Standards Institute) standards

In Europe, the ETSI TC ITS group (Technical Committee Intelligent Transport Systems) [1] works on global standards for cooperative ITS systems, which enhances ongoing ITS proposals of IEEE and ISO (International Organization for Standardization). This group develops standards related to the overall communication architecture, management, security as well as agnostic protocols associated with several layers of communication: physical, network, transport and facility layers. It defines a reference architecture for cooperative V2X communications in VANet, including the support for the IEEE 802.11p standard. In addition, it addresses other topics including specifications to protect vulnerable road users (cyclists and motor cycle riders) as well as for Cooperative Adaptive Cruise Control [63].

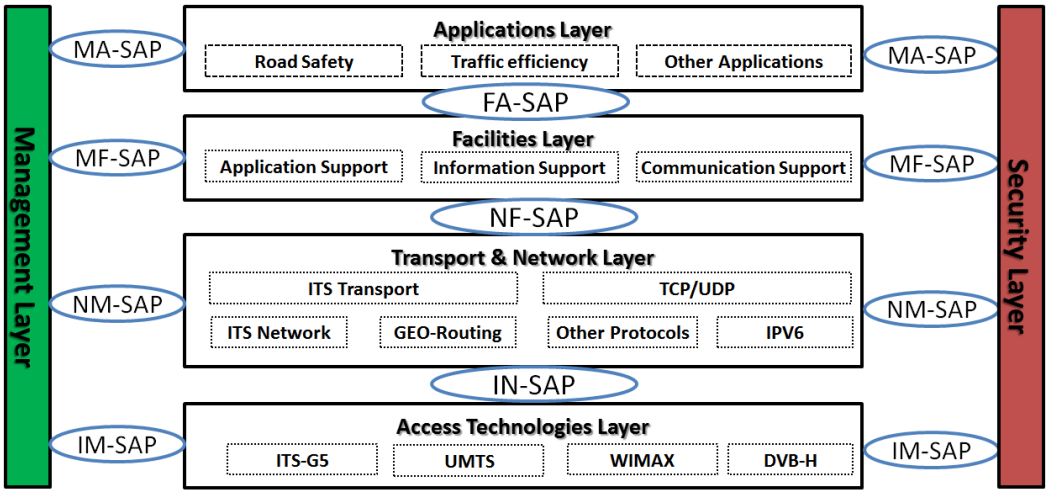
Besides the standards and technologies (used in VANet) mentioned in section 1.5.1, we present the architecture ETSI TC ITS V2X as below:

a. ETSI TC ITS V2X Reference Architecture

Figure 1-7 [63] shows the reference architecture defined by the ETSI TC ITS committee. ETSI TC ITS V2X architecture is based on modified version of IEEE 802.11p at the access layer. It provides new networking capabilities and features at the network layer, which allows using geographical addressing and supporting further communication scenarios, such as multi-hop forwarding. In addition, this architecture integrates a new facilities layer on top to support different types of applications.

b. ETSI Security Layer: TS 103 097

As shown in figure 1-7, the security layer intervenes in several layers: access, networking and facilities layers. ETSI TS 103 097 [17] reuses many features of the existing IEEE 1609.2 security standard. It specifies the main security components, including the certificate format, security headers and security profiles.



SAP: Service Access Point, **FA**: Facilities/Applications, **MA**: Management/Applications, **MF**: Management/Facilities, **NF**: Networking and Transport/Facilities, **NM**: Networking and Transport/Management, **IN**: Physical Interface/Networking and Transport, **IM**: Physical Interface/Management, **UMTS**: Universal Mobile Telecommunications System, **WIMAX**: Worldwide Interoperability for Microwave Access, **DVB-H**: Digital Video Broadcasting – Handheld.

Figure 1-7: ETSI TC ITS V2X Architecture

c. ITSC management entity

The ITSC (Intelligent Transport System Communication) management entity is part of the ETSI reference architecture as illustrated in the informative figure 1-7. It is adjacent to all the ITSC layers, it includes management elements grouped based on their functionalities for each layer, which makes it responsible for connecting different interfaces to the different ITSC layers:

- The MI-interface [14] to the ITSC access layer by using access layer management services.

- The MN-interface [15] to the ITSC networking and transport layer by using networking and transport layer management services.

- The MF-interface [16] to the ITSC facility layer by using facility layer management services.

- The AM-interface to the ITS-S applications by providing management services.

- The MS-interface [18] to the security entity by using security management services.

In addition, the management entity provides management of several services such as networking, communication, ITS application, general congestion control and service advertisement [13].

d. The Collision Risk Warning Application

Many applications are developed based on ETSI standards. The Colli-

sion Risk Warning Application [63] is one of these applications, that consists of the Collision Risk Warning (CRW) road safety application, where a vehicle (or RSU) can detect the risk of collision between two or more vehicles and broadcasts a DENM message (Decentralized Environmental Notification Messages) to all of its neighboring vehicles to avoid traffic congestion and collisions.

1.6 Challenges and issues of VANet

VANets constitute a vast area of research. Several fields have problems that still require a lot of research to address and find solutions. This is intended to enable VANets to communicate and perform their services by guaranteeing a certain level of security and quality of service with a low cost [42].

1.6.1 Routing in VANet

Routing data packets in vehicular networks is among the most important processes. Vehicular network topologies are frequently changed due to high speeds of vehicles, which implies a considerable transfer delay as well as a loss of data packets when changing the chosen paths during the routing process. In this case, the response to routing constraints requires more research to find robust protocols that allow the selection of long-lived paths by ensuring a good quality of routing in terms of transfer delay, bandwidth, packet delivery ratio, overhead, etc.

1.6.2 Vehicular network scalability

Scalability is the ability of a network to efficiently manage a large number of nodes. The density of mobile nodes in VANet varies according to the vehicular environment and the time of traffic, for example, the vehicle's number in cities or highways at the time of large congestion or during an accident can reach hundreds of vehicles. In this case, the dissemination of messages between the nodes will be very congested, which leads to a high rate of packet loss and considerable transmission latency in the network. For that, more researches are needed in this field to find effective solutions to deal with large numbers of nodes in VANet networks.

1.6.3 Computational complexity in VANet

The study of computational complexity in VANet consists in improving the computation operations to find paths between the communicating nodes in the network. The heterogeneity of nodes presents a great challenge in this type of networks. The heterogeneity of interfaces and capacity levels

makes the process of disseminating messages longer in terms of execution time.

The optimization of calculations improves the network performance and provides a good configuration to find efficient paths for optimal data transmission in the network.

1.6.4 Robustness of routing and self-configuration of VANet

The rapid mobility, the high density and the displacement of the nodes in several directions in vehicular networks make the routing of the data packets more difficult. This requires more robustness and self-configuration in the data routing protocols.

Routing robustness is the ability of the network to maintain an acceptable performance in case of network disruption. Robustness makes the network resilient against link/node failure, node removal or attacks [42]. Routing self-configuration in VANet consists of finding solutions for network problems using only their nodes without central control.

Currently, data routing techniques in VANet are projections of MANET network techniques that have relatively different mobile node context regarding speed, direction, pause time and so on. In this case, new realistic sources of inspiration in the context of VANet are needed to find robust and self-organizing solutions for designing new efficient routing protocols without central control.

1.6.5 Security of VANet

Vehicular network security is the set of policies used to control the access to the network by preventing and monitoring unauthorized access, misuse or denial of network resources. The security of VANet involves raising other issues that constitute major challenges. Authenticity, for example, is one of these issues that consists to check the validity of identities of users in the network and protect the nodes against various attacks entering the network using falsified identities. Another challenge related to the security of VANet is the confidentiality of the network which consists of limiting access or imposing restrictions on certain types of information.

Another aspect can be discussed in the security field in VANet is the anonymization of the communication. This anonymity concept consists of masking the real identities of the destination nodes, in which the adversary cannot detect their real identities. Generally, the solutions proposed in anonymity are based on the use of a third party (trusted authority for example) that generates pseudonyms for entities and ensures the relationship between real identities and pseudonyms.

In this thesis, we treat the anonymity of users in VANet, in which we

propose methods and algorithms ensuring that service during the communication.

1.7 Conclusion

The research studies done in the field of wireless vehicular networks aim to improve their operation and make transport safer, more secure, efficient, reliable and more ecological [65]. This type of network is a vast area that has a relation with multiple sub-domains and therefore many challenges and issues facing its evolution. Routing, auto-configuration, self-organization, security, etc. constitute some issues that require more studies and research to propose new solutions guaranteeing a certain level of security and quality of service at a low cost.

In this chapter, we have discussed a general view of wireless vehicular networks by defining general concepts and presenting some known axes in this domain. Four types of applications have been identified according to their contexts of use; traffic safety applications, vehicular authority services, business and entertainment applications and driving improvement applications.

During the execution of an application, nodes can communicate with each other according to one of the communication architectures: vehicle to vehicle, vehicle to infrastructure or hybrid. In this context, different types of messages can be exchanged according to the application used during the communication. Thus, we have differentiated between three deployment environments of this type of networks: urban, rural and motorway. And to differentiate the VANet from other networks, we have cited several characteristics: mobility, velocity of vehicles, energy, heterogeneity of nodes, etc.

Thereafter, we have presented some models of vehicle mobility by defining the general principle of each. Among the aforementioned models: Group-based mobility models, geographic map-based mobility models, random-based mobility models, etc. More access technologies used in the communication of nodes, and IEEE 1906 standards to standardize this communication. Finally, we have finished with challenges and issues that may face the development of wireless vehicular networks.

Chapter 2

SECURITY IN VEHICULAR AD HOC NETWORKS

2.1 Introduction

Security is an important element in networks. With the apparition of new attacks in different kinds of networks especially in VANet, it is mandatory to put a complete system of security to face these attacks. Many attacks can exist in VANet; black hole, warm hole, deny of service, etc. A security system can be defined as a set of methods and algorithms allowing to detect and/or stop malicious activities and malicious nodes.

This thesis is about the management of security in VANet. Some attacks are treated by proposing solutions and developing specific methods and algorithms. the aim behind that is to ensure the anonymity of communication within the vehicular ad hoc networks. Within anonymous networks, the identity of the sender and the receiver are hidden, in which intermediate nodes (attacker or not) cannot detect them.

We develop methods and algorithms in the proposed contributions by respecting several security services to ensure a high level of security. Digital signatures, hash functions, symmetric and asymmetric encryption are used to ensure authenticity, confidentiality, non-repudiation and integrity of messages against many attacks. Besides, we try to keep a good performance with an acceptable quality of service in the network. End-to-end delay, overhead, bandwidth and packet delivery ratio are among the most important parameters that we keep with good values during the communication, which make the proposed solutions acceptable to be implemented in VANet.

Exploiting traffic safety applications in VANet can minimize the number of road accidents. Implementation of that kind of applications needs to take in consideration different constraints of time and configurations of security. vehicles move with high speeds and the topology changes rapidly, which make connections between nodes (vehicles and RSUs) very short.

For that, the proposed security algorithms should be robust enough to meet those requirements and face different attacks. In case of an accident for example, the vehicle (crashed vehicle) or neighbour nodes (vehicles or RSUs) send alert messages to notify rescue services (civil protection and ambulance services for example). That alert messages should be sent within a short time to get the rescue as soon as possible, which is difficult to implement and need more robustness of the implemented algorithm in that kind of networks. Besides, the presence of malicious nodes presents a big problem that should be treated. The proposed protocol uses secure mechanisms to ensure several security services and face some known attacks (Man in The Middle and replay). As a security measure against this kind of attacks, we use digital signatures to ensure the non-repudiation and guarantee the correctness of messages. In that case, the attack will be detected if any alteration has occurred.

This chapter presents a general idea and definitions of some security concepts. It starts by defining security services and quality of service parameters in VANet, giving a classification for some known attacks and talking about some proposed solutions in the security field in VANet. Then, the chapter is concluded.

2.2 Security services and QoS (Quality of Service) parameters in VANet

Ensuring security in VANet risks to degrade the performance of the network and its quality of service. For that, deployment of robust and efficient solutions of security is needed to face different requirements of VANet. The proposed algorithms and methods need to make a compromise between the security against malicious activities and the performance in the network. In this part, we give a general view about different security services and parameters of QoS that should be respected by the proposed security protocol.

2.2.1 Security services

This section defines several services of security in VANet [65].

2.2.1.1 Confidentiality

The word confidential is generally used for data or resources, which means that only authorized parties can have access, and protect them to address some malicious activities and face some attacks of scam and piracy.

Our work is based on this service of security, in which transmitted messages and data must be confidential. In the first contribution (chapter 4),

this service is addressed by encrypting messages using symmetric encryption, in which Diffie-Hellman is used to generate the secret key to ensure its confidentiality. The second contribution (chapter 5) is based on the two forms of encryption: symmetric and asymmetric encryption. Besides, we use the concept of tunnels to ensure more security and confidentiality of the communication.

2.2.1.2 Authenticity

Authenticity is to ensure to the destination that the received messages are sent from the right source. Two types of authentication can be distinguished: authentication of messages and authentication of entities.

Authenticity is an important service of security in VANet as in other systems. Different applications in VANet need to treat confidential information, which is ensured by authentication.

Authentication is a verification process or control that the identity is valid. It requires that the subject provide additional information which must match exactly the identity indicated. The password is the form the most used in the authentication process.

2.2.1.3 Integrity

Integrity consists of protecting data against eventual alteration and modification. In vehicular ad hoc networks, integrity service ensures that messages are transmitted rapidly, without modification, insertion, duplication or repetition.

We ensure integrity by establishing a system of privileges to minimize access to resources. Integrity can be defined according to two axes: Integrity of messages, in which the transmitted information has not been altered and physical integrity, in which materials (dedicated to send messages, collecting information, doing encryption operations, etc.) that have not been altered.

The integrity of messages can be ensured by using hash functions like SHA [33] and MD5 [67], which are mechanisms based on mathematical functions in one way. New technology has been appeared to ensure physical integrity called Temper Proof Device (TPD). It consists of equipment resistant to sabotage and manipulation. TPDs are designed with difficult access to its components and auto-destruction.

2.2.1.4 Non-repudiation

Non-repudiation [32] is a legal concept that is widely used in information security. It can be translated into a method of ensuring that data or messages (for example) cannot be disowned or denied.

Non-repudiation provides proof of the origin of the data and prove its integrity. In VANet, this service has an impact during the communication

especially in security applications when the information must be transmitted correctly and must be recognized. The digital signature is used generally as a mechanism to offer non-repudiation service. It is used in systems when it is crucial to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or messages for example.

Non-repudiation eliminates all risks of malicious nodes to inject additional information in transmitted messages (like in The Man In The Middle attack). Any change in the content of messages can be detected by non-repudiation mechanisms.

2.2.1.5 Availability

Availability of an item to be in a state to perform a required function at a given instant of time or at any instant of time within a given time interval, assuming that the external resources, if required, are provided [35].

In VANet, availability is directly associated with the safety applications, which makes it an important part of security services. It is responsible to manage the functionalities and applications of the network and keep them functional in case of faulty or malicious conditions [73].

2.2.1.6 Access control

Access control can be defined as methods responsible for monitoring the access to the resources (if it is authorized or restricted), identifying authorized or unauthorized entities to connect to the network. This service consists of monitoring the way that the entity interacts within the network. Access control methods can be classified into three categories:

Preventive access control: the mechanisms used for that are deployed to stop unauthorized activities before being launched.

Detective access control: Detective access control mechanisms are used to detect unauthorized activities when they are launched.

Corrective access control: they are deployed to restore the system to a previous normal status before the unauthorized activity occurs.

Access control is used in VANet in different applications. In case of a traffic jam for example, Road Side Units (tricolour light) can intervene to facilitate the displacement of some particular vehicles (police, civil protection, etc). In this case, RSUs allow the exchange of information only with them rather than other vehicles.

2.2.1.7 Flexibility and efficiency

Generally, a flexible system is a system that is able to respond to potential internal or external changes affecting its value delivery, in a timely

and cost-effective manner. In our domain, the value delivery concerns the communication between nodes [8].

Flexibility and efficiency are significant security services in mobile networks, especially in VANet where vehicles move with high speeds and in different directions, which causes changes in the network topology in VANet. Consequently, new connections are established and others are gone. For that, ensuring flexibility and efficiency in VANet is an important factor to ensure good communication between nodes.

2.2.1.8 Traceability and Revocability

Traceability is the capacity to keep track of a given set or type of information to a given degree. In this thesis, the tracked information consists of the identities of vehicles, in which they should be tracked [29]. In the literature, proposed protocols of security ensuring anonymity in VANet (cited in chapter 5 and 6), hide real identities by using pseudo identities. In those solutions, it is mandatory to have specific entities responsible for obtaining real identities. Those entities are responsible for the revocation process or even to block malicious node identities when detecting their undesirable behaviour.

2.2.1.9 Privacy and anonymity

According to the CC standard [34], privacy involves "user protection against discovery and misuse of identity by other users". Anonymity is intrinsically present in the concept of privacy, but it refers to the identity of nodes. According to the CC standard "Anonymity ensures that a user may use a resource or service without disclosing the users identity. The requirements for anonymity provide protection of the user identity. Anonymity is not intended to protect the subject identity, [...] Anonymity requires that other users or subjects are unable to determine the identity of a user bound to a subject or operation".

Our objective in this thesis is to Guaranty the privacy and the anonymity of the communication within VANet. To ensure that, we propose a model inspired by the Invisible Internet Project (I2P), a known protocol in anonymity on the internet.

2.2.2 QoS parameters

As defined by the E.800 recommendation of United Nations Consultative Committee for International Telephony and Telegraphy [19], the quality of service (QoS) means « the totality of characteristics of a telecommunications service that bear on its ability to satisfy stated and implied needs of the user of the service ».

Services performed on the network can be described by different characteristics named as QoS characteristics such as speed, accuracy, dependability, availability, reliability and simplicity.

Evaluation of QoS in the network is based on a set of parameter values. In wired networks, it can be measured by two parameters: throughput and delay which are not sufficient in the vehicular ad hoc networks [56]. Measurement of QoS in VANet is difficult due to the vehicle mobility of vehicles, rapid change of network topology, transmission delay, scalability and density of vehicles in such environments like urban places.

Generally, data traffic in vehicular ad hoc networks can be classified in two kinds: real-time traffic such as safety messages and video/audio streaming and non-real-time traffic such as information data about weather, traffic road, e-map, etc [56]. Real-time traffic is a critical type of data, in which packets must be transmitted with accuracy and at the right time, for example, safety messages can become useless when they are received with a long delay.

In vehicular environments, communication between nodes is critical. In such scenarios, vehicles congestion, packets collisions and delay can be highly increased, which make maintenance of QoS in this type of networks a crucial task.

In this thesis, we aim to develop a secure protocol to encrypt the end-to-end communication and ensure the anonymity of nodes (vehicles and RSUs) in VANet. Implementing such security protocols in that kind of networks risk to degrade the performance and influence its QoS due to their long treatment during the communication. Even though, QoS parameters should have acceptable values to ensure the good providence of services in the network. This performance is evaluated by calculating some specific QoS parameters as detailed below.

In VANet, different applications have been developed to provide different services. Security and traffic safety applications can be classified as the most critical category of applications. That kind of applications are responsible for the security of drivers and passengers or even save their lives by helping to make good decisions in dangerous situations. These applications need to be executed perfectly during the communication to intervene effectively at the right time, which need a high level of performance in the network.

Security and traffic safety applications can exchange alert messages to rapidly diffuse the information in the network, which needs to send and receive them in a short time. In the proposed protocol, we try to keep having good values of the QoS parameters to meet the requirements of that kind of applications in VANet. Among important parameters that are treated in our results, the end-to-end delay that is the time needed to a packet to reach the destination. This parameter is kept with good values to allow rapid communication and exchange of important messages.

We have treated as a QoS parameter also, Packet Delivery Ratio (PDR)

which shows the ratio of the successful delivery packets during the communication (as defined later in this section). This parameter should have high values to ensure the deliverance of messages sent for security and safety purposes for example. Another QoS parameter is evaluated: the overhead, which can be generated by useless messages during the communication. It is an important factor that needs to be treated carefully and keep it with acceptable values. This parameter becomes more critical in vehicular ad hoc networks referring to wireless communication, high mobility of vehicles, traffic jam, etc.

This section defines several parameters that can be used and calculated to measure the performance and the quality of services in vehicular ad hoc networks as below:

2.2.2.1 End-to-End delay

End-to-End delay refers to the time needed for a packet to be transmitted in a network from source to destination nodes [6]. It is calculated (D_{e-e}) the equation 2.1 as below:

$$D_{e-e} = N * (D_{tran} + D_{prop} + D_{proc} + D_{que}) \tag{2.1}$$

// N: number of links;

//$D_{tran}, D_{prop}, D_{proc}, D_{que}$: transmission, propagation, processing, queuing delays.

2.2.2.2 Bandwidth

The bandwidth is an important factor which influences the system performance. It represents the maximum amount of data that can travel through a channel in the network [61]. A good bandwidth sharing between network nodes can improve performance in the network and thus provide such level of quality of services. Particularly, VANet environments require a large bandwidth to deal with some critical characteristics of this type of networks such as high mobility and a large number of vehicles in some scenarios of traffic jam.

2.2.2.3 Packet Delivery Ratio (PDR)

It is the ratio of packets received successfully compared to the emitted packets from the source node [69]. This parameter can reach 100% ratio when all issued packets are received by the destination nodes in the desired time delay. Usually in VANet environments, PDR has low values because of some conditions of high nodes mobility and speed.

PDR can be calculated as the percentage of packets delivered to their destinations relative to packets transmitted in the network (12 of chapter5). It is computed in the equation 2.2 as:

$$PDR = 100 * \frac{\sum received\_data\_packets}{\sum sent\_data\_packets}(in\%) \tag{2.2}$$

2.2.2.4 Reachability

This parameter defines the percentage of nodes receiving the sending messages compared to the total nodes number in a broadcast context [61].

2.2.2.5 Overhead

The overhead can be generated by useless messages in the network causing errors, collisions, delay, etc. Generally, control messages such as periodic beacons in vehicular ad hoc network are considered as overhead. The overhead (equation 2.3) represents the number of control packets broadcasted in the network divided by the total number of data packets received [39].

$$Overhead = \frac{\sum sent\_control\_packets}{\sum sent\_data\_packets} \tag{2.3}$$

Propagation speed: in wireless networks, the time taken in the wireless medium during packets transmission in wireless networks is defined by propagation speed. In the broadcasting context, this parameter is calculated by the number of hops required to diffuse data from a source to destination nodes.

2.2.2.6 Throughput

The throughput represents the rate of data received by a node or passing through it in the network [61].

2.2.2.7 Scalability

Scalability means the ability of the network to have a large number of nodes and to be efficiently adaptable in different simulations and scenarios [61]. This parameter is important in VANet. The number of vehicles can rise in many scenarios, especially when the mobility of vehicles is defined based on movement patterns (section 1.4.1 in chapter 1) where routes and itineraries are defined and limited according to specified directions. This limitation in directions and routes can cause congestion in some environments, mostly urban models where intersections and tight roads, for example, can be existed.

In that kind of environments in VANet, large traffic can be generated and the number of vehicles can rise quickly, which needs for the network to be scalable and be able to support a large number of nodes without influencing any services or degrading the performance.

2.2.2.8 Success rate

We can consider the success rate as a parameter to evaluate QoS. This parameter is defined as the number of successfully executed processes during the different simulations and scenarios by achieving the desired goal [61].

2.2.2.9 Jitter

It defines the end-to-end delay variation during packets transmission between the source and destination nodes [57]. Jitter can be generated when packets take different paths with different characteristics in the network. This parameter is very important in real-time applications such as video conference, VoIP, etc. The jitter formula is defined in the equation 2.4 as following:

$$jitter(P_{s,d}) = \sum_{l_i} jitter(l_i) \tag{2.4}$$

Where $P_{s,d}$ is the path between nodes s and d,
and jitter (l_i) is the jitter incurred at link l_i
for all $l_i \in l_0, ..., l_n \mid p_{s,d} =< l_0, ..., l_n >$

2.2.2.10 Link expiration time

Link expiration time represents the duration of a link by which the communication is established between two nodes in the network [57]. The link expiration time can be defined as the link lifetime. Measurement of this parameter in such networks like VANet where nodes move with high mobility is considered as an important QoS criterion. This parameter is critical in VANet, where vehicles have rapid mobility that influences the duration of communication links between nodes. Responding to that issue, the proposed protocol is based on creating tunnels and maintain them permanently to maintain links between communicant nodes during all the communication.

Link expiration time can be defined as the minimum lifetime value calculated for different links in the path between source and destination nodes (equation 2.5).

$$let(P_{s,d}) = \min(let(l_0), ..., let(l_n)) \tag{2.5}$$

Where $let(l_i)$ is the LET of link l_i for all
$l_i \in l_0, ..., l_n \mid p_{s,d} =< l_0, ..., l_n >$

2.2.2.11 Network lifetime

The network lifetime can be considered as a parameter of QoS when the vehicle energy is taken into account. Electric vehicle technology is a current topic of research which requires studies and adaptable protocols to deal with vehicle energy constraint. Network lifetime means the duration of the

network, in which all nodes can communicate between each other without failure nodes caused by energy for example.

2.3 Attacks in vehicular ad hoc networks

With the development of vehicular ad hoc network and its use in our daily life, many attacks have appeared [71].These attacks can be classified according to multiple criteria; the way to attack, the number of attackers and the malicious activity occurred during the attack.

This section presents a proposed classification of attacks in VANet. Eight types of attacks are defined in this classification:

2.3.1 Sybil attack

Sybil attack [58] is the most dangerous and difficult attacks to detect in VANet. This type of attack is performed when a node uses multiple identities in the network by theft or forging them. The attacker can behave as a set of nodes belonging to the network. Therefore, other nodes deal with the received messages from the attacker as they are sent by more than one vehicle. By using geographical routing, the attacker can broadcast incorrect information about its position in the network, which can show events in different false positions in the network. Thus, the attacker node can make the network behaviour according to its goals such as changing road of a set of vehicles to another planned road referring to its purposes. Another example which refers to this category of attacks: Node Impersonation Attack; in accident situations, a malicious vehicle belonging to the traffic accident behaves as a moving vehicle by changing its identity. In this case, this node can send incorrect information about the road conditions to the network.

2.3.2 Denial-of-service attack (DoS)

This type of attacks concerns directly the availability of resources in the network in the way to make valid activities of a system unavailable [68]. The attacker sends high-frequency signals (more requests than the system can handle) to jam the whole communication channel between nodes. The main goal of the attacker is to prevent other nodes to communicate with each other in order not to access network services such as sending and receiving safety or non-safety messages. Generally, Denial of Service attacks are launched near RSUs to prevent communication between vehicles and RSUs.

Another form of DoS attack which is more sever is performed in a distributed way (Distributed DoS), in which the attacker lunches attacks

from different locations [71]. Figure 2-1 demonstrates an example of DDoS attack from vehicles C, D and G on F.

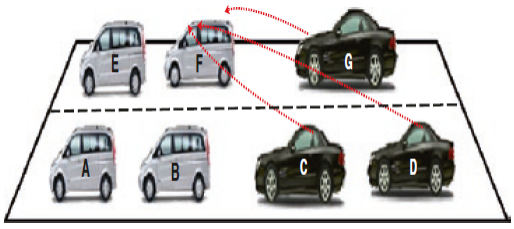


Figure 2-1: Example of DDoS attack in VANet

Various types of DoS can be distinguished in VANet, in which the malicious node launches the attack in different ways as follows:

2.3.2.1 JellyFish attack

This attack [36] concerns forwarded packets, in which the attacker can delay, disorder or periodically drop packets. The results of this attack on the network could be critical and lead to severe consequences such as decreasing performance of the end-to-end communication between nodes.

2.3.2.2 Intelligent cheater attack

The malicious node launches this attack from time to time which makes this attack more specific. The goal of the attacker is to appear as a normal node for most of the time, which makes its detection difficult [71].

2.3.2.3 Flooding attack

The goal of this attack is to make resources (CPU, bandwidth, etc.) for legitimate nodes unavailable [71]. Flooding attacks can be distinguished by two forms: data flooding attack, in which the attacker generates and sends useless data packets to all the network nodes by knowing all possible routes to reach them. For the second form; route request flooding attack, the attacker sends requests control packets to inexistent nodes in the network.

2.3.2.4 Jamming attack

In this case, the attacker generates radio frequency signals on the transmission medium to occupy it and prevent legitimate nodes to perform communication [71].

2.3.3 Blackhole attack

Among known dangerous attacks in MANET and VANet, Blackhole attack [78] represents a serious threat in the network. The malicious node

performs the attack in best conditions, it has to receive a large number of packets as mash as possible which are supposed to be forwarded in the network. To achieve that, the attacker has to be in the appropriate position (where circulate a large number of vehicles) and send false information about existing routes in the network. This information is sent to show that the attacker has the best routes in the network to push other vehicles to choose routes passing through it. In this case, the attacker discards all packets forwarded through it intentionally which causes packet losses and prevent target nodes to receive the packets emitted to them. Simulations and analysis show that such parameters as end-to-end delay, throughput, PDR, etc are affected by the blackhole attack.

When more than one attacker is present in the network, they can create their network, in which they can communicate with each other by exchanging received packets from other nodes such as route requests.

2.3.4 Wormhole attack

Wormhole or tunnelling attack [62] is realized by a collaboration of two or more attackers strategically placed in the network to exchange messages between them. The main aim of this attack is to collect and manipulate a large amount of traffic by falsifying the logical topology of the network. The attack is started when an attacker receives packets from the network and encapsulate them to other attackers by making tunnelled packet arriving with better metrics without taking into account the tunnelled distances, then, replaying the packets at the other end of the network. Encapsulating tunnelled packets transmitted between attackers makes such apart nodes believe that they have short distances between them, which shows that the malicious nodes have best paths, and therefore, deceived nodes choose these paths as the best to establish their communications.

2.3.5 Bogus information attack

Communications in vehicular ad hoc networks are based on information sent between nodes. The content of messages sent in the network is important, in which in such situations, diffused information about accident or traffic conditions can change driver is behaviour by taking other roads for example.

In this attack, the malicious node exploits the impact of message contents by disseminating fake information between nodes in the network [71]. The malicious node generates and spreads false messages about its environment. Its goal is to manipulate vehicles and change their behaviour according to their malicious intent. Two forms can be distinguished for the bogus information attack:

2.3.5.1 False position information

In safety applications in VANet, position information of some events such as accidents and road conditions must be correct. Any kind of position falsification will have a big impact on the behaviour of vehicles.

This attack consists of disseminating false position information by the attacker to the network, which causes critical problems in security, reliability and performance [71].

2.3.5.2 Sensor tampering

In this type of attacks, the attacker deceives sensors of OBU system to make its vehicle as it is in real situations such as congestion, face to climate change, etc [71]. Its goal is to push its vehicle to generate false information about fake conditions captured by deceived sensors. For example, the attacker can generate messages about a traffic jam situation by braking within short times making the OBU system deals as in real situations of traffic jam. Therefore, the attacker broadcasts safety messages about this condition over the network.

Two sub-types can be defined from Sensor tampering attack: illusion attack and GPS spoofing attack.

a. Illusion attack

Illusion attack is realized through two steps: Firstly, the attacker must achieve a suitable situation of traffic according to their messages which will be disseminated later. Secondly, sending false but valid messages among vehicles in the target network. For example, the attacker can send false emergency brake warning messages to some vehicles. By receiving these messages, the targeted vehicles believe the messages and decelerate their speed or switch to another lane. In such cases, malfunction of sensors causes an illusion attack by generating false messages based on data outputs of sensors.

b. GPS spoofing attack

GPS spoofing or tunnel attack consists of generating false geographical positions by malicious vehicles. The attacker uses GPS simulators which generate signals stronger than original GPS signals sent by satellite to send false geographical position information to other vehicles in the network. A victim vehicle accepts the received position and behaves as it is its real position.

2.3.6 Replay attack

The idea is that the attacker gathers and stores information about such conditions as traffic events, climate changes, etc, then reuse this information later [70]. Taking a situation of an accident, for example, vehicles

near the accident generate messages to inform distant nodes to take other roads to avoid the congestion. In this case, the attacker vehicle stores these original messages to resend it later in the network to deceive nodes about an accident situation in a wrong time.

We treat this attack in the first contribution (chapter 4), in which the attacker is supposed to resend the transmitted messages another time. The proposed solution for that is to include the sending time in the transmitted messages then sign them before the transmission.

2.3.7 Man-In-The-Middle (MITM) attack

In MITM [82], the attacker aims to intercept communications between two parties without detecting this interception. The attacker can change the content of exchanged messages according to its purposes. The MITM is generally launched in the Diffie-Hellman key exchange method when no authentication.

Our first contribution is about a proposed secure model based on Diffie-Hellman method in VANet. In this model, we propose a solution to face the MITM attack that is supposed to be launched in several stages and different times.

2.3.8 Passive eavesdropping attack

Passive eavesdropping attack or also known as traffic analysis attack or stealth attack presents a first step before the real attack [71]. It is usually used before implementing blackhole and DoS attacks.

In this attack, the attacker gathers information about the whole communications of the network by using the characteristics of the wireless medium. The attacker monitors and listens to the established communication before starting the attack, which takes more time but represents a sophisticated way to attack. In the proposed contributions, this attack is faced by encrypting messages in the way to make it difficult for an attacker to eavesdrop the communication. Signature mechanisms, encryption algorithms and tunnels are used to anonymize the communication, which makes the attacker unable to know neither the message content nor the identities of communicant nodes.

2.4 Solutions for some attacks in VANet

This section presents a set of proposed solutions in the literature for certain attacks in VANet. We consider Sybil and Deny of Service attacks, in which we present two solutions for each.

2.4.1 Proposed solutions for Sybil attack

In [47], the authors present a method for defending against multi-source Sybil attacks in VANet. They propose an event-based reputation system (EBRS) which detects illegitimate identities in communication and stop the spread of false messages in Sybil attack. EBRS authenticates each vehicle with a local certificate with its local RSU and assigns each event with a dynamic reputation value and a trusted value. This event can be a traffic accident, traffic jam, etc. The event reputation value can be defined as the number of times that the vehicle has sensed the event. The event trusted value is the number of distinct vehicles that have detected the event.

Before starting communication, each vehicle computes its public key (PK) and pseudonym (PID) to send them to the trusted authority (TA) through its local RSU to validate its values. In case when PK and PID are not valid, the local certificate of the vehicle cannot be generated. Otherwise, the TA confirms to the RSU to proceed in the local certificate generation process, in which this local certificate is stored by the RSU in its certificates list LC.

After receiving its local certificate, the vehicle Ve can establish communication with other nodes. When an event has occurred, the vehicle creates an event entry in its events list EL and broadcasts warning messages including its pseudonym and encrypted local certificate to its neighbours. The receiver vehicle has to verify the legitimacy of Ve by sending the pseudonym and the certificate to the RSU. If Ve is legitimate, the RSU sends a validation about the concerned vehicle Ve legitimacy to Vr.

After receiving confirmation information from the local RSU, Vr has to verify the integrity of the warning message sent by Ve. If it is not legitimate the vehicle Vr will ignore it. Otherwise, Vr updates the event trusted and reputation values of the occurred event in its EL or creates a new event entry if it did not exist before. The driver could not be notified about the occurred event until both event reputation and trusted values reach its corresponding thresholds, and therefore, the vehicle rebroadcasts the warning message to its neighbours. In this case, the driver can do some actions to deal with the event. In case when an attacker uses false pseudonym and session key, the local RSU broadcasts warning messages about a Sybil attack and reports to TA to trace the real identity of this attacker.

The kNN classification method [49] is proposed to detect Sybil attack in VANet. It is based on evaluating driven patterns of vehicles to classify them into two groups: malicious and benign vehicles. Generally, benign vehicles have similar driven patterns compared to malicious vehicles which are erratic. The driven pattern of vehicles within a period of time is described by a Driven Pattern Matrix (DPM).

A vector of five parameters $(x_{i1}, x_{i2}, ..., x_{i5})$ defines the driven pattern of a vehicle at certain time: time, location, velocity, acceleration and the

variation of acceleration of vehicle. Consequently, the driven pattern of vehicle within a time period (from t_1 to tn) is described by a matrix (DPM), in which each line in this matrix is about a different time as shown in matrix 2.4.1:

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\ . & . & . & . & . \\ x_{n1} & x_{n2} & x_{n3} & x_{n4} & x_{n5} \end{bmatrix}$$

Figure 2-2: matrix (DPM) of the driven pattern of a vehicle

Five eigenvalues can be calculated for each vehicle within a period according to the recorded parameters. In the proposed method, the authors choose the two biggest values to represent the driven pattern of the vehicle within (t1, t2), which allows to represent it on a two-dimensional surface with the two biggest eigenvalues axes.

To classify vehicles according to their driven patterns, the proposed method calculates the difference between vehicle driven patterns using the two eigenvalues values. This difference between two vehicles V_i and V_j is defined by the Minkowski distance between their eigenvalues vectors V_i(1, 2) and V_j (1,2) respectively.

By this classification, vehicles in the network can be classified in two groups, in which each vehicle is labelled with -1 or 1 to define its class; malicious or benign respectively. The classification of an arriving vehicle is based on the labels of the k (1, 3, 5 ...) nearest neighbours that represent vehicles that have the smaller driven pattern distances to the arriving vehicle.

2.4.2 Proposed solutions for DoS

Authors propose a trust-based framework for reliable data delivery and DoS defence in VANet (TFDD) [55] which defines a set of modules and parameters used by each vehicle. Each module in TFDD framework has specific functionalities in the detection process of DoS and DDoS attack.

The parameters in this framework are used to describe some characteristics of the exchanged messages in the sense of trust and quality of these messages during the communication with neighbours, and each vehicle uses the suitable module to calculate periodically these parameters and compared them to predefined thresholds in order to supervise the communication and detect DoS attack if it has occurred. The determination of thresholds values is very important, in which a suitable value can increase the detection rate of DoS attack.

According to the proposed method, each vehicle assigns to each one of its neighbours an Honesty weight (H) value initialized by the value (1).

When a vehicle V_i senses that its neighbour V_j sends packets to it more than a predefined threshold, V_i uses Intrusion Detection Module (IDM) to decrease H value of V_j locally. By using Delayed Verification Module (DVM), the vehicle V_i calculates the trust weight (TM) parameter for every message received from V_j and combines periodically the calculated H parameter with the Quality weight (Q) parameter which represents the quality of the received packets from V_j. This combination is to determine the DoS weight (DW) about V_j.

After calculating the aforementioned parameters, the Decision Module (DM) combines DW and TM values to calculate the t_n value, which is a trust value between the two vehicles V_i and V_j. The main two parameters in TFDD framework are DW and TN, in which a DoS attack can be detected according to their values when they are not included within the interval of the two predefined thresholds. In this case, the vehicle V_j can be placed in the local blacklist of V_i and consequently in the global blacklist at the trusted authority which can prevent it from network operations.

Soryal et al. [75] present a solution to detect DoS attacks in IEEE 802.11 DCF. They use a Markov chain model to generate an adaptive threshold, which is the maximum rate of messages any node can send over time tacking into account the number of other nodes. If a node notices the number of CTS (Clear-to-Send) messages received per second for a destination address is above the threshold, the sender of the CTS messages is tagged as an attacker. The simulation results show that the threshold-based approach can detect the attacker(s) efficiently. However, the solution is not scalable because it is designed for nodes communicating with a single hotspot which acts the same as an RSU.

2.5 Conclusion

This chapter presents an overview about the security field in VANet, in which it presents security services that need to be provided in that kind of networks. In this theses, the proposed solutions are based on several services namely authenticity, confidentiality, non-repudiation, integrity, access control, availability, flexibility, efficiency, traceability, revocability, privacy and anonymity.

After that, we present different quality of service parameters that should be respected to keep an acceptable level of performance in the network, namely: end-to-end delay, overhead, bandwidth, packet delivery ratio and so on. Then we classify different attacks that can be launched in VANet according to some specific parameters to a set of types. Eight types of attacks can be defined: Sybil, Deny of Service, Blackhole, Wormhole, Bogus information, Replay, Man-In-The-Middle and Passive eavesdropping attacks.

Research in the security field in vehicular ad hoc networks is very ad-

vanced. A large area of works takes place to propose solutions against different attacks. Presenting some of these works, we analyze some proposed solutions against the Sybil and deny of service attacks in VANet.

Chapter 3

ANONYMITY IN VANET USING THE INVISIBLE INTERNET PROJECT (I2P)

3.1 Introduction

Anonymizing communication became a substantial issue to secure networks from potential attacks. Several studies are conducted in this field aiming to ensure secure and anonymous communication. In vehicular ad hoc network (VANet), this concept is used in different application areas like military domains, in which hiding destinations identities is crucial.

In this thesis, we propose a model of security to ensure anonymity in vehicular ad-hoc network. This model is inspired from the Invisible Internet Project (I2P), in which some of I2P mechanisms and algorithms are used in VANet. I2P is designed to anonymize the communication on the internet, which is different from VANet in terms of mobility and connectivity of nodes. Critical characteristics of VANet (high mobility, fast topology change, etc) should be considered when proposing new solutions, especially for security algorithms that affect performances of the network. To this aim, we adapt the mechanisms and methods of the I2P protocol to respond to the critical characteristics of VANet.

I2P is a new open-source internet technology that provides internet services anonymously. It offers many applications including anonymous emailing, web hosting and file sharing. In fact, I2P appeared as a new technology after the Tor [3]. Recent studies show that I2P is faster than Tor and offers a high level of security and anonymity for some applications within Darknet.

In fact, I2P is a recent protocol and it is currently being developed. Indeed, many versions have been developed to enhance its operation and ensure a high level of security. I2P has been proposed as a modification to Freenet [9] in February 2003, then it grows into its platform called

"anonCommFramework" [10] in April 2003 to become the Invisible Internet Project in July 2003. That version has been updated until today [10]. I2P relies on robust algorithms and mechanisms, which make it quicker and more secure than other anonymous internet technologies. For this reason, we have chosen I2P as a reference model to achieve a high level of security and anonymity, by keeping a good quality of service and acceptable performance in the network.

The proposed model is based on some algorithms and mechanisms of I2P. Indeed, tunnels and I2P encryption algorithms based on digital signatures and authentication mechanisms are used. The aim is to make the proposed protocol more secure by ensuring anonymity, integrity, non-repudiation and confidentiality.

We continue this chapter by detailing the anonymity aspect and present some solutions and models used on the internet to provide that service. Afterwards, we justify our choice to use the I2P protocol as a reference model in anonymity for VANet. The followed sections define the I2P and detail the method of message exchange used in that protocol. Then, we explain the message processing inside tunnels and the role of the network database during the communication. Thereafter, we present different encryption levels and we distinguish between characteristics of the internet where I2P is originally implemented and vehicle ad hoc networks. Finally, the chapter is concluded.

3.2 Anonymity as a security issue in VANet

Our work in this thesis concerns the security of communication in VANet. More precisely, we aim to make this communication secure and anonymous by encrypting messages and hiding the identities of nodes. Anonymizing communication within VANet and ensure its security against several attacks can be achieved by using a secure model having a high level of security and anonymity. Many models and protocols are developed for that aim for wired networks like the Internet. In this thesis, we aim to propose a secure model inspired by one of the most known protocols in this field and adapt it to VANet.

3.2.1 Anonymity in Internet

In fact, research about anonymity in wired networks is more developed than vehicular ad hoc networks. Various protocols are developed where Freenet [9], Tor [3] and I2P [20] are the most known secure tools of anonymity in internet.

3.2.1.1 Freenet

Freenet can be defined as a distributed anonymous information storage and retrieval system on the Internet. Communications between Freenet nodes are encrypted and routed through other nodes to make it difficult to determine who requesting the information and disclose its content [51].

3.2.1.2 Tor (The Onion Routing)

"The Tor network is a group of volunteer-operated servers that allows people to improve their privacy and security on the Internet. Tor's users employ this network by connecting through series of virtual tunnels rather than making a direct connection, thus allowing both organizations and individuals to share information over public networks without compromising their privacy" [28].

3.2.1.3 I2P (Invisible Internet Project)

I2P is an anonymous network where the nodes exchange messages anonymously and securely. That exchange is end-to-end encrypted, it relies on virtual tunnels as well as other mechanisms to ensure the anonymity and the security of the communication.

3.2.2 Choosing I2P as a reference model in anonymity for VANet

VANet is a form of mobile ad hoc networks, where communications are established between a group of vehicles and RSUs within the range of each other. VANets have critical characteristics where the vehicles can move with high speeds and the topology can change rapidly. Therefore, providing security and anonymity within VANet requires the application of robust models of security and anonymity that meets these characteristics.

Choosing the best protocol for VANet implies the application of a model with a high level of security, anonymity and performance, in addition to having acceptable characteristics that can be applied in VANet.

For instance, the Tor protocol is used to browse the "normal" web without revealing the IP address. It allows the anonymization of the TCP connections origin. In contrast, I2P uses both TCP and UDP transport protocols, which are more suitable for our work. The final objective of this work is to provide security and anonymity of the communication in VANet regardless which transport protocol is used. Tor uses bidirectional circuits to ensure the anonymity of the communication. In I2P, the security and the anonymity of the connection between the source and destination are reinforced by using unidirectional tunnels, in which four tunnels are required to establish the communication between two nodes. Moreover, the tunnels in I2P have a short lifetime. Thus, the number of samples that

an attacker can use to launch an active attack is decreased, unlike circuits in Tor, which generally have a long lifetime.

Tor provides non-sufficient anonymization of the flow. It anonymizes the source of web browsing or instant messaging sessions. During the communication, the application transmits additional information about the identity of the person, which requires to use special browsers like web browsers based on Firefox or Tor Browser, as well as other applications specially modified to preserve the anonymity of their users, which is different from I2P that provides the anonymization of the source and the destination without sending additional information.

For Freenet, this protocol is designed as a distributed data store to publish documents (web pages, PDF, images, videos, etc) anonymously and resistant to the censorship, which is not the objective in this thesis. We aim to secure and anonymize the communication between the nodes, which exactly provided by the I2P protocol.

Finally, after a general comparison between I2P, Tor and Freenet, I2P seems to be the most suitable protocol to be adapted in VANet. To this reason, the proposed algorithms will rely on I2P as a reference model to develop new mechanisms of security and anonymity. I2P uses robust mechanisms and strong algorithms to reinforce the security and the anonymity of the communication compared to Tor and Freenet. The chapter below presents the I2P protocol in detail to discover the used algorithms and methods to inspire the new protocol.

3.3 Definition of I2P

The Invisible Internet Project (IIP or I2P) [20] is a project proposed in 2003. It aims to secure the communication and anonymize the users within a subnetwork on the internet. It is intended to protect communication against oversight and monitoring of third parties such as internet service providers.

Many types of applications in I2P networks are developed such as email, file-sharing, anonymous web-hosting, etc. I2P is a peer-to-peer-based low latency anonymous communication network.

I2P exposes a sublayer in the stack protocol, it is based on TCP, UDP and IP protocols. I2P is peer-to-peer-based low latency anonymous communication network aiming to ensure integrity, security, anonymity, scalability and self-organizing in the system. In the I2P protocol conception, no central point in the network, in which pressure can be exerted to compromise the security or anonymity of the system.

3.4 Protocol stack of I2P

This section presents I2P in the form of a set of layers structured in a protocol stack [22]. Each layer provides protocols responsible for extra capabilities and services. Figure 3-1 shows the protocol stack of layers of I2P protocol.

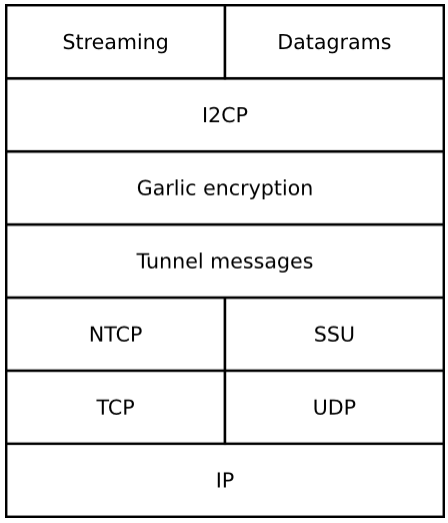


Figure 3-1: Protocol stack of I2P

3.4.1 Internet Protocol (IP) layer

this layer is responsible for addressing hosts and routing packets on the internet. It is independent of the I2P protocol, in which I2P messages can be routed based on the used routing protocol on the internet.

3.4.2 Transport layer

This layer provides host-to-host communication services for applications. Its role is the establishment of temporary communications between applications to route data between them. It provides services such as connection-oriented communication, multiplexing, flow control and reliability. This layer provides UDP and TCP protocols [43] as defined below:

UDP (User Datagram Protocol) provide basic functions for transmission with no reliability.

TCP (Transmission Control Protocol), this protocol ensures reliable routing of data messages by using acknowledgements and other mechanisms.

3.4.3 I2P Transport layer

It represents the layer number one in the protocol stack of I2P. It provides encryption of connections between every two I2P routers in the tunnel (Hop to Hop connection). Three protocols are implemented to provide these capabilities:

3.4.3.1 SSU (Secure User Datagram Protocol)

SSU protocol is created to use UDP in I2P. It provides encrypted, reliable connection-oriented, point-to-point data transport, IP detection and NAT (Network Address Translation) traversal services. This version is called semi-reliable, in which unacknowledged messages are repeatedly transmitted until reaching a maximum number of times.

3.4.3.2 NTCP (NIO-Based Transmission Control Protocol)

NTCP protocol is built on top of TCP in I2P. NCTP provides encrypted, reliable connection-oriented and point-to-point data transport. By default, it uses the IP/Port detection option of SSU.

3.4.3.3 NTCP2

NTCP2 is an enhanced version of NTCP. The two protocols coexist together and offer more flexibility in I2P. NTCP2 is created to improve the resistance of NTCP to different forms of automated identification and attacks.

3.4.4 I2P tunnel layer

This layer is responsible for the encryption of tunnel connections. The tunnel encryption is layered, in which the first tunnel node repeatedly encrypts the message according to the number of nodes in the tunnel as well as assigns the message at each encryption with instructions for each tunnel node. Afterwards, it forwards the message in the tunnel. Every tunnel node receiving that message decrypts it to get the instructions to forward it to the next tunnel node. The exchanged messages in this process are tunnel messages. They are encrypted messages that contain encrypted I2NP messages [11] and encrypted instructions for their delivery. This process is described in section 3.6.3.

3.4.5 I2P Garlic layer

It is responsible for garlic encryption during the communication. It provides the delivery of garlic messages. These messages are known as I2NP messages (I2P Network Protocol). They are encrypted and wrapped in

each other to be secured and anonymous between the source tunnel and the destination tunnel.

The aforementioned layers represent the layer of the core I2P router functionality. Thereafter, a set of layers of the I2P stack protocol are presented, in which each layer provides additional functionalities to the I2P protocol. Generally, these layers allow simple usage of I2P protocol. It should be mentioned that these layers are not a part of the recent I2P.

3.4.6 I2P client layer

This layer provides the capability for any user to use I2P over the network without getting direct use of I2P routers. This is done through the I2P Client Protocol (I2CP) protocol. I2CP allows secure and anonymous communication between the client application and the I2P router using an I2CP TCP socket. This protocol is used by the client application and the I2P router.

To start the communication, the client application transmits to the I2P router instructions about the needed anonymity, reliability, latency, tradeoff and the time to send messages. For the router, it uses I2CP to request the client application for authorization to use some of the tunnels during the communication as well as to inform the client application about the time of the received messages.

3.4.7 I2P end-to-end transport layer

This layer allows using TCP-like and UDP-like functionalities on top of I2P by providing streaming and datagram libraries.

3.4.7.1 Streaming library

The streaming library (real-time stream) [27] is technically part of the "application" layer. It provides a critical function for almost all existing I2P applications, allowing TCP-like streams across I2P, and enabling existing applications to be easily ported to I2P.

3.4.7.2 Datagram library

It is an end-to-end transport library [4] for client communication. This library represents an implementation of UDP-like messages over I2P to allow easier porting of existing applications of I2P.

Until now, the essential layer in the I2P protocol stack have been presented (as represents figure 3-1). Additional two optional libraries can be added to allow easier implementation on top of I2P. These layers are presented as follows:

3.4.8 I2P application interface layer

This layer provides a set of tools like I2PTunnel [12], SAM1 [23], SAM2 [24], SAM3 [25] and BOB [2] to allow interfacing with and provides services on I2P.

I2PTunnel: with the creation of I2PTunnel tool, the user can provide or interface with the I2P services. it is creation is based on the IP address:port of the destination. Then, a corresponding 516 bytes destination key will be generated for the I2P service.

SAM1 is a client protocol for interacting with I2P.

SAM2 is an improved version of SAM1. It allows to manage several sockets on the same I2P destination in parallel.

SAM3 provides a UDP port for sending datagrams via I2P, and can return I2P datagrams to the client's datagram server.

BOB: in SAM2, the data is transited through the same client<–>SAM socket, which is complicated to manage for the client. In BOB, similarly in SAM3, each I2P socket matches a unique client<–>SAM socket, which is easy to handle.

3.4.9 I2P application proxy layer

It concerns proxy systems in I2P. It provides many tools like Socks proxies [26], HTTP client/server, IRC client, etc.

3.5 Exchange of messages in I2P

To establish a connection between two parts in the I2P network, four tunnels at least have to be created (figure 3-2): one (or several) inbound and outbound tunnels for each part.

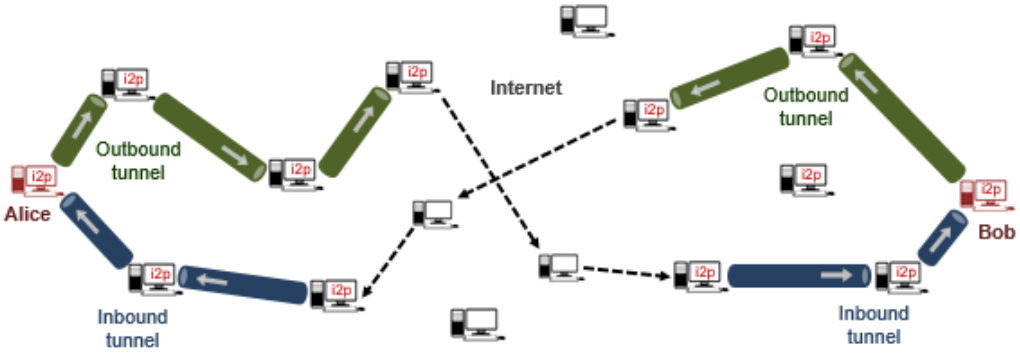


Figure 3-2: Message exchange in I2P

Each I2P router creates its own inbound and outbound tunnels and maintains their existence in the network. This router (creator) can have multiple inbound and outbound tunnels. To send messages, it uses one of its outbound tunnels. Similarly, for the reception, it uses one of its inbound tunnels. To this reason, four tunnels (at least) are required to establish communication between two parts.

To create inbound and outbound tunnels, the node (creator) should have the different identities of I2P routers included in the tunnel by contacting one of the I2P network databases. These databases are distributed in the I2P network and contain identities of all I2P routers. Registration of identities is done the first time when connecting to the network.

As represented in figure 3-2, the client application communicates with the I2P sender router (Alice) to send messages to the I2P receiver router (Bob) that transmits the received messages to the server application. Alice must get the inbound tunnel entrance of Bob and choose one of his outbound tunnels to transmit the messages. When the messages reach the last router of the chosen outbound tunnel, they are routed toward Bob's inbound tunnel using internet routing protocols. When the first I2P router of Bob's inbound tunnel receives the messages, it forwards them in the tunnel to Bob, then to the server application. Acknowledgements are sent back to Alice using the same process between tunnels. If Alice wants to acknowledge their messages, Bob sends back the required acknowledgement through its outbound tunnel, then, Alice's inbound tunnel, then to Alice.

3.6 I2P tunnels and message processing

This section define and presents the several types of tunnels used in I2P during the communication, then details their creation process and the message processing inside tunnels.

3.6.1 Definition and types of tunnels

An I2P tunnel is a list of selected I2P routers (tunnel nodes) forming a direct path responsible of transmitting encrypted messages in one direction. In the outbound tunnels, messages are transmitted from the creator router to the network, while in inbound tunnels, messages are transmitted from the network to the creator. Each tunnel includes a gateway, participants and endpoint nodes as shown in figure 3-3.

According to the standard format of tunnels in I2P, the number of tunnel nodes may vary between one to seven I2P nodes, which defines the length of the tunnel: from 0-hop tunnels to 6-hop tunnels. In 0-hop and one-hop tunnels, participants do not exist, and the creator of the tunnel can be the gateway and endpoint at the same time. Generally, 0-hop and one-hop tunnels ensure less security, two-hop and three-hop tunnels are

usually preferred in terms of security, and tunnels that are longer than 3-hop do not offer additional protection according to some recent studies.

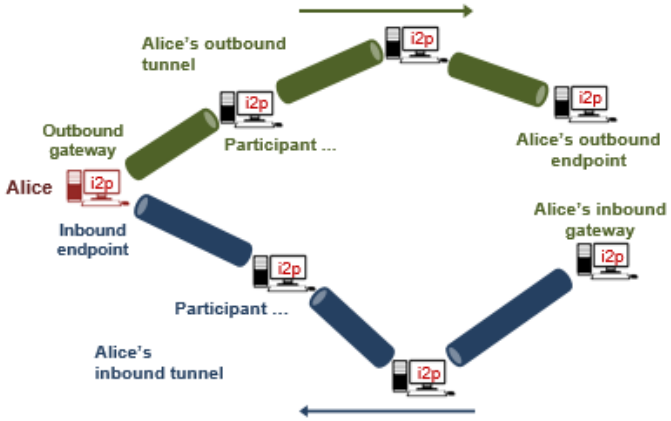


Figure 3-3: I2P tunnels

3.6.2 Tunnel creation process

Every I2P router should create its outbound and inbound tunnels; at least, one outbound tunnel to send data messages and one inbound to receive them. Before creating a tunnel, the I2P router (creator) should have the different identities of the I2P routers that will be included in the tunnel. The creator can obtain these identities by contacting the I2P network databases (NetDB). These databases are distributed in the I2P network (section 3.7).

The tunnel creation process is accomplished by sending a tunnel build message as shown in figure 3-4. This message consists of a specific number of records, where each record is dedicated to be rewritten immediately by one potential peer in the tunnel. The peer replaces the sent record by a reply record putting its response if it agrees to be in the tunnel or not. This response is specified according to some defined rejection reasons such as bandwidth, router shutdown, etc [20].

Each record in the build message is asymmetrically encrypted using ElGamal encryption [5] according to the specific peer that will read it. The second layer of symmetric encryption AES [5] is used at each hop to expose the asymmetrically encrypted record in the appropriate time. After achieving the number of peers required in the tunnel, a reply message is retransmitted back to the creator.

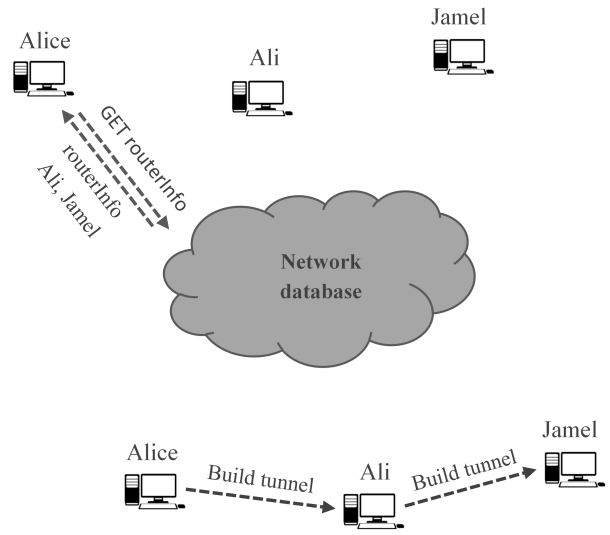


Figure 3-4: Tunnel creation process

Two types of tunnel build messages can be defined: TBM (Tunnel Build Message) and VTBM (Variable Tunnel Build Message). TBM message is a request responsible of the creation of tunnels with a length of 8 peers. It contains 8 records, which lead to the creation of 7-hop tunnel that is the longest practical tunnel in I2P. VTBM contains 1 to 8 records. The number of records can be changed according to the tunnel length needed.

After reaching the last peer in the tunnel, a reply message of the same type and the same length as the tunnel build message is AES encrypted and sent back to the creator. The reply message can be TBRM (Tunnel Build Reply Message) or VTBRM (Variable Tunnel Build Reply Message) if the request is TBM or VTBRM respectively. After receiving the reply message, the creator checks the response of each contacted peer; if all peers agree, the tunnel is considered created and it can be used immediately. Otherwise, if anyone refuses to be in the tunnel, the tunnel is discarded. The reply message may include agreements and rejection reasons mentioned by the contacted peers, in which this information are noted in each peer's profile to be used later in the evaluation of peer tunnel capacity.

Created tunnels are maintained periodically by sending a Delivery Status Message in the tunnel. The creator tests two tunnels at once (one outbound and one inbound) by sending this message through the outbound tunnel then passing through the inbound tunnel.

3.6.3 Message processing inside tunnels (tunnel encryption)

After creating tunnels, nodes can communicate with each other by using their outbound tunnels and inbound tunnels to send and receive messages respectively. To know which inbound tunnel is chosen by the destination,

the source contacts the network database to obtain that information as shown in figure 3-5.

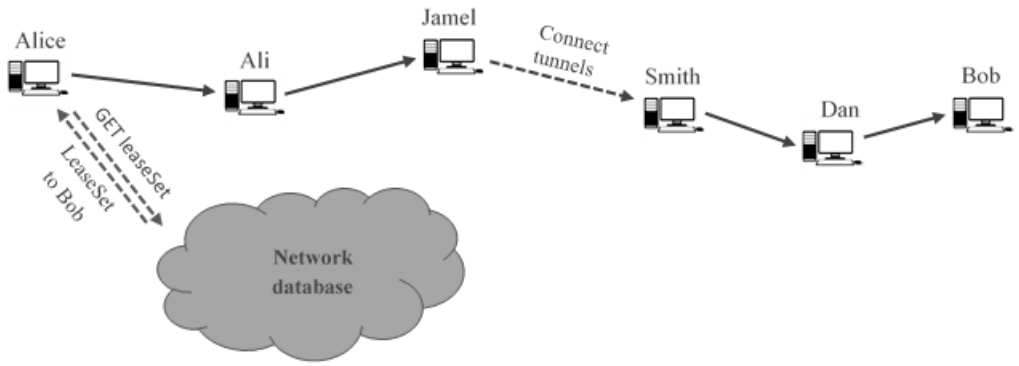


Figure 3-5: Connection to the NetDB to establish communication with the destination

After getting information about the inbound tunnel of the destination, the source can send its data messages as described in the following two steps:

3.6.3.1 Outbound tunnel processing

The processing of data messages is performed according to the tunnel node that sends or receives the message (figure 3-6).

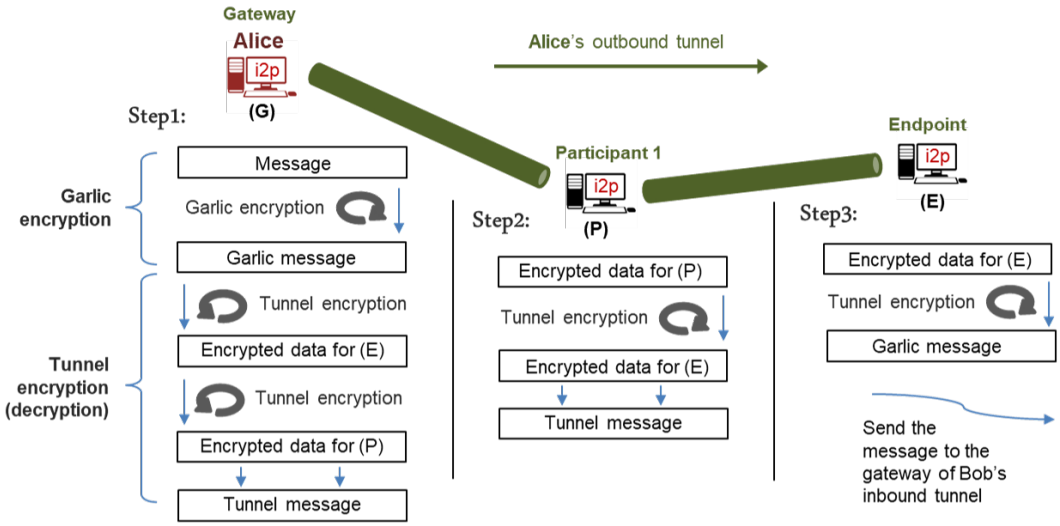


Figure 3-6: Garlic and tunnel encryption in outbound tunnels

The tunnel processing is described as follows for every tunnel node:

Outbound gateway is the I2P router that creates the outbound and inbound tunnels (the creator). It plays the role of the inbound endpoint

for the created inbound tunnels. After the generation of data messages by the I2P client application, the outbound gateway encrypts them and adds instructions that will be used later by the outbound endpoint.

This router encrypts the messages using the garlic encryption and then iteratively encrypts them using the tunnel encryption. The tunnel encryption is performed inversely, where the router decrypts the messages several times according to the number of nodes in the outbound tunnel. Afterwards, it forwards them through one of its outbound tunnels to the next tunnel node.

Outbound participants are the intermediate I2P routers in the tunnel (between the creator and the last I2P router in the tunnel). The participants encrypt the received messages (from the gateway or the previous participant in the tunnel). Then, the participant forwards it to the next participant until reaching the outbound endpoint.

Outbound endpoint acts as the other participants by encrypting the received messages (for the last time) to obtain the garlic message. This message is generated by the outbound gateway using the garlic encryption. It is issued with some delivery instructions about the entry of the inbound tunnel of the destination. In this stage, the endpoint follows these instructions by sending the messages on the internet to reach the correct inbound gateway of the destination.

3.6.3.2 Inbound tunnel processing

Similarly, in inbound tunnels, data messages are processed according to the tunnel node that sends or receives the message (figure 3-7).

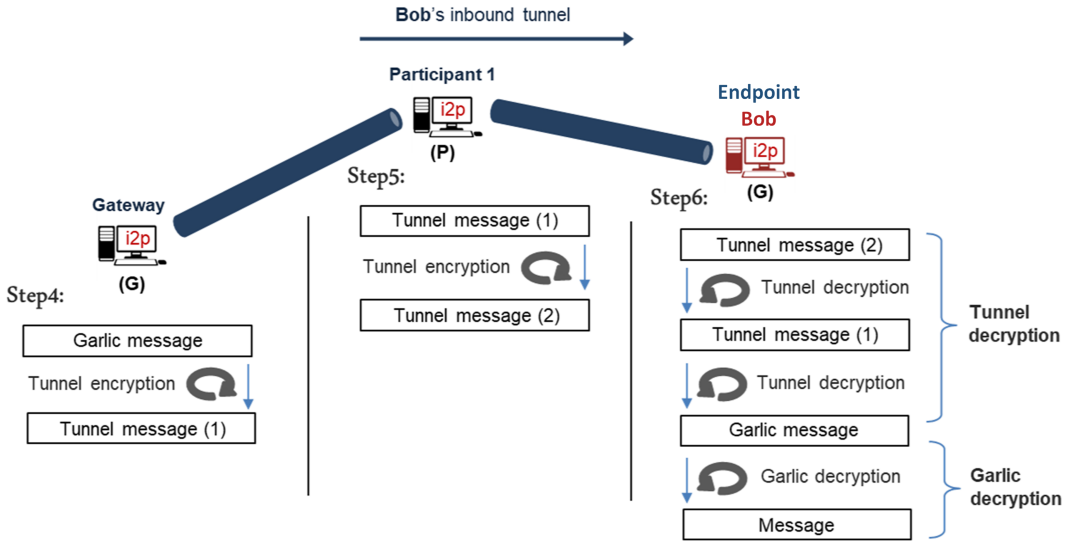


Figure 3-7: Garlic and tunnel encryption in inbound tunnels

Inbound gateway is the input I2P router of the inbound tunnel. At reception, the inbound gateway as inbound participants acts in the same way by encrypting the received messages and forwards them to the next I2P router in the tunnel until reaching the destination.

Inbound participants perform the same process as outbound participants. They encrypt the received messages and then forward them to the next node in the tunnel.

Inbound endpoint is the creator of the inbound tunnel. It decrypts them iteratively according to the I2P routers included in the inbound tunnel of the destination. This decryption is done to obtain the garlic message that will be decrypted using the garlic decryption and then retransmitted to the server application.

3.7 The I2P database (NetDB)

The network database contains the identities of the routers in the network. It is distributed and hosted in different I2P routers. Each I2P router sends its identity (the contact addresses, identification keys like public keys and signing public keys) to this database. The content of this database is distributed in the way that every I2P router can get the information needed about any other I2P router in the network.

Two types of metadata can be stored in the NetDB database: RouterInfo and LeasSet. Different I2P routers send them to the NetDB through their outbound tunnels.

RouterInfo contains the necessary data that allow the routers to contact a particular router in I2P. $RouterInfo$ is signed and it includes the router identity (ElGamal encryption key, a public signing key and a certificate), the contact addresses (transport protocol and port numbers) and arbitrary text options.

LeasSet is a set of leases, in which each lease contains the identity of the gateway of the inbound tunnel created by an I2P router.

After creating tunnels, the creator sends the identities of their inbound gateways as a leaseSet to the NetDB to allow other routers to contact it. Besides, the lease includes the public keys that are used to encrypt the message. An encrypted leaseSet is encrypted and each lease is encrypted by a separate key. It should be stated that the router can regularly revoke its previous $LeaseSets$ by sending an empty $LeaseSets$ (zero leases).

Routers send their $RouterInfos$ to the NetDB directly while $LeaseSets$ need to be sent anonymously by using outbound tunnels to the NetDB.

$RouterInfos$ and $LeaseSets$ are published in Database Store Messages (DSM) to NetDB databases. DSMs are wrapped in garlic messages to hide the content from the outbound gateway.

NetDB databases are hosted in some selected router (Floodfills) with specific characteristics. A $floodfill$ is an I2P router with high performance and powerful characteristics. It advertises itself as a floodfill in the network. After the storage of metadata ($RouterInfos$ or $LeaseSets$) in the NetDB, the router checks if the metadata is successfully registered. To do so, it sends a lookup message after 10 seconds to another $floodfill$ (different from the first one), and then the $floodfill$ checks the NetDB database and responds to the router. A lookup message is end-to-end encrypted using garlic encryption to avoid spying on the outbound endpoint.

3.8 I2P encryption layers

The communication in I2P is secured and encrypted using a large number of cryptographic techniques and algorithms: the complete list includes 2048bit ElGamal encryption, 256bit AES in CBC mode with PKCS#5 padding, 1024bit DSA signatures, SHA256 hashes, 2048bit Diffie-Hellman negotiated connections with station to station authentication and ElGamal/AES + SessionTag.

ElGamal is used for asymmetric encryption. Encrypted messages have 514 bytes of size. In I2P, ElGamal encryption is used:

- For the encryption of tunnel build messages.

- For an end-to-end encryption as part of ElGamal/AES+SessionTag (from source to destination nodes).

- To encrypt some NetDB stores and queries sent to floodfill routers.

AES is used for symmetric encryption in different cases in I2P:

- For transport encryption (section 3.8.4).

- For an end-to-end encryption as part of ElGamal/AES+Session Tag.

- For encryption of some NetDB stores and queries.

- For encryption of periodic tunnel test messages. These messages are delivered through the tunnels then come back to the router itself.

In I2P, the communication is end-to-end encrypted according to four levels as described in figure 3-8: I2P Control Protocol (I2CP), garlic, tunnel and transport encryption [20].

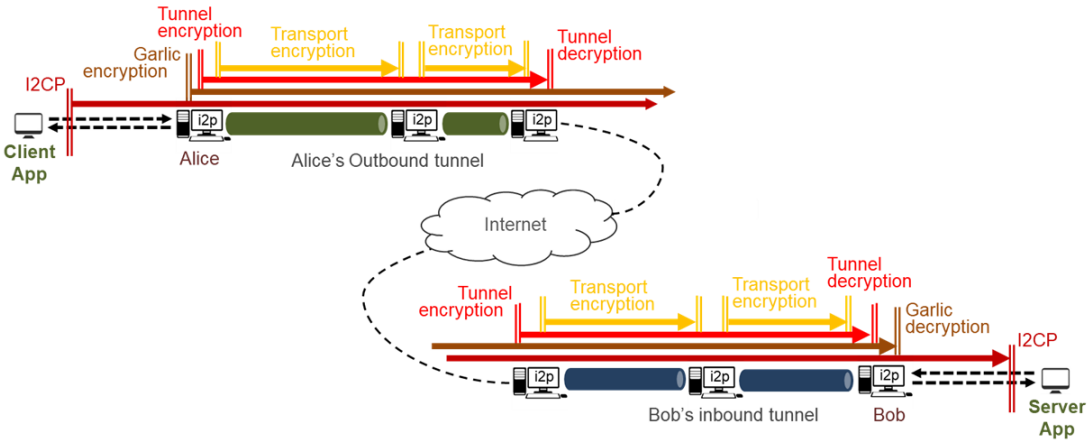


Figure 3-8: I2P encryption layers

3.8.1 I2CP encryption

I2P Control Protocol (I2CP) can be considered as an interface between clients (client applications) and the I2P router. It secures the exchange and enables asynchronous messages, which are sent and received over a single TCP socket. The client application informs the router about the anonymity, the reliability, the latency tradeoffs to make, and where to send messages. Consequently, the router chooses the tunnels that can be used to send and receive messages and informs the client when a message has arrived.

Messages are transmitted from the client application to the router after they are I2CP encrypted at the client level. These messages are encrypted as well at the server applications level (from client application to server application).

3.8.2 Garlic encryption

In garlic encryption, messages are El Gamel/AES encrypted, in which multiple messages are bundled together in one message. Each one of them is called "clove" and it is associated with its delivery instructions that are exposed at the endpoint node.

The sender I2P router (outbound gateway) uses the garlic encryption to encrypt the messages in the garlic format. Messages are encrypted by the sender router and decrypted by the destination router.

Most of the messages exchanged between two nodes in I2P are garlic wrapped. The sender bundles its current lease in the message, which allows the destination to reply without consulting the network database.

3.8.3 Tunnel encryption

For the tunnel encryption, messages are encrypted inside the tunnel from the gateway to the endpoint. The tunnel encryption is launched by the gateway node. For outbound tunnels, the gateway encrypts (decrypts) the messages several times according to the number of nodes in the tunnel. Then, each node in outbound or inbound tunnels excluding the creators encrypt only one time the received messages. Upon receipt by the inbound endpoint (the destination), it decrypts the messages iteratively until getting the processed data. This encryption process is detailed in section 3.6.3 (Message processing inside tunnels (tunnel encryption)).

3.8.4 Transport encryption

A transport in I2P is a point-to-point communication between two routers. Messages are encrypted by each tunnel node then decrypted by the next one in the tunnel. Transport encryption uses 256 bytes Diffie-Hellman to generate the secret key followed by symmetric AES256/CBC encryption [10]. This provides perfect forward secrecy on the transport links. It is important to point out that I2P supports three transport protocols simultaneously: SSU, NTCP and NTCP2 as defined in section 3.4.3.

3.9 Difference between I2P and VANet

Vehicular ad hoc networks have critical characteristics that influence communication and degrade the quality of services. Communication types in VANet are classified into three types: communication between vehicles (V2V), communication between vehicles and infrastructure (V2I) and hybrid communication between V2V and V2I. Diverse environment models exist in VANet: highway, rural and urban, in which several layouts such as roads, buildings, junctions, trees, traffic regulations, etc can exist.

This diversity in types of communication and environment models may influence the communication. Indeed, the presence of obstacles can cause disruption during exchanges as well as the high speed of vehicles and fast topology changes that pose significant problems of radio propagation instability.

The velocity of node combines the speed and direction of the movement [31], its value can be changed from 0 km/h in traffic jam situations to more than 200 km/h in highways. This high mobility of vehicles causes rapid topology changes, which causes link failures. The links lifetime on highways is generally 50 seconds for vehicles going in the same direction. For vehicles in the opposite directions, links can have lifetimes less than 5 seconds, which represents short durations and consequently causes broken links during the communication. At low velocities when the vehicles move slowly, the network will be dense and consequently overloaded with a large

number of messages, which influence QoS parameters such as end-to-end delay, and the packet delivery ratio.

I2P network has totally different features compared to VANet. It is a part of the internet, in which I2P routers are generally connected using wired links which are permanent and rarely broken. Stability of routers in I2P makes the communication more stable with fewer link failures, and consequently, created tunnels within the I2P network can have a long lifetime (10 minutes) and a high bandwidth offered by the wired links during the communication.

3.10 Application of I2P in VANet

This section proposes a secure model inspired by the I2P protocol to provide anonymity and security in VANet. I2P is taken as a reference model to adapt its algorithms and mechanisms according to requirements of VANet such as high mobility and rapid topology change.

The encryption algorithms are used to secure messages and digital signatures to ensure their integrity and authenticate users. In our work, three levels of encryption used in I2P are implemented; transport, tunnel and garlic encryption. Besides, creating tunnels and maintain them using the tunnel creation and maintenance process implemented in I2P.

The problem that occur in our proposed model in VANet is the use of multiple protocols of security, which can influence the communication and degrade the network performance. As detailed in this chapter, sending data between two nodes for instance needs the creation of four tunnels at least. Moreover, the encryption and decryption processes executed by tunnel nodes, which can delay the communication. For this reason, development of tunnel creation and maintenance algorithms or encryption algorithms in VANet need to take in consideration different characteristics of this category of networks. Given the high speed of vehicles and the rapid change of topology, the developed algorithms must be more secure and robust to be successfully implemented in VANet.

Adaptation of I2P algorithms in VANet has been done in two contributions. The work is structured into two steps to simplify and organize the implementation of the protocol in VANet as shown below:

a. Contribution 1:"A secure communication model using lightweight Diffie-Hellman method in vehicular ad hoc networks"

In this study, we propose using the transport encryption layer implemented in I2P in the attempt to secure the communication in VANet. Moreover, a novel approach of security is presented to face some known attacks, in which we design a model of communication that combines digital signature and message authentication mechanisms in order to securely generate the secret key. Thus, achieve integrity, confidentiality, session key

security and non-repudiation.

Transport encryption is the lowest encryption layer in I2P. It is implemented inside the tunnel between each two successive tunnel nodes. In this study, the transport encryption is assumed that is implemented between the source and the destination nodes. The encryption process is simplified by assuming that the source and the destination nodes are successive tunnel nodes in a tunnel.

For the next study, when tunnels are created, the same proposed protocol will be implemented assuming that successive tunnel nodes in a tunnel act as source and destination nodes.

b. Contribution 2: "A new protocol to anonymize communication in VANet"

The second contribution is divided into two parts:

- Part 1: "Anonymizing communication in VANet by applying I2P mechanisms"

In this part, we continue the previous work by creating tunnels and implementing the tunnel and the garlic encryption algorithms. They respectively represent the second and third layers of encryption above the transport encryption layer in I2P. As an initiation for monitoring tunnels in VANet, we create tunnels and establish communication between vehicles in a static way and without maintenance. Simultaneously, the encryption of messages is ensured by adapting the tunnel and garlic encryption layers, in which the operation mode of each level is simplified by using an asymmetric algorithm.

- part 2: "A new proposed protocol to anonymize communication in VANet"

In the second part, we treat the mobility of vehicles, in which a tunnel maintenance algorithm is developed to maintain the existence of the created tunnels in the network. This algorithm is integrated in the proposed protocol in first part to create an improved version of the protocol that can be successfully implemented in VANet.

3.11 Conclusion

I2P protocol is still in development. Its founders claim the need for deeper studies to achieve a stable release with more functionalities and security mechanisms.

Applying I2P algorithms in VANet is very complicated due to the great difference between their characteristics. Application of an I2P mechanism in VANet needs to respond to different requirements of this category of networks. On the one hand, the adaptation of an I2P algorithm of security

needs to respect certain required QoS parameters such as end-to-end delay, overhead, packet delivery ratio, etc. On the other hand, it ensures the same level of security guaranteed by I2P in VANet.

In the proposed approach, we try to cop with the different requirements of VANet and adapt the I2P mechanisms and algorithms. As detailed in the previous section, the adaptation of the I2P protocol in vehicular ad hoc networks is done in two contributions. the first one concerns the transport encryption layer, where we use the same principle used in I2P with some updates in the overall mechanism. The second one which is the most important part concerns the tunnel and garlic encryption layers. This part is divided into two steps:

1- Creating tunnels in a static way and implement the garlic and tunnel encryption layers of I2P.

2- Implementing the proposed protocol in real VANet by adding a tunnel maintenance algorithm, which maintains the existence of the created tunnels in the network.

Chapter 4

CONTRIBUTION 1: A SECURE COMMUNICATION MODEL USING LIGHTWEIGHT DIFFIE-HELLMAN METHOD IN VANET

4.1 Introduction

Development of security algorithms in vehicular ad hoc networks (VANet) is more critical when it comes to their critical characteristics such as high mobility of vehicles, rapid change in topology and wireless nature of communication, which require the implementation of lightweight security algorithms.

Progressively, attacks in VANet become numerous and different types of attacks appeared. According to the purpose of the attack, the malicious node can eavesdrop, usurp identity, alter or drop messages. Some known attacks can be launched in VANet: man-in-the-middle (MITM) attack, replay attack, Sybil attack, denial-of-service attack, etc. Facing these attacks and responding to security requirements consist of using encryption algorithms and secure mechanisms, which allow the encryption of the communication and reducing misbehaviours of malicious nodes in different attacks.

Another aspect is treated in this thesis is the anonymity of communication in VANet. The objective is to secure and anonymize this communication by preventing other entities to know the real identities of communicating nodes and encrypting the content of messages.

Contributing to this issue, we took the Invisible Internet Project (I2P)

[20] used on the internet as a reference model, and we adapt its security mechanisms to VANet. The I2P protocol is chosen thanks to its high security and anonymity level compared to other protocols like Freenet [9] and Tor [3]. I2P is designed for wired networks, which makes its application to VANet more difficult and needs to take into account their different critical characteristics.

As an initiation to implement I2P in VANet, this contribution proposes a secure model based on the first layer of encryption of I2P "Transport encryption" to ensure integrity, confidentiality, authenticity and non-repudiation. I2P transport encryption is used to encrypt messages between each two successive tunnel routers in I2P. However, in this contribution, the transport encryption is implemented between the source and destination nodes (considering that the source and the destination nodes are two successive tunnel nodes) to simplify the implementation of the protocol.

The proposed model is designed to prevent attackers to read, alter or even drop exchanged messages without detection, if a malicious node causes any changes, it will be detected immediately and the communication will be re-established using another possible solution as it is detailed in this chapter. For that, a symmetric encryption algorithm is used to encrypt messages based on a session key. The invulnerability of symmetric algorithms is related to the way used to exchange the session key between the source and destination nodes. Diffie-Hellman is the most used method to generate this secret key. However, this method cannot ensure the authenticity of nodes and some attacks like MITM attack can be launched easily and the communication can be decrypted. The Diffie-Hellman algorithm must be implemented with an authentication mechanism to ensure that each message is related to its node.

In this work, a lightweight Diffie-Hellman method is used to securely exchange the secret key on one side and to respond to different critical characteristics of VANet on another side. Two attacks are treated: MITM and replay attacks by using a signature mechanism to authenticate participant nodes during the communication.

This chapter is organized as follows. Introducing the objective of the first contribution in the section 4.2. Discussing several related works in section 4.3. Section 4.4 is an overview of the main topics related to the contribution: Diffie-Hellman exchange, Man-In-The-Middle and replay attacks. The section 4.5 explains the operation of the proposed model and section 4.6 shows its security in different cases of attacks. Finally, we end by a conclusion in section 4.7.

4.2 Implementing the transport encryption of I2P in VANet

Our goal in this thesis is to provide security and anonymity of communication in VANet. For that, the I2P is chosen as a reference protocol to implement its secure algorithms and mechanisms in VANet. The adaptation of I2P to VANet is simplified by dividing this work into several parts. Each part deals with a set of algorithms and mechanisms providing security and anonymity of communication in VANet.

In this contribution, we develop the first version of the I2P protocol in VANet. We propose a model based on I2P transport encryption layer to provide secure communication in VANet. This model combines digital signature and message authentication and symmetric encryption algorithms to securely generate the secret key and encrypt messages, and thereby, achieve authenticity, integrity, confidentiality and non-repudiation. Similarly, to I2P transport encryption layer, in this approach, the Diffie-Hellman algorithm is used as a method to exchange the secret key during the communication.

The section below presents a state of the art about approaches have been proposed to provide security of communication in VANet.

4.3 Related work

Authors in [66] proposed a novel conditional privacy-preserving authentication scheme to secure the communication in VANet. Its scheme combines road side units (RSU)-based and tamper-proof device-based schemes. They use trusted authority (TA) to set up the system parameters. TA and RSUs are the only entities that are responsible for authentication of vehicles and tracing and revocation of malicious vehicles. From an anonymity perspective, OBUs are supposed to use pseudo-ID during communication.

Wang and Yao in [80] have shown a local identity-based anonymous message authentication protocol in VANet. This approach uses a components layer: Certificate Authority (CA) and Traffic Management Center (TMC), dedicated to security management that delivers long-term certificate for RSUs and vehicles. When the node is malicious, CA can revoke its certificate. Using an additional layer of entities responsible for security management can generate a large number of additional messages intended to ensure authentication, which does not correspond to VANet environments in some scenarios.

In [84], authors proposed a new attribute-based authenticated protocol for secure communication of VANet. They combine the attribute-based sincryption (ABSC) and the attribute-based signature (ABS) with VANet. They use ABSC to obtain the secret keys used during the communication from RSUs and use ABS to make communication anonymous with vehicles

and RSUs. In this scheme, keys management centre (KDC) is the most important entity which distributes keys to all vehicles in the region. The way of key distribution is based on vehicles behaviour, and sometimes, it needs more time to decide by the KDC to deliver the key to the vehicle, and it can cause a high delay in some cases.

Another contribution in [52], it consists of a highly efficient randomised authentication in VANet. Using homomorphic encryption in this protocol, each vehicle can ensure anonymity by self-generating some authenticated identities according to the established communications. The verification of these randomised identities can be done by a collaboration of a pair of authentication servers.

In [87], a new distributed aggregate privacy-preserving authentication protocol in VANet is presented. It is based on a security tool called Multiple Trusted Authority One-Time Identity-Based Aggregate Signature (MTA-OTIBAS). MTA-OTIBAS scheme is used to mitigate the issue of time-consuming and huge volume of data used in cryptographic algorithms. In this scheme, the root TA is the most important entity in the authentication process, in which it generates and controls the system parameters. According to this proposition, an entity can only verify the signatures generated by a vehicle in the same and/or neighbouring group(s) and not in other distant groups in the whole network.

Authors in [81] proposed a secure official vehicle communication protocol for VANet. This proposal combines homomorphic key agreement and symmetric encryption with a mechanism of digital signature and messages authentication. In this work, a large number of messages is used between the key management server and civil servants during the generation of the session key and even to access the internet resources, which does not respond to VANet characteristics in some scenarios.

Authors in [88] proposed in their paper a conditional privacy-preserving authentication scheme that secures the communication and in parallel reduce the communication overhead. The idea in this work is to use the registration list instead of the revocation list, which allows to prevent the attacker from continually sending malicious information.

Ying and Nayak [85] presented an anonymous and lightweight authentication based on a smart card for secure vehicular networks. In this approach, the TA assigns each user in the network with an anonymous certificate and session key. The special feature in this work is the intervention of the user to authenticate the vehicle. He can log in by introducing the smart card to the terminal device (OBU) and inputting the required parameters. Generally, the recent studies in VANet aim to enhance driving and make these networks smarter with the minimum intervention of users, while in this proposition, the user has to intervene in the authentication process which is uncomfortable and not practical.

Another work about authentication in VANet is proposed in [77]. It consists of an anonymous authentication scheme based on PIMPv6. In this

scheme, two layers are added to manage the authentication and security of communication: STR which generates public parameters and private key for other entities, in addition to playing the role of group manager that issues the group certificate for the legal vehicle. The second layer is LMA which manages mobility of vehicles and helps them for the generation of the variable pseudonyms.

In [76] authors proposed a trust-based authentication technique for cluster-based VANet. The approach used in this work is based on a clustering method. The main idea is to create clusters of vehicles before estimate the trust degree of each node to select cluster heads. Providing authentication in this scheme refers to the use of the digital signature of messages sent and encrypted by the sender using public/private keys distributed by trust authority. In this approach, the creation of clusters and selection of cluster heads cause more delay in such scenario in VANet, which influences the authentication process.

In the field of research on authentication, most of the works use an additional entity (or entities) playing the role of trusted party that issues and manages security parameters for other entities. This implies usually exchange messages with this party to ensure security and authentication of communication, which generates a large number of additional messages, while critical characteristics in VANet involve minimising the number of messages as well as possible. In the proposed approach, no use of additional entities and the only communication needed for authentication is established before starting the exchange of data messages between nodes with the nearest RSU, which reduces the total number of messages and delay during the communication. Thus, according to the proposed authentication mechanism, each vehicle can verify the validity of the received messages and detect several known attacks when they occurred.

Before starting the description of the proposed model, an overview of security concepts such as the Diffie-Hellman method, Man-In-The-Middle and replay attacks is presented in the section below.

4.4 Overview of security concepts

This section presents an overview of some security requirements in this contribution: Diffie-Hellman method that is used by the I2P transport encryption layer, thus, this method is used in the proposed model trying to ensure the same level of security provided by the I2P layer. After that, two known attacks in VANet are presented: Man-In-The-Middle and replay attacks. The Diffie-Hellman method suffers against the Man-In-The-Middle attack, where, it is needed to implement security mechanisms to face this attack. Besides, the replay attack is treated due to its bad impact on the communication, in which it can cause disturbance and problems on the communication behaviour. That is why these two attacks are chosen (as

examples of attacks) among different attacks. Furthermore, the main goal behind this contribution is to implement the I2P transport encryption layer as a first step to apply the I2P protocol in VANet.

4.4.1 Diffie-Hellman exchange

In symmetric algorithms, the most critical issue is the security of the key, in other words, how can the source and destination nodes exchange this key securely and in a wireless environment. In this contribution, a novel approach based on symmetric algorithms is proposed, in which a lightweight Diffie-Hellman method is used to exchange the secret key.

Diffie-Hellman method [79] is used to exchange keys in the network in a secure way. Its operation is based on the use of two public parameters g and p, and a private number Prk_{DH} (Diffie-Hellman private key) to calculate a public number Pk_{DH} (Diffie-Hellman public key) knowing that g is a primitive element and p is a large prime number. Figure 4-1 describes the Diffie-Hellman exchange as following:

- A (and B) calculates $Pk_{DH(A)}$ ($Pk_{DH(B)}$) and sends it in public to B (to A), (knowing that p and g are public).
- A and B generate the Diffie-Hellman secret key $Sk_{DH(A,B)}$.

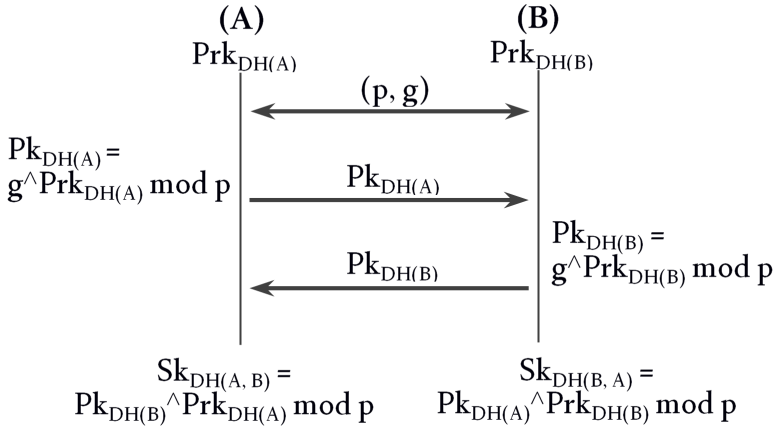


Figure 4-1: Diffie-Hellman exchange

4.4.2 Man-In-The-Middle (MITM) attack

Despite the efficiency of Diffie-Hellman method in the confidentiality of the secret key, it is vulnerable to the MITM attack [82]. The Diffie-Hellman algorithm does not authenticate participant nodes in the network, which makes easy to malicious nodes to eavesdrop messages and possibly change their contents without any detection.

The MITM attack is launched by an intermediate node that intervenes secretly between tow end nodes during the communication. This node

can listen to or even alter messages secretly. Active eavesdropping is an example of the MITM attack, in which a malicious node can listen and control messages between the source and destination nodes.

MITM attack can be launched during the Diffie-Hellman exchange. When two nodes S and D exchange theirs $Pk_{DH(S)}$ and $Pk_{DH(D)}$ respectively, the attacker M intervenes (figure 4-2) to generate two secret keys: a key between S and M, and the second between M and D as calculated below:

$$
\begin{aligned}
S : Sk_{DH(S,M)} &= (g^{Prk_{DH(M)}} modp)^{Prk_{DH(S)}} modp \\
D : Sk_{DH(D,M)} &= (g^{Prk_{DH(M)}} modp)^{Prk_{DH(D)}} modp \\
M : Sk_{DH(M,S)} &= (g^{Prk_{DH(S)}} modp)^{Prk_{DH(M)}} modp, \\
Sk_{DH(M,D)} &= (g^{Prk_{DH(D)}} modp)^{Prk_{DH(M)}} modp
\end{aligned}
$$

Using the two generated secret keys, M can decrypt the communication between S and D without detection. Responding to this attack, a model of communication that ensures the authenticity of messages is proposed, in which a signature mechanism is used to sign the exchanged messages, and each message is identified by its real sender node.

Figure 4-2: MITM attack during the Diffie-Hellman exchange

In parallel, a dangerous known attack in VANet must be faced, which is the replay attack. The section below describes this attack.

4.4.3 Replay attack

The idea of replay attack [70] is to gather and store messages by an attacker, then reuse them later as a legitimate node. In VANet, malicious nodes launch this attack to collect messages generated by nodes in some conditions as traffic events, accidents, etc. and reuse them later. The attacker here aims to deceive vehicles about false situations. For an example of congestion (because an accident event), the nearest nodes generate alert messages to inform distant vehicles to take other roads and avoid the congestion. Here, the attacker gathers these messages and resends them later

at the planned time and in the right place.

This contribution treats the MITM and the replay attacks during Diffie-Hellman exchange and communication according to the proposed communication model.

4.5 The proposed communication model

4.5.1 Model overview

4.5.1.1 Model purpose

This chapter presents the first contribution to implement I2P in VANet. We aim to provide secure and anonymous communication in VANet by using I2P algorithms and mechanisms. As an initiation, we implement the first layer of encryption used in I2P "Transport encryption layer" to encrypt communication between the source and destination. In parallel, we aim to face the MITM and replay attacks in different scenarios. The proposed model should be adapted and enhanced according to different requirements of VANet. In VANet, RSUs can have a high level of performance and security between them compared to vehicles due to their wired communication. The section below describes how the proposed approach can exploit this feature to provide better results.

4.5.1.2 Exploiting security and performance of RSUs

In the Diffie-Hellman method, the exchange of keys is established directly between the source and destination nodes using intermediate nodes. This way of exchange is less efficient for VANet. From the security perspective, malicious nodes can easily intercept the exchanged keys when they are part of the chosen route. From the performance perspective, Pk_{DH} keys have large size (up to 2,048 bytes for RSA algorithm) which decreases performances when using wireless connections.

Generally, RSUs communicate with each other using a wired connection that can be exploited to exchange large data size. In the proposed model, it is supposed that RSUs host databases (as described in the next section) and large data are frequently exchanged. Due to high performance on computation speed and high bandwidth compared to wireless connections, RSUs can share the contents of their databases rapidly and more securely using secured wired channels as represents figure 4-3.

In this contribution, new features are used to improve the performance of the protocol. the following section presents these features.

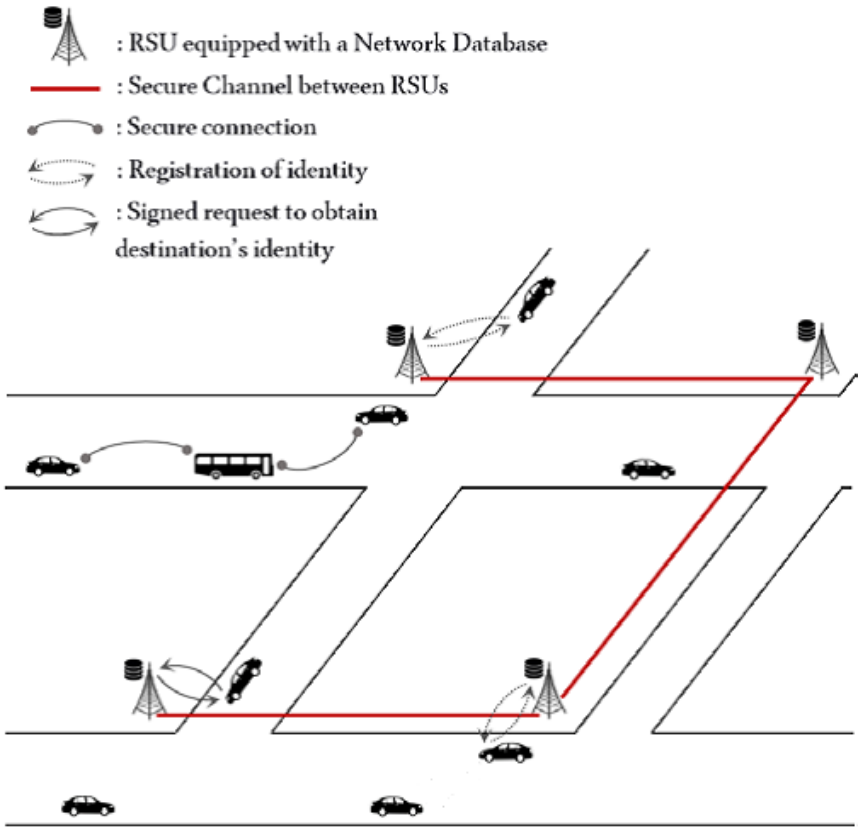


Figure 4-3: A general scheme of the proposed model

4.5.1.3 Using indexes and network databases

As a security measure, it is supposed that a vehicle can have multiple Diffie-Hellman public keys, in which each key is attributed with an index allowing to differentiate it from other keys. This allows the node to use different Diffie-Hellman public keys with different nodes during the communication, which makes the deduction of session keys more difficult for attackers. According to the proposed model, RSUs sign each data received from vehicles using the administrated private signing key, this is based on RSU's signature mechanism to ensure the authenticity of messages during the communication.

In the proposed model, the exchange of Pk_{DH} keys is not done directly between the two end nodes, but they were exchanged and recorded previously in network databases (NetDB) hosted in RSUs. Each vehicle records its identity (during the identity registration phase (IRP), section 4.5.2.1) to a NetDB. All NetDBs, share periodically their contents between each other to have identities (including Pk_{DH} keys) of all vehicles in the network.

Generally, the use of a central database in the network is more preferable compared to distributed databases regard to the frequent updating process of entries. Whereas, in our communication scenarios, entries are

recorded and diffused between RSUs just in the first time when the network is created. After that, RSUs add only entries of new nodes that enter into the network. Moreover, entries have a long lifetime (about hours and even days) which makes the updating process launched for long times. For that, the use of a central database in our scenario does not offer a great benefit regarding the updating process compared to distributed databases on one side. In the other side, using central database causes long delays when each RSU consults this central database, contrary to distributed databases that respond immediately to vehicles requests.

The section below describes the Diffie-Hellman method according to the proposed model.

4.5.1.4 Lightweight Diffie-Hellman exchange using routing protocols

Taking a scenario of two nodes S and D want to communicate between each other. Firstly, instead of exchanging Pk_{DH} keys between S and D (using long paths between them sometimes), each of them has just to contact the closest NetDB (closest RSU) to obtain the key of the other node. However, assuming that vehicles have multiple Pk_{DH} keys, S and D have to inform each other about which Pk_{DH} key is chosen. For that, it is supposed that S and D exchange indexes of their selected keys. This exchange is performed at the network layer during the routing process.

Different routing protocols either reactive or proactive protocols can be used in the exchange of the indexes. In the former type like in AODV protocol, the route is often discovered just before sending data. In this type, Pk_{DH} key indexes are exchanged between S and D during the route discovery process. For proactive protocols (OLSR for example), routes are discovered previously between all nodes in the network even no communication is launched between them, in this case, all nodes can exchange their indexes during the routing process in advance.

The proposed model is designed to urban environments where a high number of RSUs can exist and vehicles have low speed (up to 50 km/h). It is supposed that the IP address, private and public signing keys (Prk_{Sig} and Pk_{Sig}) of RSUs are configured by the administrator, where each vehicle knows the IP address, Pk_{DH} and Pk_{Sig} of RSU (Pk_{Sig} and Prk_{Sig} are used in the signature process). The operation of the model is divided into six phases as below: Identity Registration Phase (IRP), Encrypted IRP (EIRP), Lightweight Diffie-Hellman Exchange Phase (LDHEP), Encrypted Communication Phase (ECP), Identity Update Phase (IUP) and Identity Revocation Phase (IVP) that is triggered if a revocation of an identity is needed. These phases are detailed as follows in the next section. The notations used in the rest of this chapter are decrypted in the table 4.5.1.4.

V	Vehicle
RSU	Road side unit
S	Source vehicle
D	Destination vehicle
NetDB	Network database
Pk_{DH}	Diffie-Hellman public keys
$Pk_{DH(S)\_i}$	Diffie-Hellman public key with index i of vehicle S
Prk_{DH}	Diffie Hellman private keys
$Prk_{DH(S)\_i}$	Diffie-Hellman private key with index i of vehicle S
Sk_{DH}	Diffie-Hellman secret key
i, j	Indexes of Diffie-Hellman public keys
Pk_{Sig}	Public signing key
Prk_{Sig}	Private signing key
Sig_{PrkSig}	Signature algorithm using Prk_{Sig}
t_{Sig}	Time to send message after signature
h	Hash function (MD5, SHA-1, etc.)
e_{Sk}	Encryption algorithm using session key
de_{Sk}	Decryption algorithm using session key
t	Current time
Lt	Life time of node identity in NetDB
t_Lt	Temporary (short) life time
TI	Interval of time
ΔT	Time threshold to detect replay attack
RegRq	Registration request
RegRp	Registration reply
RegAck	Registration acknowledgement
ChNgh	Checking neighbours
CRegRq	Encrypted registration request
ChRq	Checking request
ChRp	Checking reply
IdtRq	Identity request
IdtRp	Identity reply
UpdRq	Update request
UpdRp	Update reply
RevRq	Revocation request
RevAck	Revocation acknowledgement
CMRP	Control message of the routing protocol
Data	TCP/UDP packet
mSize	Bits number of message
Cut(n, m) (Mg)	Bits from position n to m of message Mg
encData	Encrypted data (TCP/UDP packet)
encMsg	Encrypted message

Table 4.1: Notation description 1

4.5.2 Model phases

This section presents the operation of the proposed model according to six phases: IRP (IRRP, IRCP), EIRP, LDHEP, ECP, IUP and IVP phases. These phases are described as following:

4.5.2.1 Identity Registration Phase (IRP)

This phase is divided into two sub-phases: identity registration request sub-phase (IRRP) and identity registration checking sub-phase (IRCP).

a. Identity Registration Request sub-Phase (IRRP)

Initially, when a vehicle enters the network and passes near to an RSU, it can be notified that it is close to this RSU by receiving beacons. These beacons are diffused periodically by each RSU in the network. Each beacon contains the IP address of the RSU sender. At reception, the vehicle registers its identity to that RSU by sending a registration request (RegRq) including its $@ip_{(v)}$, $Pk_{DH(v)}$ and $Pk_{Sig(v)}$. Pk_{DH} keys are sent in such order within the RegRq message according to their indexes.

When receiving RegRq, RSU records temporary a set of entries according to the number of keys within Pk_{DH}. Each entry contains $@ip_{(v)}$, $Pk_{DH(v)\_i}$, index (i), $Pk_{Sig(v)}$ and temporary lifetime. Then, RSU generates a signature of the received data in RegRq along with the time of sending a registration reply (RegRp). Here, as represents figure 4-4 (and algorithm 1), the RSU sends to the vehicle a RegRp message including its signature $Sig_{(RSU)}$. This message used to allow the vehicle to verify if the received data in RegRq corresponds to its real identity or not. After that, the RSU waits for a certain time for the confirmation from the vehicle to record these entries for a long lifetime. During the waiting time, RSU does not publish the temporary recorded entries to other RSUs and even to vehicles requesting for this identity.

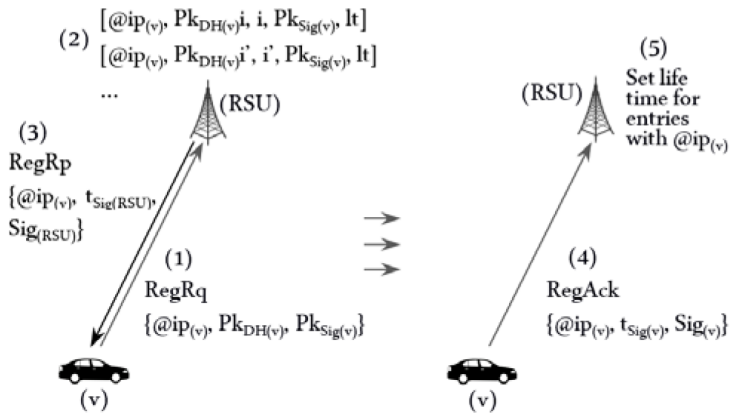


Figure 4-4: Identity Registration Request sub-phase

At the reception of RegRp message, the vehicle verifies the sending time of this message by the RSU and then the correctness of its signature. If the RegRp is sent within a small interval (the replay attack is not launched) and the information is valid, the vehicle resends a registration acknowledgement (RegAck) to the RSU. the RegAck message is signed using $Prk_{Sig(v)}$ as detailed in algorithm 1. It is a confirmation from the vehicle to confirm the registration and means that the recorded entries in correspond to its identity in the network database NetDB.

Algorithm 1 IRRP sub-phase algorithm

Let $n \in N$ */ n: number of* Pk_{DH} *for one node*
V: **For each** Prk_{DH_i} (i=1 to n)
$Prk_{DH\_i} \leftarrow g^{\wedge}Prk_{DH\_i} \bmod p$
End for each
$Pk_{DH} \leftarrow \{Pk_{DH\_1}, Pk_{DH\_2}, \ldots Prk_{DH\_n}\}$
$V \rightarrow RSU$: RegRq $\{@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}\}$
RSU : **For each** Pk_{DH_i}
NetDb.addEntry($@ip_{(v)}$, Pk_{DH_i}, i, $Pk_{Sig(v)}$, t_Lt)
End for each
$Sig_{RSU} \leftarrow Sig_{PrkSig(rsu)}$ $(@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(rsu)})$
$RSU \rightarrow V$: RegRp $\{@ip_{(v)}, t_{Sig(rsu)}, Sig_{RSU}\}$
V: **if** $t - t_{Sig(rsu)} < \Delta T$ **then**
$Hsh_V \leftarrow h$ $(@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(rsu)})$
$Hsh_{RSU} \leftarrow de_{PkSig(rsu)}$ (Sig_{RSU})
if $Hsh_V = Hsh_{RSU}$ **then**
$Sig_V \leftarrow Sig_{PrkSig(v)}$ $(@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(v)})$
Else repeat **IRRP**
End if
Else repeat **IRRP**
End if
$V \rightarrow RSU$: RegAck $\{@ip_{(v)}, t_{Sig(v)}, Sig_V\}$
RSU: **if** $t - t_{Sig(v)} < \Delta T$ **then**
$Hsh_{RSU} \leftarrow h$ $(@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(v)})$
$Hsh_V \leftarrow de_{PkSig(v)}$ $(Sig_{(v)})$
if $Hsh_V = Hsh_{RSU}$ **then**
For each entry with $@ip_{(v)}$
(NetDb.getEntry(@ip)).setLifeTime()
End for each
End if
End if
Each TI Diffuse last recorded entries to other RSUs

By receiving RegAck, the RSU checks if no possibility of Replay attack using $t_{Sig(v)}$ field. RSU searches for the suitable entry in its database and verifies the correctness of the signature in RegAck using $Pk_{Sig(v)}$ in the

entry. If the signature is correct (other cases is discussed in section 4.4), the RSU records the previous temporary entries for a long defined lifetime.

In this case, the RSU can publish the identity of this vehicle and reply to the requests searching for this identity (Pk_{DH} key) of that vehicle. Periodically, each RSU diffuses the recent recorded entries in its NetDB database to other RSUs using secure wired channels between each other.

After a period of time, all vehicles entered in the urban environment, they have diffused their identities to RSUs. Thus, all RSUs have shared their NetDBs contents. Similarly, each new vehicle enters to the network, it follows the IRRP process to register its identity.

As a result, each RSU (database) will have the same content about all nodes in the network. In this case, each vehicle can obtain information about any other node in the network by contacting one of RSUs.

b. Identity Registration Checking sub-Phase (IRCP)

At the previous sub-phase, after sending the RegAck message to the RSU, the vehicle does not know if RegAck is received by the RSU or not. This message can be intercepted and dropped by a malicious node or simply cannot reach the RSU because of routing problems. With no RegAck received, the RSU deletes the entry of the vehicle when its temporary lifetime expires. The problem here is that the vehicle considers that its identity is successfully recorded in the NetDB database while it is not.

Treating this problem, a registration checking process is added. It is triggered after the identity registration request sub-phase (IRRP) with a period (longer than the time required to diffuse the information to all RSUs).

During the identity registration checking sub-phase (IRCP), each vehicle consults the closest RSU to check if its identity is successfully recorded and distributed between NetDBs (figure 4-5). To do so, the vehicle sends a checking request (ChRq) signed with $Prk_{Sig(v)}$ (algorithm 2).

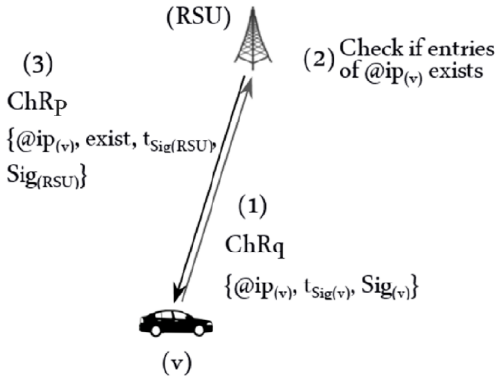


Figure 4-5: Identity Registration Checking sub-phase

When the RSU receives ChRq message, it searches in its NetDB for the corresponding entry, then verifies the signature received in ChRq us-

ing the v's public signing key $Pk_{Sig(v)}$ (existed in the entry). If ChRq is valid (other cases discussed in section 4.4), the RSU sends a checking reply (ChRp) including 'exist' flag and signed with its private signing key $Prk_{Sig(rsu)}$ (algorithm 2). 'exist' flag can be true if the identity is recorded and distributed successfully between NetDBs, or false if it is not recorded. The RSU calculates Sig_{rsu} according to the value of 'exist' flag.

Algorithm 2 IRCP sub-phase algorithm

V: $Sig_V \leftarrow Sig_{(v)}$ $(@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(v)})$
$V \rightarrow$ *RSU: ChRq* $\{@ip_{(v)}, t_{Sig(v)}, Sig_V\}$
RSU : **if** $t - t_{Sig(v)} < \Delta T$ **then**
if *NetDb.entryExist* $(@ip_{(v)})$ **then**
$Hsh_{RSU} \leftarrow h\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(v)})$
$Hsh_V \leftarrow de_{PkSig(v)}\ (Sig_V)$
if $Hsh_{RSU} = Hsh_V$ **then**
exist $\leftarrow$ *true*
$Sig_{RSU} \leftarrow Sig_{PrkSig(rsu)}\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, exist, t_{Sig(rsu)})$
Else *no response (ChRp)*
End if
Else
exist $\leftarrow$ *false*
$Sig_{RSU} \leftarrow Sig_{PrkSig(rsu)}\ (@ip_{(v)}, exist, t_{Sig(rsu)})$
End if
Else *no response (ChRp)*
End if
RSU $\rightarrow$ *V: ChRp* $\{@ip_{(v)}, exist, t_{Sig(rsu)}, Sig_{RSU}\}$
V: **if** $t - t_{Sig(rsu)} < \Delta T$ **then**
if *exist* **then**
$Hsh_V \leftarrow h\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, exist, t_{Sig(rsu)})$
Else $Hsh_V \leftarrow h\ (@ip_{(v)}, exist, t_{Sig(rsu)})$
End if
$Hsh_{RSU} \leftarrow de_{PkSig(rsu)}\ (Sig_{(rsu)})$
if $Hsh_V = Hsh_{RSU}$ **then**
if *exist* **then** *real identity is recorded*
Else *repeat* **IRRP**
End if
Else
Repeat **IRCP** *for number of times*
if *no valid response* **then** *repeat* **IRRP**
End if
End if
Else
Repeat **IRCP** *for number of times*
if *no valid response* **then** *repeat* **IRRP**
End if
End if

After receiving the ChRp message and verifying $t_{Sig(rsu)}$ and validity of the signature, the vehicle checks the value of exist flag. If true, it ensures that its identity is recorded with success in the network database NetDB, and thereby, other nodes can obtain its identity by contacting any RSU in the network. If exist flag is false, the identity is not recorded and the vehicle should repeat the IRRP sub-phase.

4.5.2.2 Encrypted Identity Registration Phase (EIRP)

EIRP has the same principle as the identity registration phase (IRP) but in an encrypted way. This phase is launched just one time for a node. It is used in some cases when IRP is failed, in which an attacker registers falsely the v's identity during IRRP (as discussed in section 4.6.1). In that case, v cannot receive RegRp, so it continues to send RegRq (till receiving the RegRp from an RSU). When the RSU receives RegRq many times from v while it finds v's identity is registered (falsely) in the NetDB, it must check if the registered v's identity is fake or real by launching the EIRP phase.

We propose the EIRP process thanks to two advantages of VANet: the mobility of vehicles and the characteristics of urban environments. Movement of vehicles and density of the network in urban environments most of the time help the vehicle to have new neighbours over time and contact different RSUs from different positions according to its displacement.

For that, when the RSU receives RegRq many times, it sends to that vehicle (according to the IP address of v) a checking neighbours (ChNgh) message. The RSU must contact v using two or more routes (in each route, the previous node pr -the neighbour- to v must be different from the other pr in other routes). If all routes from RSU to v have the same pr to v, the EIRP is cancelled, and consequently, RSU will continue to contact v till it has two pr or more (in routes from RSU to v). ChNgh contains $@ip_{(rsu)}$ and the ciphertext of $@ip_{(rsu)}$ and R using $Pk_{Sig(v)}$ (after using $Prk_{Sig(rsu)}$), in which R is a random number generated by RSU (figure 4-6).

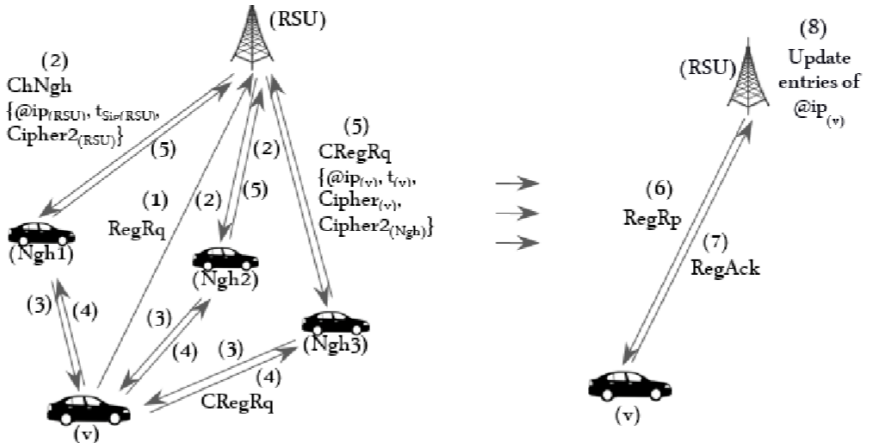


Figure 4-6: Encrypted Identity Registration Request sub-phase

At reception, v decrypts the ChNgh message to get R and to verify if the sender is the real RSU. Vehicle v responds RSU with an encrypted registration request (CRegRq) message which includes $@ip_{(v)}$ and the ciphertext of ip(v), $Pk_{DH(v)}$, $Pk_{Sig(v)}$, R and ip(Ngh), in which Ngh is the previous node that sends ChNgh to v. This ciphertext is generated using $Pk_{Sig(rsu)}$. When the node Ngh receives CRegRq, it adds its encrypted IP address to this message using $Pk_{Sig(rsu)}$ (after using $Prk_{Sig(ngh)}$). After that, the node Ngh forwards the CRegRq message to the RSU.

Algorithm 3 EIRP phase algorithm

V→ RSU: RegRq {@ip(v), PKDH(v), PKSig(v)}
RSU: Nb_Ngh ← 0 ; //Number of routes with different neighbours to v
RSU: Cipher1(RSU) ← dPrkSig(rsu) (R, t(RSU))
Cipher2(RSU) ← ePKSig(v) (Cipher1(RSU))
// Contact v through 2 or more different routes
RSU → V: ChNgh {@ip(RSU), tSig(RSU), Cipher2(RSU)}
*V : **if** t - t(RSU)<ΔT **then***
Cipher1(RSU) ← dPrKSig(v) (Cipher2(RSU))
R, t'(RSU) ← ePKSig(RSU) (Cipher1(RSU))
Cipher(v) ← ePHSig(rsu) (@ip(v), PKDH(v), PKSig(v), R, @ip(Nhg), t(v))
***Else** repeat **EIRP** after an interval of time*
End if
V → Ngh: CRegRq {@ip(v), t(v), Cipher(v)}
Ngh: Cipher1(Ngh) ← dPrkSig(Ngh) (@ip(Ngh))
Cipher2(Ngh) ← ePKSig(RSU) (Cipher1(Ngh))
Ngh → RSU: CRegRq {@ip(v), t(v), Cipher(v), Cipher2(Ngh)}
*RSU : **if** t - t(v)<ΔT **then***
@ip(v), PKDH(v), PKSig(v), R, @ip(Nhg), t'(v) ← dPrKSig(RSU) (Cipher(v))
***if** R is the same **then** //ChNgh and CRegRq are not altered*
Cipher1(Ngh) ← dPrKSig(RSU) (Cipher2(Ngh))
@ip'(Ngh) ← ePKSig(Ngh) (Cipher1(Ngh))
if** @ip(Ngh)=@ip'(Ngh) **and** @ip(Ngh)!=@ip(v) **then
// Ngh is a real neighbour to v
Nb_Ngh ← Nb_Ngh+1;
***Else** repeat **EIRP** after an interval of time*
End If
***Else** repeat **EIRP** after an interval of time*
End If
***Else** repeat **EIRP** after an interval of time*
End If
// Repeat steps (4-9) for each route
***if Nb_Ngh >= 2 then** RSU continues **EIRP** in the same way as **IRP** by sending **RegRp** to v using one route.*
*At reception of **RegAck**, RSU replaces v's identity with the new one.*

At reception, the RSU decrypts each CRegRq it received and verify its validity (algorithm 3). At the same time, the RSU counts the number of different IP addresses of Ngh nodes according to the chosen routes to v. The RSU must receive the CRegRq message from v through two routes having two different neighbours to v at least. In that case, the RSU sends a RegRp to v as described in the identity registration request sub-phase (IRRP), then, the identity registration checking sub-phase (IRCP) is launched in the same way as in the IRP phase.

The EIRP method is based in two parameters to detect if the identity registration is attacked or not: R is used to ensure that ChNgh and CRegRq messages are not altered, and Nb_Ngh to take into account the views of neighbours. If M who sends/alters RegRq so it can get R but it cannot have more than one neighbour (itself) else it cannot get R.

4.5.2.3 Lightweight Diffie-Hellman Exchange Phase

After verifying the registration of identities in network databases, vehicles can start establishing encrypted communication by calculating Sk_{DH} key and then Sk key. Contrary to classical use of Diffie-Hellman method, in which the source and destination nodes exchange directly Pk_{DH} keys, in the proposed approach, these nodes consult network databases to obtain the public key of each other.

Taking an example of two vehicles S and D that want to communicate with each other. Before consulting a NetDB to obtain the $Pk_{DH(S)\_i}$ key, D should know which $Pk_{DH(S)\_i}$ key's index is chosen by S. For that, S sends an index (i) referencing the chosen $Pk_{DH(S)\_i}$ in the NetDB database. According to the proposed approach, the two nodes can know which Pk_{DH} keys are chosen by each other during the routing process. It is supposed that indexes are exchanged at the network level by using control messages of the routing protocol (CMRP). Indexes have small sizes that do not affect network performances.

The way of exchange of indexes is different according to the routing protocol used, in which routes can be reactively or proactively discovered. For reactive protocols like AODV, indexes can be exchanged using control messages. A source can send its chosen index by the route request (RREQ) message, and destination can reply by route a reply (RREP) message including its index. For this type of protocols, only nodes that want to communicate, exchange these indexes. In proactive protocols like OLSR, Pk_{DH} indexes are shared at the creation of the network using the topology control (TC) and Hello messages to reach all nodes in the network.

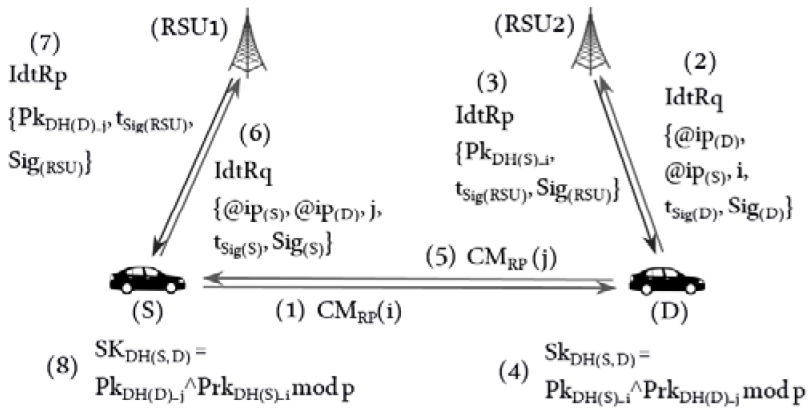


Figure 4-7: Lightweight Diffie-Hellman Exchange Phase

After transferring indexes (as represents figure 4-7), S and D know which Diffie-Hellman public key is chosen by the other node. In this case, each of them consults the closest RSU to obtain the appropriate Pk_{DH_i} by sending the identity Request (IdtRq). Supposing that the vehicle D want to obtain $Pk_{DH(S)}$ key, it sends the IdtRq message signed with $Prk_{Sig(D)}$ and including the index chosen by the vehicle S as described in algorithm 4.

At the reception of IdtRq, RSU searches for the corresponding entry in its NetDB after verifying the time and signature. Then it resends to D an Identity Reply message (IdtRep) including the $Pk_{DH(S)\_i}$, time of sending and its signature.

After receiving Pk_{DH} keys, S and D can calculate Sk_{DH} as below:

$$S : Sk_{DH} = (g^{Prk_{DH(D)\_j}} \ mod \, p)^{Prk_{DH(S)\_i}} \ mod \, p$$

$$D : Sk_{DH} = (g^{Prk_{DH(S)\_i}} \ mod \, p)^{Prk_{DH(D)\_j}} \ mod \, p$$

Algorithm 4 LDHEP phase algorithm

$S \rightarrow D$: CM_{RP} $\{i_{(PkDH(S))}\}$
D: $Sig_D \leftarrow Sig_{PrkSig(D)}$ $(@ip_{(D)}, @ip_{(S)}, i, t_{Sig(D)})$
$D \rightarrow RSU$: $IdtRq$ $\{@ip_{(D)}, @ip_{(S)}, i, t_{Sig(D)}, Sig_D\}$
RSU: **if** $t - t_{Sig(D)} < \Delta T$ **then**
  $Hsh_{RSU} \leftarrow h$ $(@ip_{((D)}, @ip_{(S)}, i, t_{Sig(D)})$
  $Hsh_D \leftarrow de_{PkSig(D)}$ (Sig_D)
  if $Hsh_D = Hsh_{RSU}$ **then**
    $entry \leftarrow NetDb.getEntry(@ip_{(S)}, i)$
    $Pk_{DH\_i} \leftarrow entry.getPk_{DH}()$
    $Sig_{RSU} \leftarrow Sig_{PrkSig(rsu)}$ $(@ip_{(D)}, @ip_{(S)}, Pk_{DH\_i}, t_{Sig(rsu)})$
  Else no response (IdtRp)
  End if
Else no response (IdtRp)
End if
$RSU \rightarrow D$: $IdtRp$ $\{Pk_{DH\_i}, t_{Sig(rsu)}, Sig_{RSU}\}$
D: **if** $t - t_{Sig(rsu)} < \Delta T$ **then**
  $Hsh_V \leftarrow h$ $(@ip_{(D)}, @ip_{(S)}, Pk_{DH\_i}, t_{Sig(rsu)})$
  $Hsh_{RSU} \leftarrow de_{PkSig(rsu)}$ (Sig_{RSU})
  if $Hsh_V = Hsh_{RSU}$ **then**
    $Sk_{DH} \leftarrow (Pk_{DH(S)\_i})^{\wedge}Prk_{DH(D)} \bmod P$
  Else consults again same or other RSU
  End if
Else consults again same or other RSU
End if
$D \rightarrow S$: CM_{RP} $\{j_{(PkDH(D))}\}$
S: $Sig_S \leftarrow Sig_{PrkSig(S)}$ $(@ip_{(S)}, @ip_{(D)}, j, t_{Sig(S)})$
$S \rightarrow RSU$: $IdtRq$ $\{@ip_{(S)}, @ip_{(D)}, j, t_{Sig(S)}, Sig_S\}$
RSU: **if** $t - t_{Sig(S)} < \Delta T$ **then**
  $Hsh_{RSU} \leftarrow h$ $(@ip_{(S)}, @ip_{(D)}, j, t_{Sig(S)})$
  $Hsh_S \leftarrow de_{PkSig(S)}$ (Sig_S)
  if $Hsh_S = Hsh_{RSU}$ **then**
    $entry \leftarrow NetDb.getEntry(@ip_{(D)}, j);$
    $Pk_{DH\_j} \leftarrow entry.getPk_{DH}();$
    $Sig_{RSU} \leftarrow Sig_{PrkSig(rsu)}$ $(@ip_{(S)}, @ip_{(D)}, Pk_{DH\_j}, t_{Sig(rsu)})$
  Else no response (IdtRp).
  End if
Else no response (IdtRp).
End if
$RSU \rightarrow S$: $IdtRp$ $\{Pk_{DH}j, t_{Sig(rsu)}, Sig_{RSU}\}$
S: **if** $t - t_{Sig(rsu)} < \Delta T$ **then**
  $Hsh_V \leftarrow h$ $(@ip_{(S)}, @ip_{(D)}, Pk_{DH\_j}, t_{Sig(rsu)})$
  $Hsh_{RSU} \leftarrow de_{PkSig(rsu)}$ (Sig_{RSU})
  if $Hsh_V = Hsh_{RSU}$ **then**
    $Sk_{DH} \leftarrow (Pk_{DH(D)\_j})^{\wedge}Prk_{DH(S)} \bmod P$
  Else consults again same or other RSU
  End if

4.5.2.4 Encrypted Communication Phase (ECP)

Once the ECP phase is reached (algorithm 5), It is ensured that all previous phases are executed legitimately without attacks, and thereby, S and D (example in previous phase) have calculated the true Sk_{DH} key. After that, the two nodes generate MASQUE with a same value to calculate the Sk key. Sk is calculated by XOR operation between MASQUE and Sk_{DH}. The MASQUE value is changed for each message for the two nodes in order to reduce the possibility of attack.

Algorithm 5 ECP phase algorithm

Let n, m ∈ N / n > m
S: MASQUE ← Cut(0, m-1) (Pk_DH(S)) + Cut(mSize(PkDH(D))−n+m, mSize(PkDH(D))-1) (Pk_DH(D))
Sk ← Sk_DH ⊕ MASQUE
D: MASQUE ← Cut(0, m-1) (Pk_DH(S)) + Cut(mSize(PkDH(D))−n+m, mSize(PkDH(D))-1) (Pk_DH(D))
Sk ← Sk_DH ⊕ MASQUE
S: // encrypt data to send it to D
encData ← e_Sk (Data)
// generate MASQUE and then Sk for encryption of next message
MASQUE ← Cut(0, m-1) (Data) + Cut(mSize(Data)−n+m, mSize(Data)-1) (Data)
Sk ← Sk_DH ⊕ MASQUE
Sig_S ← Sig_PrkSig(s) (encData + t_Sig(s))
S → D: encMsg {Sig_S, t_Sig(s), encData}
D: **if** *t - t_Sig(s) < ΔT* **then**
Hsh_D ← h (encData + t_Sig(s))
Hsh_S ← de_PkSig(s) (Sig_S)
if *Hsh_D = Hsh_S* **then**
Data ← de(Sk) (encData)
MASQUE ← Cut(0, m-1) (Data) + Cut(mSize(Data)−n+m, mSize(Data)-1) (Data)
Sk ← Sk_DH (+) MASQUE
Else *// forged message*
No response
End if
Else *// Reply attack*
No response
End if
// Similar process when D sends message to S
// MASQUE and Sk are calculated after each message and used for encryption and decryption of next message

First time, when S and D get Pk_{DH} keys of each other, MASQUE can be generated for both of them as following in equation 4.1.

$$MASQUE_{(S)} = MASQUE_{(D)}$$
$$MASQUE_{(S)} = first\ m\ bits\ (Pk_{DH(S)}) + last\ n-m\ bits\ (Pk_{DH(D)}) \quad (4.1)$$

$//n$: *number of bits of MASQUE*

After that, each node calculates Sk, and then, encrypted communication can be started. The encryption process is based on a symmetric algorithm, in which TCP and UDP packets are encrypted using Sk. An authentication header including the time of sending and signature is added to the encrypted packet as represented in figure 4-8.

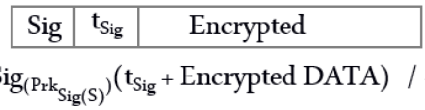


Figure 4-8: Encrypted message format

During the communication, S and D recalculate Sk by regenerating MASQUE as describes the equation 4.2.

$$MASQUE = first\ m\ bits + last\ (n-m)\ bits\ of\ clear\ data\ of\ the\ previous\ message \tag{4.2}$$

At the reception of the encrypted message, D reads the time field to check if a replay attack has been launched, and then, verifies the signature of the message.

4.5.2.5 Identity Update Phase (IUP)

When a vehicle wants to update its identity (for security measures) by changing its Pk_{DH} or/and Pk_{Sig} (in the NetDB), it can send an update request (UpdRq) to the RSU (figure 4-9). The UpdRq message contains the new identity and it is signed by $Prk_{Sig(v)}$ (algorithm 6).

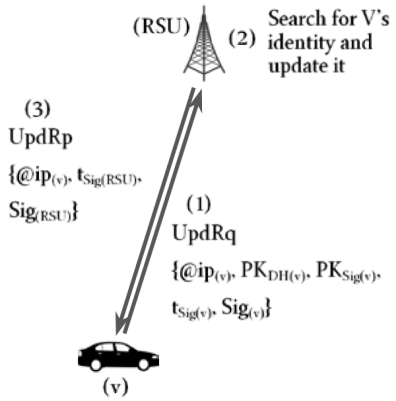


Figure 4-9: Identity Update Phase

After verifying the validity of UpdRq, the RSU searches for its corresponding identity in the NetDB and updates it according to UdpRq's

content. After that, RSU confirms the update by sending an update reply (UpdRp) to v and requests other NetDBs to update v's identity.

Algorithm 6 IUP phase algorithm

$V: Pk_{DH} \leftarrow \{Pk_{DH\_1}, Pk_{DH\_2}, ..., PK_{DH\_n}\}$
$V \rightarrow RSU: UpdRq\ \{@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(v)}, Sig_{(v)}\}$
RSU: **if** $t - t_{Sig(v)} < \Delta T$ **then**
$Hsh_{RSU} \leftarrow h\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(v)})$
$Hsh_V \leftarrow de_{PkSig(v)}\ (Sig_V)$
if $Hsh_{RSU} = Hsh_V$ **then**
For each entry with $ip_{(v)}$
Drop this entry
End for each
For each Pk_{DH_i}
$NetDb.addEntry\ (@ip_{(v)}, Pk_{DH\_i}, i, Pk_{Sig(v)}, t\_Lt)$
End for each
$SigRSU \leftarrow Sig_{PrkSig(rsu)}\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(rsu)})$
End if
Else no response (UpdRq)
End if
$RSU \rightarrow V: UpdRp\ \{@ip_{(v)}, t_{Sig(rsu)}, Sig_{RSU}\}$
V: **if** $t - t_{Sig(RSU)} < \Delta T$ **then**
$HshV \leftarrow h\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(RSU)})$
$HshRSU \leftarrow de_{PkSig(RSU)}\ (Sig_{RSU})$
if $Hsh_V = Hsh_{RSU}$ **then** update the identity
Else repeat **IUP**
End if
Else repeat **IUP**
End if

4.5.2.6 Identity Revocation Phase (IVP)

When the vehicle wants to revoke its identity (in rare cases when the vehicle will not be used anymore for example), it can contact the nearest RSU by sending a revocation request (RevRq) signed with $Prk_{Sig(v)}$ (figure 4-10).

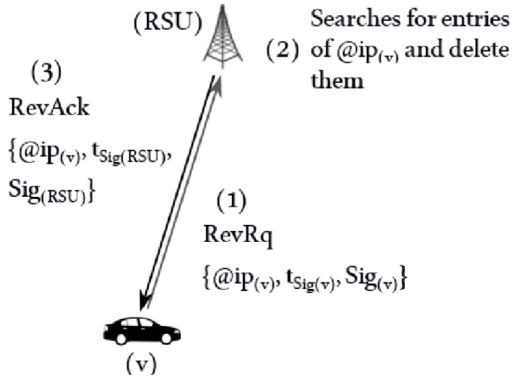


Figure 4-10: Identity revocation Phase

At reception, RSU searches in its NetDB for the corresponding entry, then verifies the signature received in the RevRq message using $Pk_{Sig(v)}$ (existed in the entry in the NetDB). If RevRq is valid (other cases discussed in section 4.6.4), the RSU deletes the corresponding entries from the NetDB and sends a revocation acknowledgement (RevAck) signed with $Prk_{Sig(RSU)}$ as detailed in algorithm 7.

Algorithm 7 IVP phase algorithm

V: $Sig_V \leftarrow Sig_{(v)}$ $(@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(v)})$
$V \rightarrow RSU$: RevRq $\{@ip_{(v)}, t_{Sig(v)}, Sig_V\}$
RSU : **if** $t - t_{Sig(v)} < \Delta T$ **then**
$Hsh_{RSU} \leftarrow h\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(v)})$
$Hsh_V \leftarrow de_{PkSig(v)}\ (Sig_V)$
if $Hsh_{RSU} = Hsh_V$ **then**
$Sig_{RSU} \leftarrow Sig_{PrkSig(rsu)}\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(rsu)})$
For each entry with $@ip_{(v)}$
Drop this entry
End for each
Else no response (RevAck)
End if
Else no response (RevAck)
End if
$RSU \rightarrow V$: RevAck $\{@ip_{(v)}, t_{Sig(rsu)}, Sig_{RSU}\}$
V: **if** $t - t_{Sig(rsu)} < \Delta T$ **then**
$Hsh_V \leftarrow h\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, t_{Sig(rsu)})$
$Hsh_{RSU} \leftarrow de_{PkSig(rsu)}\ (Sig_{(rsu)})$
if $Hsh_V = Hsh_{RSU}$ **then**
V's identity is revoked successfully
Else
Repeat **IVP** for a number of times
if no valid response **then**
Change route to RSU or consult another one
End if
End if
Else
Repeat **IVP** for a number of times
if no valid response **then**
Change route to RSU or consult another one
End if
End if

After receiving the RevAck message and verifying $t_{Sig(RSU)}$ and validity of the signature with no error, the vehicle ensures that its identity is revoked and deleted successfully from the consulted NetDB. In parallel, the RSU sends a request to other NetDBs to delete the identity of this vehicle.

The following part presents the analysis that shows the security of the proposed model.

4.6 Security analysis

This part shows the security of the proposed model against MITM and replay attacks in different situations. According to the proposed communication design, MITM attack can be launched in different ways as well as at different times to recover the session key and decrypt messages. Similarly, malicious nodes can launch a replay attack to reuse any original message of legitimate nodes according to their aims.

Facing these attacks in this approach, a signature mechanism and a communication design are used to detect them at the launch time. We analyze four cases of attacks and we show their failures according to the proposed model in the next sections.

4.6.1 Attack 1 during the IRRP sub-phase

4.6.1.1 Attack process

The attacker M can intervene during the identity registration request sub-phase IRRP between a vehicle v and RSU. M aims to change the Pk_{DH} key of v in RegRq to record it falsely in the NetDB. As represent figure 4-11, when M intercepts the registration request RegRq of v, it replaces $Pk_{DH(v)}$ by $Pk_{DH(M)}$ and $Pk_{Sig(v)}$ by $Pk_{Sig(M)}$. Consequently, when a node looks for identity of v, it obtains $Pk_{DH(M)}$ instead of $Pk_{DH(v)}$.

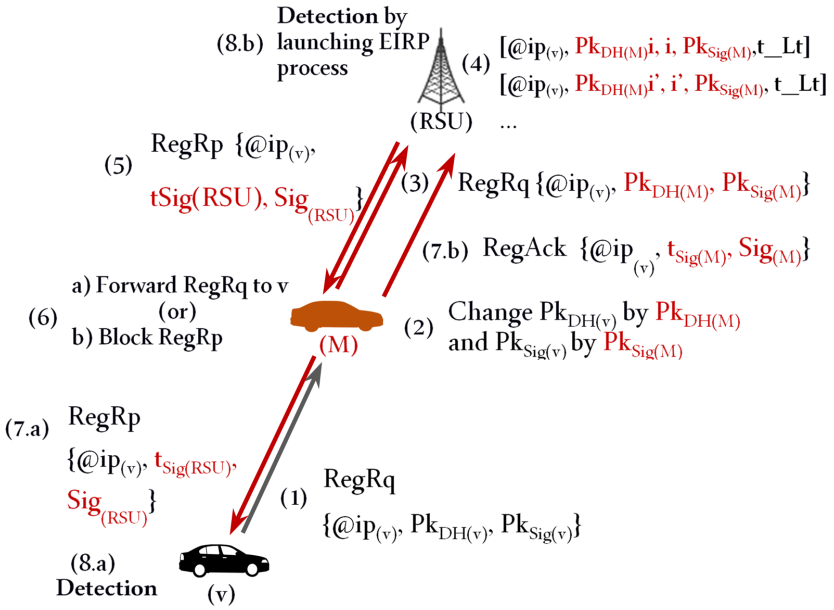


Figure 4-11: MITM attack during the IRRP sub-phase

This attack can be launched for different vehicles in the network. Supposing that two vehicles v1 and v2 have been attacked, when they want to calculate Diffie-Hellman secret key, the attacker M can intervene and calculate $Sk_{DH}(M, v1)$ and $Sk_{DH}(M, v2)$ to encrypt/decrypt messages with v1 and v2 respectively.

Another way the attacker can use is not to forward the registration reply (RegRp) to v but blocks it and resends a registration acknowledgement (RegAck) to RSU. In this case, the RSU registers the false identity of v in the network database NetDB.

Figure 4-11 and algorithm 8 represent the attack during the IRRP sub-phase when M intervenes to alter the RegRq message or block it to send RegAck to RSU.

Algorithm 8 Algorithm of attack 1 during the IRRP sub-phase

(M alters RegRq and forwards RegRp to V)

(2) M: replace $Pk_{DH(v)}$ *with* $Pk_{DH(M)}$

(3) $M \rightarrow$ *RSU: RegRq* $\{@ip_{(v)}, Pk_{DH(M)}, Pk_{Sig(M)}\}$

(4) RSU: // record falsely M's identity temporary as V

For each $Pk_{DH(M)\_i}$

$NetDb.addEntry(@ip_{(v)}, Pk_{DH(M)\_i}, i, Pk_{Sig(M)}, t\_Lt)$

End for each

$SigRSU \leftarrow SigPkSig(rsu)\ (@ip(v), PkDH(M), PkSig(M), tSig(rsu))$

(5) (6.a) (7.a) RSU $\rightarrow M \rightarrow$ *V: RegRp* $\{ip_{(v)}, t_{Sig(rsu)}, Sig_{RSU}\}$

(M alters RegRq and sends RegAck to RSU)

(2), (3), (4) and (5) are the same

(6.b) M: Blocks RegRp

(7.b) $M \rightarrow$ *RSU :ReqAck* $\{@ip_{(v)}, t_{Sig(M)}, Sig_{(M)}\}$

4.6.1.2 Detection and solution process

It is supposed that RSU responds the vehicle v with RegRp (algorithm 9). When M alters RegRq, the RSU records temporarily the identity of v as received (in RegRq), then sends RegRp including signature of false received information to v. At reception, v checks RegRp sending time according to the time threshold to detect Replay attack, then verifies the correctness of signature according to information that it sends in RegRq. Here, the vehicle v can discover the incoherence of the message and detect that its communication has been attacked. In this case, as a solution, the vehicle repeats the IRRP sub-phase by changing the route to the same RSU, or contacts other RSU to avoid the attacker as well as possible.

Algorithm 9 Detection and solution algorithm for Attack 1 during the IRRP sub-phase

(When M forwards RegRp to V)
// RSU records temporary @ip(v), Pk_DH(M), Pk_Sig(M) as identity of V and sends RegRp {ip(v), t_Sig(RSU), Sig_RSU}
(8.a) V: ***if*** $t - t_{Sig(RSU)} < \Delta T$ ***then***
$Hsh_V \leftarrow h$ (@ip(v), $Pk_{DH(v)}$, $Pk_{Sig(v)}$, $t_{Sig(RSU)}$)
$Hsh_{RSU} \leftarrow de_{PkSig(RSU)}$ ($Sig_{(RSU)}$)
if $Hsh_V = Hsh_{RSU}$ ***then***
//RegRq and RegRp are valid
Send RegAck to RSU
Else => Detection of MITM attack
Repeat ***IRRP****. (change route to RSU or consult other one)*
End if
Else => Detection of Reply attack
Repeat ***IRRP****. (change route to RSU or consult other one)*
End if
(When M sends RegAck to RSU)
8.b RSU: launches ***EIRP*** *process*

When the second attack is launched, RSU will register v's false identity with no detection. In the same time, v is waiting for RegRp from RSU (but it is blocked by M), so v will continue to send RegRq to RSU. Taking into account the mobility of v and urban environment characteristics (as discussed in section 4.5.2.2), v can have new neighbours over time and contact different RSUs. This process can be repeated many times until no attacker presented (by discovering by v that the sent RegRq is coherent).

In parallel, RSU receives many RegRqs from v while it finds v's identity registered in the NetDB. In that case, RSU must check for the first and last time if the registered identity is fake or real, so it launches the EIRP process.

In the end, it is ensured that each vehicle can detect if the RSU has received the real identity or the falsified one altered by the attacker. Therefore, any vehicle does not confirm the registration of the false identity, and consequently, the NetDB database deletes it after its short lifetime expires (t_Lt).

4.6.2 Attack 2 during the IRRP sub-phase

4.6.2.1 Attack process

In this scenario, the attacker intervenes when the vehicle sends the registration acknowledgement (RegAck) to the RSU. After recording the temporary identity of v, RSU waits for v's RegAck.

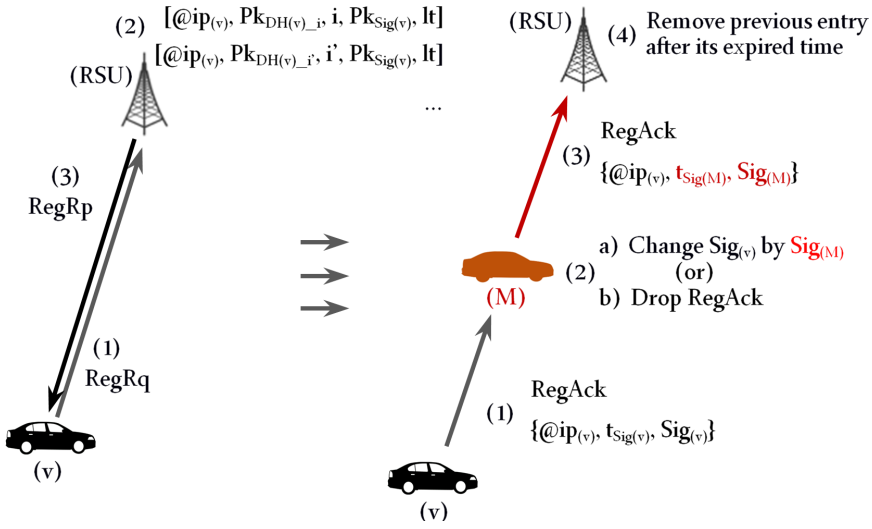


Figure 4-12: Attack during the IRRP sub-phase

As shown in figure 4-12, v responds the RSU by a RegAck message to record its identity for a long lifetime in the NetDB. In this case, M can launch the attack in two ways: drops RegAck or alters its content to make it invalid and then ignored by the RSU (algorithm 10). M aims to prevent the RSU to receive the valid confirmation from v, and consequently, the RSU deletes the temporary recorded entry of v.

Algorithm 10 Algorithm of attack 2 during the IRRP sub-phase

(M alters RegAck)

M: change $Sig_{(v)}$ by $Sig_{(M)}$
$Sig_M \leftarrow Sig_{(M)}$ ($@ip_{(v)}$, $Pk_{DH(M)}$, $Pk_{Sig(M)}$, $t_{Sig(M)}$)
M → RSU: RegAck {$@ip_{(v)}$, $t_{Sig(M)}$, Sig_M}
RSU: ***if*** *t - $t_{Sig(M)}$ < ΔT* ***then***
$Hsh_{RSU} \leftarrow Hash$ ($@ip_{(v)}$, $Pk_{DH(v)}$, $Pk_{Sig(v)}$, $t_{Sig(M)}$)
$Hsh_V \leftarrow De_{PkSig(v)}$ (Sig_M)
if *Hsh_{RSU} = Hsh_V* ***then***
// Record previous entries (of V) for a long life time
NetDb.getEntries($@ip_{(v)}$).setLifeTime()
Else if *NetDb.getEntry($@ip_{(v)}$).t_Lt = 0* ***then***
Drop this entry
End if
End if

(M drops RegAck)

M: Drop RegAck
RSU: ***For each*** *NetDb.getEntry().t_Lt = 0*
Drop this entry
End for each
// The entry of V is one of these entries when RSU has not received
$RegRq_{(v)}$ and t_Lt = 0

4.6.2.2 Detection and solution process

Even if the attacker dropped or altered RegAck message, the RSU deletes real identities (saved temporary) because of no reception for valid RegAck. For that, v has to verify if its identity is recorded or deleted.

Responding to this problem, the identity registration checking sub-phase (IRCP) sub-phase is added in order to verify the registration of vehicles identities (figure 4-13 and algorithm 11). After sending RegAck to RSU, v waits for a period of time (during which its identity is shared with other RSUs). After this waiting time, v can launch the IRCP sub-phase by consulting the closest NetDB.

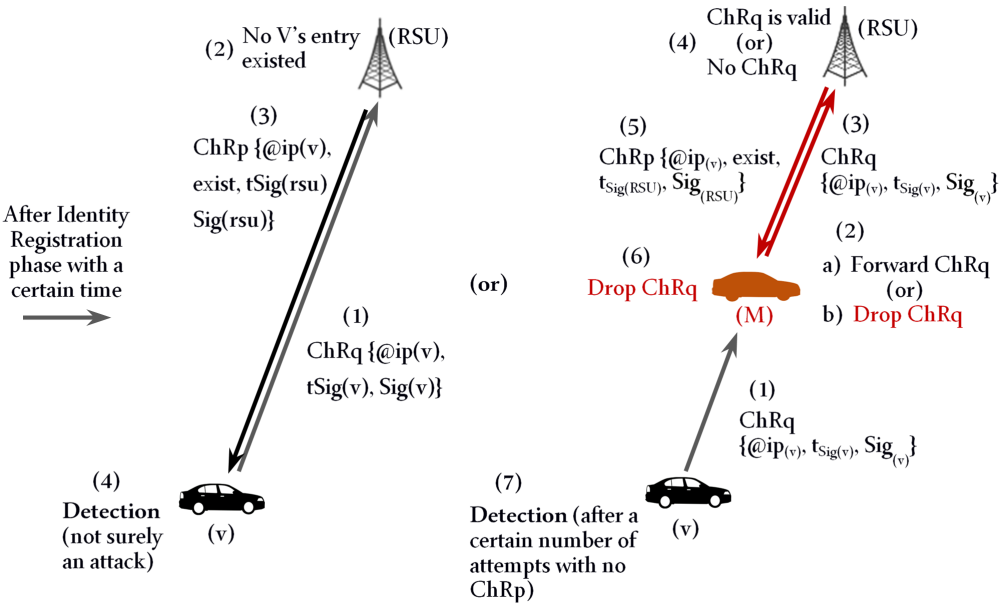


Figure 4-13: Detection of previous attack (with/no attack) during IRCP phase

After receiving the checking request (ChRq) from v, the RSU verifies the existence of v's identity. Due to the previous attack, the identity is deleted, so the RSU responds to the vehicle that its entry does not exist. In this case, v knows that its RegAck was not received. For that, v repeats the IRRP sub-phase to register its identity again.

Another case is when an attacker is present, it can drop or alter ChRq message. Consequently, The RSU will not receive a valid message and thereby, no response (ChRp). After a waiting time (with no ChRp is received from RSU), v repeats IRCP sub-phase for a number of times by changing the route to the RSU or consulting another one. If no ChRp received, v detects that an attacker is present, then it restarts the IRRP sub-phase.

Algorithm 11 Detection and solution algorithm for attack 2 during the IRRP sub-phase

(With no attack)

(4) V : ***if*** $t - t_{Sig(rsu)} < T$ ***then***
if *exist* ***then***
$Hsh_V \leftarrow h\ (@ip_{(v)}, Pk_{DH(v)}, Pk_{Sig(v)}, exist, t_{Sig(rsu)})$
Else $Hsh_V \leftarrow h\ (@ip_{(v)}, exist, t_{Sig(rsu)})$
End if
$Hsh_{RSU} \leftarrow de_{PkSig(rsu)}\ (Sig_{RSU})$
if $Hsh_V = Hsh_{RSU}$ ***then***
if *exist* ***then***
Real identity is recorded
Else => Possibility of MITM attack *during* ***IRRP***
Repeat ***IRRP***
End if
Else *repeat* ***IRCP*** *for a number of times (change route to the RSU or consult another one)*
End if
if *no valid response* ***then***
Repeat ***IRRP***
End if
Else => Detection of Reply attack
Repeat ***IRCP*** *for a number of times (change route to the RSU or consult another one)*
if *no valid response* ***then***
Repeat IRRP
End if
End if

(With attack)

(7) V: ***if*** *no ChRq from RSU (after a waiting time)* ***then***
Repeat ***IRCP*** *for a number of times (change route to the RSU or consult another one)*
if *no valid response* ***then***
Repeat ***IRRP***
End if
End if

4.6.3 Attack 3 during the Lightweight Diffie-Hellman Exchange Phase

4.6.3.1 Attack process

Another possibility of attack can happen when it is launched during the LDHEP phase. The attacker can intervene between S and D when they exchange indexes of Pk_{DH} keys. The attacker can intercept these indexes to obtain the requested Pk_{DH} keys of S and D.

According to the proposed model, one malicious node cannot succeed to perform the MITM attack during this phase. Indexes are exchanged through control messages of the routing protocol (CMRP) that contain by

default the IP address of the node (S or D).

To succeed the attack, M should change CMRP content and put its IP address and its chosen index to push the receiver node to obtain its $Pk_{DH(M)}$ key from the RSU, while this process does not match the MITM attack, because the receiver will detect easily the attacker M through its IP address in CMRP.

Another way to launch this attack during this phase is by the collaboration of multiple attackers, and acting together, in which each one of them performs a specific role in the attack and intervenes in the right time and right place as represents figure 4-14.

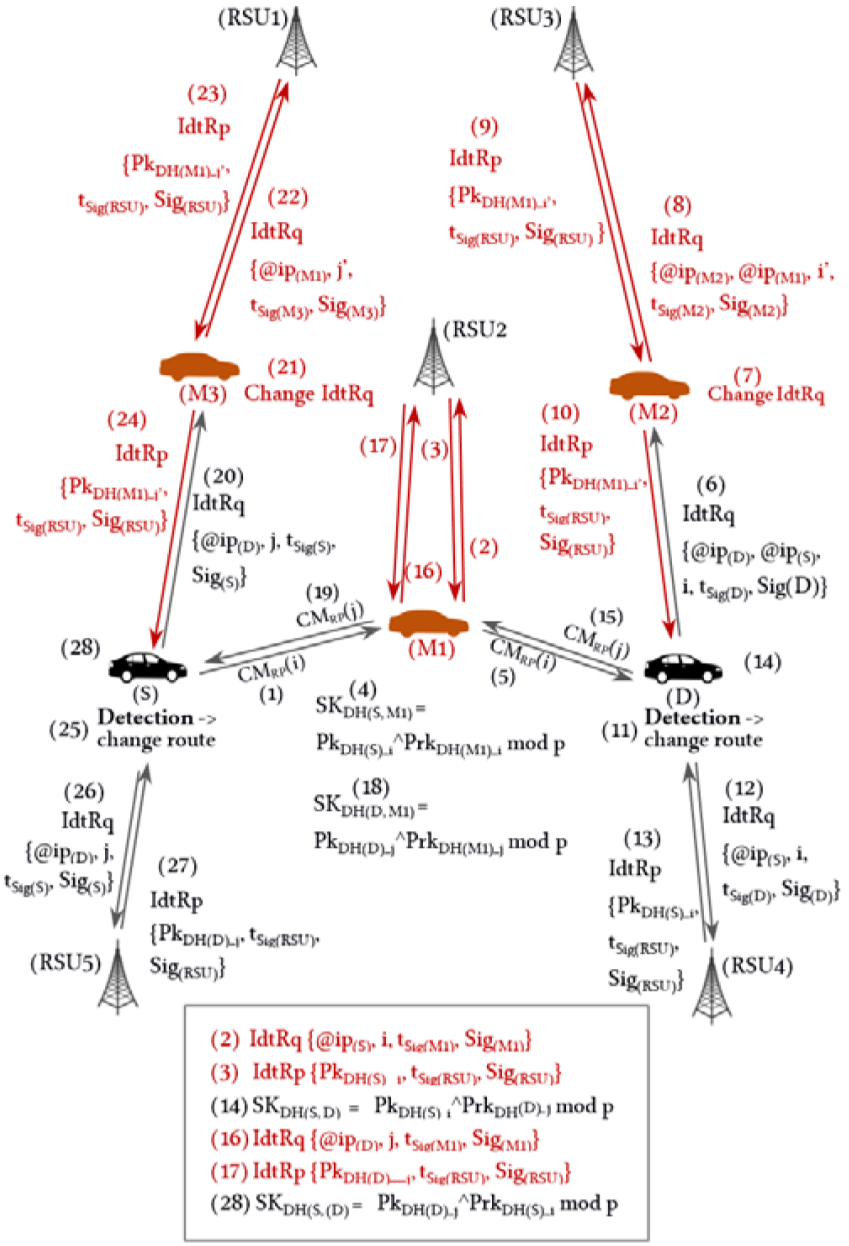


Figure 4-14: A collaborative attack during the LDHEP phase

Figure 4-14 shows a collaborative attack between three malicious nodes: M1 located between S and D, and M2 and M3 are located between vehicles and RSUs. In this scenario, the aim of this attack is to allow M1 to create two keys: $Pk_{DH}(M1, S)$ with S and $Pk_{DH}(M1, D)$ with D to decipher communication between them. Algorithm 12 details the attack.

Algorithm 12 Algorithm of attack 3 during the Lightweight Diffie-Hellman Exchange Phase

(Attack of M1)

(2) M1: Consults RSU to obtain $Pk_{DH(S)\_i}$,
$Sig_{M1} \leftarrow Sig_{PkSig(M1)}(@ip_{(M1)}, @ip_{(S)}, i, t_{Sig(M1)})$
$M1 \rightarrow RSU$: IdtRq $\{@ip_{(M1)}, @ip_{(S)}, i, t_{Sig(M1)}, Sig_{M1}\}$
(3) RSU: searches for $kK_{DH(S)\_i}$
$RSU \rightarrow M1$: IdtRp $\{Pk_{DH(S)\_i}, t_{Sig(rsu)}, Sig_{RSU}\}$
(16) M1: consults RSU to obtain $Pk_{DH(D)\_j}$,
$Sig_{M1} \leftarrow Sig_{PkSig(M1)}(@ip_{(M1)}, @ip_{(D)}, j, t_{Sig(M1)})$
$M1 \rightarrow RSU$: IdtRq $\{@ip_{(M1)}, @ip_{(D)}, j, t_{Sig(M1)}, Sig_{M1}\}$
(17) RSU: searches for $Pk_{DH(D)\_j}$
$RSU \rightarrow M1$: IdtRp $\{Pk_{DH(D)\_j}, t_{Sig(rsu)}, Sig_{RSU}\}$

(Attack of M2)

(7) M2: Change $@ip_{(D)}$ *with* $@ip_{(M2)}$, $@ip_{(S)}$ *with* $@ip_{(M1)}$ and i with i'
$Sig_{M2} \leftarrow Sig_{PkSig(M2)}(@ip_{(M2)}, @ip_{(M1)}, i', t_{Sig(M2)})$
(3) $M2 \rightarrow RSU3$: IdtRq $\{@ip_{(M2)}, @ip_{(M1)}, i', t_{Sig(M2)}, Sig_{M2}\}$
(4) RSU3: // verify validity of IdtRq and search for $Pk_{DH(M1)\_i'}$ *entry*
$Sig_{RSU} \leftarrow Sig_{PkSig(rsu)}(@ip_{(M2)}, @ip_{(M1)}, Pk_{DH(M1)\_i'}, t_{Sig(rsu)})$
(5) (6) $RSU3 \rightarrow M2 \rightarrow (RSU)$: IdtRp $\{Pk_{DH(M1)\_i'}, t_{Sig(rsu)}, Sig_{RSU}\}$

(Attack of M3)

(21) M3: Change $@ip_{(S)}$ *with* $@ip_{(M3)}$, $@ip_{(D)}$ *with* $@ip_{(M1)}$ and j with i'
$Sig_{M3} \leftarrow Sig_{PkSig(M3)}(@ip_{(M3)}, @ip_{(M1)}, i', t_{Sig(M3)})$
(3) $M3 \rightarrow RSU1$: IdtRq $\{@ip_{(M3)}, @ip_{(M1)}, i', t_{Sig(M3)}, Sig_{M3}\}$
(4) RSU1: // verify validity of IdtRq and search for $Pk_{DH(M1)\_j'}$ *entry*
$Sig_{RSU} \leftarrow Sig_{PkSig(rsu)}(@ip_{(M3)}, @ip_{(M1)}, Pk_{DH(M1)\_i'}, t_{Sig(rsu)})$
(5) (6) $RSU1 \rightarrow M3 \rightarrow S$: IdtRp $\{Pk_{DH(M1)\ i'}, t_{Sig(rsu)}, Sig_{RSU}\}$

Firstly, S sends CMRP to find a route to D. M1 intercepts the message to obtain the index of $Pk_{DH(S)}$, then forwards CMRP to D as represents figure 4-14. At the same time, M1 consults network database to get $Pk_{DH(S)\_i}$.

When D receives CMRP, it sends a IdtRq message to a NetDB database to obtain $Pk_{DH(S)\_i}$, while attacker M2 intercepts the IdtRq message and changes it according to the IP address and Pk_{DH}'s index (i') of M1, then forwards it to RSU3 to get $Pk_{DH(M1)\_i'}$ (figure 4-14). M2 collaborates with M1 to make D calculates the Sk_{DH} with M1 and not S. The next paragraph 'detection and solutions' shows how the vehicle D can detect this attack and obtain the right $Pk_{DH(S)\_i}$.

In parallel, when receiving CMRP from S, D resends CMRP with its chosen $Pk_{DH(D)}$'s index to S to confirm the route between them (from a reactive routing point of view).

Similarly, the attacker M1 can get the index and then consult a NetDB to obtain the $Pk_{DH(D)\_j}$ key as represented in figure 4-14.

After receiving CMRP from D, S does the same process as D to get $Pk_{DH(D)\_j}$ and in the same way M2 acts, M3 does to get $Pk_{DH(M1)\_i'}$ as shown in figure 4-14.

4.6.3.2 Detection and solution process

In the proposed model, this attack can be detected by S and even D separately (same treatment for S and D). Signing messages with RSU play an important role to detect different attacks. In this scenario, D (similarly to S) can detect the presence of the attacker during the communication when it verifies the correctness of information included in the identity reply IdtRp (algorithm 13). When M2 (or M3) intercepts the identity request (IdtRq) from D, it replaces the IP address and index of Pk_{DH} key according to M1, then sends it to RSU that responds D with a signed IdtRp containing $Pk_{DH(M1)\_i'}$.

Algorithm 13 Detection and solution algorithm for attack 3 during the Lightweight Diffie-Hellman Exchange Phase

(For D)
(11) D: if $t - t_{Sig(rsu)} < T$ ***then***
$Hsh_V \leftarrow h\ (@ip_{(D)}, @ip_{(S)}, Pk_{DH(M1)\_i'}, i, t_{Sig(rsu)})$
$Hsh_{RSU} \leftarrow De_{PkSig(rsu)}\ (Sig_{RSU})$
if $Hsh_V = Hsh_{RSU}$ ***then***
// IdtRp is valid, no attack
Calculate $Sk_{DH(S,\ D)}$
Else => Detection of MITM attack
Change route to RSU or consult other one
End if
Else => Detection of Reply attack
End if

(For S)
(11) S: if $t - t_{Sig(rsu)} < T$ ***then***
$Hsh_V \leftarrow h\ (@ip_{(D)}, @ip_{(S)}, Pk_{DH(M1)}, j, t_{Sig(rsu)})$
$Hsh_{RSU} \leftarrow De_{PkSig(rsu)}\ (Sig_{RSU})$
if $Hsh_V = Hsh_{RSU}$ ***then***
IdtRp is valid // no attack
Calculate $SK_{DH(S,\ D)}$
Else => Detection of MITM attack
Change route to RSU (or consult other one)
End if
Else => Detection of reply attack
End if

Vehicle D can detect this attack by verifying the signature of RSU contained in IdtRp according to the IP addresses (S and D) and index sent in IdtRq. This verification can show the incoherence between the requested information and those received. In this case, the vehicle can change the route the RSU or consult another one.

4.6.4 Attack 4 during the Identity Revocation Phase

4.6.4.1 Attack process

In this attack, RevRq message can be intercepted and dropped (or altered) by a malicious node to avoid reaching the RSU (or to be ignored by the RSU), and thereby, vehicle's identity will not be revoked as shows figure 4-15.

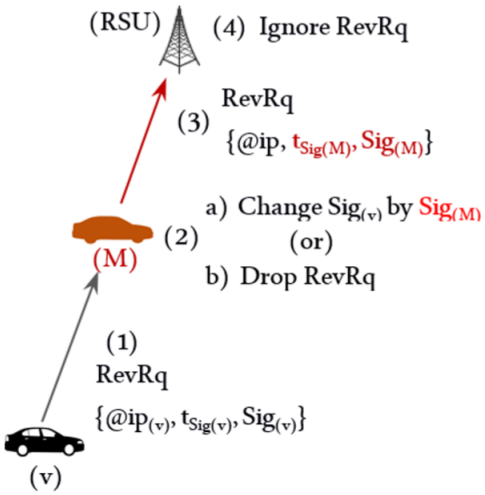


Figure 4-15: Attack during the IVP phase

In figure 4-15, M can launch the attack in two ways: intercepts RevRq message and drops it or make it invalid to be ignored by the RSU. Algorithm 14 describes the execution of the attack.

Algorithm 14 Algorithm of attack 4 during the Identity Revocation Phase

(M alters RevRq)
M: Change $Sig_{(v)}$ by $Sig_{(M)}$
$Sig_M \leftarrow Sig_{(M)}$ ($@ip_{(v)}$, $Pk_{DH(M)}$, $Pk_{Sig(M)}$, $t_{Sig(M)}$)
$M \rightarrow RSU$: RegAck {$@ip_{(v)}$, $t_{Sig(M)}$, Sig_M}
*RSU: **if** $t - t_{Sig(M)} < \Delta T$ **then***
$Hsh_{RSU} \leftarrow h$ ($@ip_{(v)}$, $Pk_{DH(v)}$, $Pk_{Sig(v)}$, $t_{Sig(M)}$)
$Hsh_V \leftarrow de_{PkSig(v)}$ (Sig_M)
if** $Hsh_{RSU} = Hsh_V$ **then
// Revoke V's identity
Else
Ignore RevRq
// V's identity is not revoked
End if
Else
Ignore RevRq
End if

(M drops RevRq)
M: Drop RevRq
RSU: // no RevRq is received, V's identity is not revoked

4.6.4.2 Detection and solution process

Either the attacker drops or it alters RevRq (algorithm 15), no reaction from the RSU, and v's identity will not be revoked. Treating this problem, v waits for a response (RevAck) from the RSU during a period of time. If no response after that, v can resend RevRq to the RSU for a number of times. If no RevAck received, v changes the route to the same RSU (or consults another one) and repeat the IVP phase.

Algorithm 15 Detection and solution algorithm for attack 4 during the Identity Revocation Phase

*(7) V: **if** no RevAck from RSU (after a waiting time) **then***
*Repeat **IVP** for a number of times*
if** no valid response **then
*Repeat **IVP** (change route to the RSU or consult another one)*
End if
End if

4.7 Conclusion

Generally, security management in any communication network influences some QoS parameters like end-to-end delay, overhead and packet delivery ratio. Specifically, in VANet, any proposed security mechanism has to treat

these parameters carefully and responds to different critical characteristics of this type of networks.

In this work, a novel approach is proposed to ensure the security of communication in VANet by implementing the transport encryption layer used in I2P. In this approach, we assumed that the encryption process is implemented between the source and destination nodes contrary to I2P where transport encryption is implemented between each two successive tunnel nodes in a tunnel.

The proposed model in this approach is designed to meet requirements of VANet, in which we have reduced the number of messages and even number of fields within messages as well as possible. Thus, it is assumed that great parts of data needed for Diffie-Hellman exchange are transferred using secured channels between RSUs using a wired connection.

As a continuity for this contribution, we aim to secure and anonymize the communication by creating tunnels and implementing encryption algorithms inspired by I2P. Each node has secure tunnels and uses them to communicate with any other node in the network. Messages can be exchanged through two tunnels created by the two end communication nodes.

Chapter 5

CONTRIBUTION 2: A NEW PROTOCOL TO ANONYMIZE COMMUNICATION IN VANET

5.1 Introduction

Besides enhancing the quality of services and developing new applications in VANet, a large area of research focuses on the security domain. The goal is to find security solutions and mechanisms to fight against different attacks. Anonymizing communication shows another aspect of security, which is based on hiding the real identities of the destination nodes, in which the adversary cannot detect their real identities. Generally, multiple searches in anonymity are based on using a third party (trust authority for example) that generates pseudonyms for entities and ensure the relation between real identities and pseudonyms.

Contributing to this issue, we continue the adaptation of the Invisible Internet Project (I2P) [20] in VANet. The I2P protocol is designed to be used in the internet network to create a subnetwork of connecting I2P routers. In the I2P network, the communication is secured and anonymous, in which the exchanged messages are encrypted and the sender and receiver node identities cannot be detected by adversaries. I2P is efficient in terms of security and anonymity due to its security mechanisms used during the communication. It sets up encryption algorithms, signatures and certificates ensuring the integrity, confidentiality, authenticity and anonymity of the exchanged messages.

Functionalities and mechanisms used by I2P are heavy to be applied directly to VANet. Amendments are needed in different algorithms and methods used by I2P to respond to different critical characteristics of VANet. After that, it can be applied easily and having efficient results in the network in terms of quality of service on one side, and to fight against some attacks on the other side.

As a continuity for the previous contribution “A secure communication model using lightweight Diffie-Hellman method in vehicular ad hoc networks”, we develop a tunnel creation algorithm and implement the garlic and tunnel encryption layers during the communication in the second contribution. To simplify the tunnel creation process, it is supposed that tunnels are created statically without maintenance (part 1 of the contribution). Then the mobility of vehicles is treated in part 2 where we develop a tunnel maintenance algorithm allowing to use the same mechanism in real vehicular networks.

Our contribution is based on two encryption levels: garlic and tunnel encryption. As an initiation to apply I2P in VANet, the operation mode of these algorithms is simplified where for each encryption level, an asymmetric algorithm is used. We show the effectiveness and the security of the proposed model by analysing different cases of anonymity and showing performance results. The simulation of different scenarios is launched using the NS3 platform.

In this chapter, we introduce the second contribution about the creation of tunnels and encryption of messages in section 5.2. Then, we discuss several related works in section 5.3. Section 5.4 is a presentation of part 1 (of the second contribution) "Anonymizing communication in VANet by applying I2P mechanisms" [45], followed by the second part "A new proposed protocol to anonymize communication in VANet" in section 5.5. Finally, we end this chapter with a conclusion.

5.2 Creating tunnels and encrypting messages

Anonymizing communication becomes a substantial issue to enhance security and face attacks. Huge researches are made in this field aiming to ensure secure and anonymous communication. In vehicular ad hoc networks, this concept is used in different application areas like military domains, where hiding destinations identities is necessary to avoid consequences attacks.

The objective in this chapter is to propose a model of security to ensure anonymity in VANet. This model is inspired from the I2P protocol, we continue the previous work by adapting some of the I2P mechanisms and algorithms to several requirements of VANet. This model is based on creating tunnels and maintaining their existence. Thus, we implement the tunnel and garlic encryption layers in addition to the transport encryption layer implemented in the previous contribution.

The proposed model is supposed to be implemented in urban environments, where speeds of vehicles are less than 50 km/h and large number of RSUs and vehicles can exist, which help the creation and maintenance of encrypted tunnels. We divide this work into two contributions; creating tunnels statically in the network without maintenance and implementing

the tunnel and garlic encryption layers in the first contribution. For the second one, we develop a tunnel maintenance algorithm to deal with the mobility of vehicles and maintain the existence of tunnels in real VANet.

The section below presents solutions and approaches providing security and anonymity of communication in VANet.

5.3 Related work

In [41], Authors propose a novel security protocol to ensure anonymous authentication. The architecture defined is based on three levels for pseudonym-based anonymous authentication. It uses four VANet elements, namely: Road Side Units (RSUs), vehicles, Certification Authority (CA) and a new element Pseudonym Server (PSS) that is responsible for assigning pseudo identities to vehicles. PSSs are distributed and have all other PSSs certificates, which makes handover easy.

Similarly, in [48], an anonymous authentication scheme is proposed by using the Proxy Mobile IPv6 model (PMIPv6) in VANet. According to PIMPv6 system, a trusted STR entity (System-level Trust Route) generates public and private parameters for other entities, and certificate legal vehicles through the identity-based group signature mechanism. Managing mobility and generating variable pseudonyms for vehicles is attributed to Local Mobility Anchor entities. RSUs are used as Mobility Access Gate entities which are responsible for mutual authentication. The scheme here and in [48] uses more entities, which cause the generation of additional messages. Registration and authentication requests from each node (especially in VANet) can raise the overhead and reduce the packet delivery ratio in the network.

Authors in [83] propose a security protocol to ensure an anonymous handover authentication designed for vehicular LTE-A (Long Term Evolution Advanced) networks. The roaming between service and target nodes is secured using elliptic curve cryptography. The user authentication is processed by the use of a smart card. Handover between two RSUs cannot be executed correctly till contracting the previous RSU, which is not guaranteed in some cases according to VANet characteristics.

[40] Proposes an anonymous authentication with a conditional privacy-preserving scheme for VANet. As several solutions, the trust authority (TA) is responsible for delivering the system parameters besides generating the original and dummy user identities for vehicles and RSUs. The node uses its authorization key to generate its anonymous certificate for each temporary short time. In this model, TA can trace the real identity of a node based on its anonymous certificate and consequently revoke the privacy of the malicious nodes to avoid further damage. The approach here and in [40] is based on using smart cards that involve the assistance of the user, which is not convenient in VANet.

The authors in [59] present a secure and anonymous conditional privacy-preserving authentication scheme providing security and privacy in VANet. Before sending a message, a signature process is applied to guarantee its integrity and authenticity. This process can treat a batch of messages together at one time instead of treating them one by one.

Most of the works in this domain use generally three or more entities: vehicles, RSUs and trust authority. The general idea is to use pseudonyms to hide the real identity of the node. The generation of pseudonym needs a frequent exchange with the dedicated entity (usually the trust authority). This exchange can generate an additional delay and allow several attacks to be launched.

The section below details the first part of the second contribution.

5.4 Part 1: Anonymizing communication in VANet by applying I2P mechanisms

5.4.1 Model overview

Anonymising communication is a way to face multiple attack threats. Applications in many domains use this concept to guarantee the privacy of users and secure communication. Military apps, for example, can use that to hide war vehicles identities to communicate anonymously without being tracked by the enemies.

In this work, we put a secure protocol that ensures anonymity. We use a mechanism based on tunnels and an encryption model inspired by I2P protocol in the way to hide real destinations identities. This work is a continuation of the previous contribution [46] to apply I2P in VANet.

This mechanism is set up in layer 3 using Ad hoc On-demand Distance Vector (AODV) protocol [64]. A sublayer responsible for the tunnel creation process is added by creating messages and tables to ensure the good functioning of the process. The section below presents an overview of the AODV protocol.

5.4.1.1 AODV (Ad hoc On-demand Distance Vector)

AODV [64] is a reactive vector-based protocol capable of both unicast and multicast routing. In the network, each node has the information on its direct neighbourhood that represents the first one-hop relays to reach the remote nodes (more than one jump). The path calculation is based on the distance measured by the number of intermediate relay between the source node and the destination node.

In this protocol, establishing routes between nodes is performed only when needed (just before sending data packets), which reduces the overhead in the network (especially in scenarios with less of vehicles). AODV

only maintains routes currently used for a certain time interval [86].

In AODV, before sending data to a destination, the source node checks in its routing table if it has a valid route to that destination, otherwise it starts the route discovery process by broadcasting a route request (RREQ) (figure 5-1). The RREQ packet contains the IP address, the source and destination node sequence numbers [72]. Sequence numbers are used to maintain the consistency of routing information that has a higher sequence number indicating that a newer route exists.

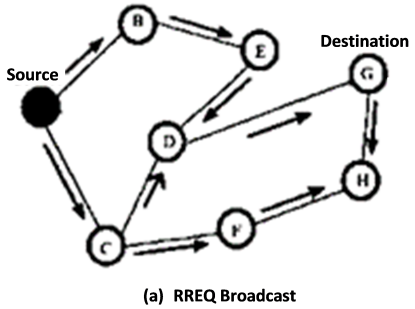


Figure 5-1: Route discovery with RREQ in the AODV protocol

Each node receives the RREQ packet updates its routing table by adding a route to the source node [74], this method of recording the previous hop is known as backward learning, then checks if it is the destination or if it has a route to the destination with a sequence number greater than or equal to that indicated in the RREQ packet. If not, it rebroadcasts the RREQ packet [72]. Otherwise, it responds with a Route Reply packet (RREP) in unicast to the source node. The RREP must be sent a sequence number greater than or equal to that of RREQ. In this case, each node has received the RREP, it updates its routing table by adding an entry to the destination mentioned in this packet (figure 5-2) [38].

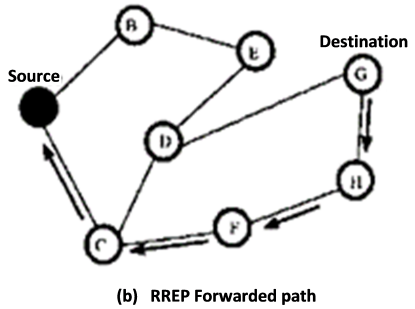


Figure 5-2: Response of the destination node by RREP

Each entry in the routing table corresponds to a destination, it consists of a set of fields including essentially the identifiers of the recipient and of

the next node used as a relay to reach the recipient, the sequence number and the expiration time of the entry.

Upon reception of RREP, the source node sends the data packets to the next nodes according to the route discovered, in which each node determines the next hop from its routing table. If the source node does not receive a response after a timeout, it resets the route discovery process and expects a time longer than the first interval. If no response is received after three attempts, the source node abandons the route discovery process which will not be restarted until after a certain delay.

A route is considered active when it is used to transmit data packets periodically. When the source node ends sending data packets and the timer for that route expires, the route entry is cleared from the intermediate node routing table. If a link in an active route breaks, the node that detects the break sends a Route Error packet (RERR) to the source. In this case, this node may restart the route discovery process again if it is needed [38].

The section below presents the process of the proposed mechanism. The notations used in this part and the next part are decrypted in table 5.4.1.1.

TREQ	Tunnel Request message
TREP	Tunnel Reply message
TCRT	Tunnel Creation message
TACK	Tunnel Acknowledgment message
TunnelT	Tunnel Table
TID	Tunnel Identifier
M	Clear message
Mge	Garlic encrypted message
$e_{\_Pk}$	Asymmetric encryption
$de_{\_Prk}$	Asymmetric decryption
$eg_{\_Pk}$	Garlic encryption
$dg_{\_Prk}$	Garlic decryption
$et_{\_Pk}$	Tunnel encryption
$dt_{\_Prk}$	Tunnel decryption
Hp	Hop count
T1, T2, T3	Periods of time
THELLO	Tunnel HELLO
$TCRT\_U$	Tunnel Creation Update message
$TACK\_U$	Tunnel Acknowledgment Update message

Table 5.1: Notation description 2

5.4.2 The operating principle of the proposed protocol

This section describes the proposed protocol as follows: the definition of vehicles to RSUs, the tunnel creation process and the communication.

5.4.2.1 Definition of vehicles to RSUs

Before starting the communication, each node registers its identity (called nodeInfo) into a network database (NetDB) located in RSUs. NetDBs are distributed and hosted in some specific infrastructures (RSUs). The identity registration is presented in the previous work [46]. In this contribution, an identity includes in addition, a public encryption key used in the garlic and tunnel encryption algorithms (section 5.4.2.3).

As detailed in the first contribution, the identity registration phase is divided into two sub-phases; Identity registration request sub-phase (IRRP) and identity registration checking sub-phase (IRCP):

a. Identity registration request sub-phase

In this sub-phase, the vehicle registers its identity in the NetDB temporary by using the three messages (figure 5-3); registration request (RegRq), registration reply (RegRp) and registration acknowledgement (RegAck).

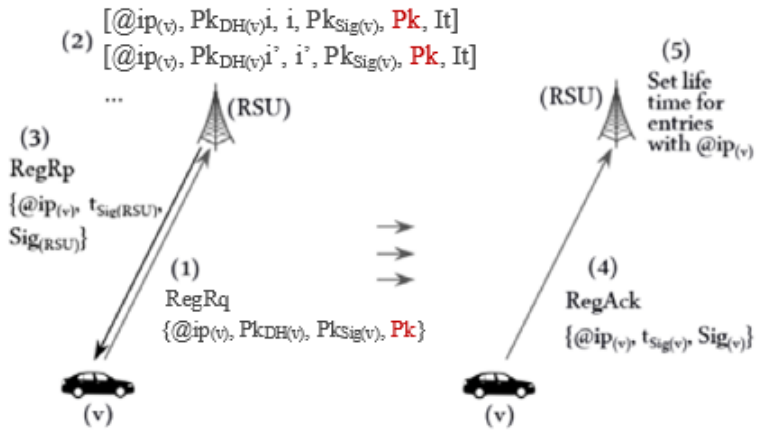


Figure 5-3: Identity registration request sub-phase

This sub-phase starts by sending its identity (which includes the public encryption key as mentioned before) to a near NetDB (RSU) using the RegRq message. At the reception, when the NetDB receives the nodeInfo, it checks if the received information exists before records it temporary. As a security issue, the RSU signs the received information with its private key and resends it to the vehicle in a RegRp message. In this case, the NetDB (RSU) cannot publish the registered temporary identities either to other NetDBs or to the vehicles requesting for these identities. When the vehicle receives the RegRp message, it resends a RegAck, a signed

message (acknowledgement) sent to the NetDB to confirm the reception of the RegRp. In this case, the registration is confirmed and the identity can be registered for a long time in the database.

Moreover, RSU starts distributing the registered identity with other RSUs (NetDBs) in the network. After a period of time, nodeInfos of vehicles entered in the network have been registered in all NetDBs in the network, and each node can get any other node's identity by requesting any NetDB database. This process is detailed in chapter 3 section 4.5.2.1.

b. Identity registration checking sub-phase

After a period of time, the vehicle checks the registration of its identity in the NetDB or to another one by launching the second sub-phase IRCP. It is based on two messages (figure 5-4); Checking Request (ChRq) and Checking Reply (ChRp). In this sub-phase, the vehicle sends the signed ChRq message to a NetDB to check if its identity exists. The NetDB sends the ChRp message including the response to the vehicle. The IRCP sub-phase is present in detail in chapter 3 section 4.5.2.1.

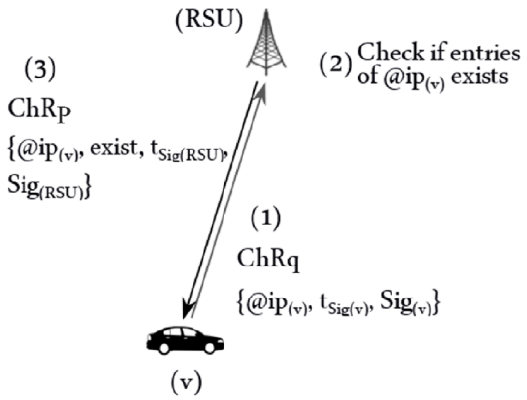


Figure 5-4: Identity registration checking sub-phase

After the identity registration phase, each node has to create its inbound and outbound tunnels to start the communication. In the proposed model, two-hop tunnels are used. According to recent studies in I2P [30], a two-hop and three-hop tunnels are usually preferred in terms of security. Tunnels longer than three-hop do not offer additional protection. The section below presents the second phase in the proposed protocol.

5.4.2.2 Tunnel creation process

The tunnel creation takes place in two steps: the first one for getting the tunnel nodes' identities, the second to launch the tunnel creation process.

a. Getting tunnel nodes' identities

The first tunnel node (TN1) is responsible for the selection of the second tunnel node (TN2) and similarly, TN2 is responsible for the third tunnel

node (TN3). Tunnel nodes are not necessarily neighbours (figure 5-5).

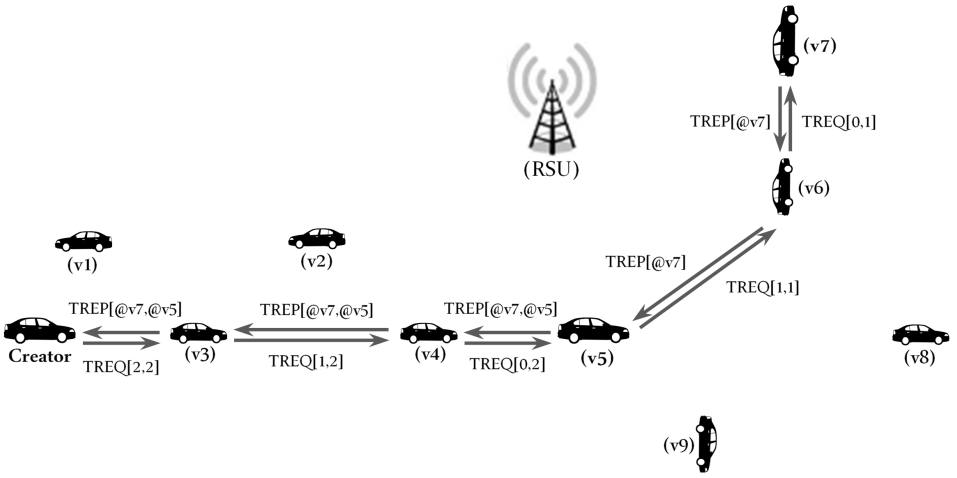
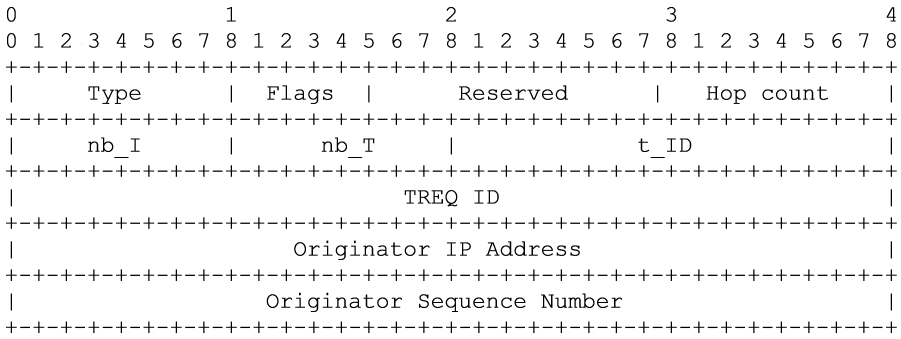


Figure 5-5: Getting tunnel nodes' identities

TN1 selects the tunnel nodes without knowing their IP addresses, but according to two random numbers i and j. It broadcasts a tunnel request message (TREQ [i, j]) knowing that j is the number of tunnel nodes and i is the number of intermediate nodes between them. TREQ is treated by nodes in the same way as RREQ (Route Request) message used in AODV protocol. Long tunnels under high mobility will be more difficult to maintain. For that, it is required to use short tunnels; two-hop tunnels (j=3) and i is selected randomly in the interval [1, 2] to avoid a high number of intermediate nodes. TREQ message includes a set of fields (figure 5-6).



Type: Type of the AODV message
Flags: |J|R|G|D|U| same flags of RREQ message in AODV
Reserved: Not used
Hop Count: Hop count between tunnel nodes
nb_I: Number of intermediate nodes between tunnel nodes
nb_T: Number of tunnel nodes in the tunnel
t_ID: Temporary identifier generated by S and TN2 to identify their addresses
TREQ ID: Identifier of the TREQ
Originator IP Address: IP address of the source of TREQ
Originator Sequence Number: Sequence Number of the source of TREQ

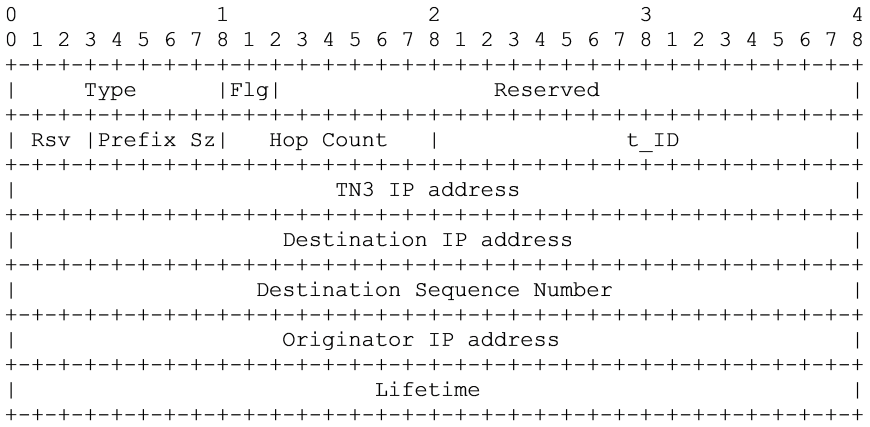
Figure 5-6: Tunnel Request (TREQ) message format

When TN2 is selected by the TREQ message, it does the same process by broadcasting a new TREQ to select TN3. At reception by TN3, it replies TN2 by an encrypted tunnel reply including TN3 IP address (TREP[@TN3]) using TN2's public key as detailed in algorithm 16. After receiving and decrypting this message, TN2 sends an encrypted TREP [@TN2, @TN3] to TN1 using TN1's public key.

Algorithm 16 Algorithm of getting tunnel nodes' identities

```
j = 3; INodes_max = 2; // Max number of intermediate nodes
                          between tunnel nodes
TN1: i ← random (0, INodes_max);
TN1: j ← j-1;
TN1: Broadcasts TREQ [i, j];
For each node N receives TREQ [i, j]:
N: if (i > 0) then
     i ← i -1;
     Rebroadcasts TREQ [i, j];
   else
     j ← j-1;  // j=1 so N is TN2;
     TN2: i ← random (0, INodes_max);
     TN2: broadcasts TREQ [i, j] to select TN3;
   end if
Each node N receives TREQ [i, j]:
N: if (i > 0) then
     i ← i-1;
     Rebroadcasts TREQ [i, j];
   else
     j ← j-1;  // j=0 so N is TN3;
     TN3: e_TREP ← e_Pk(TN2) (TREP [@TN3]); // encryption using TN2's
                                               public key
     TN3: sends e_TREP to TN2;
   end if
TN2: TREP [@TN3] ← de_Prk(TN2) (e_TREP); // decryption using TN2's
                                            privet key
TN2: e_TREP ← e_Pk(TN1) (TREP [@TN2, @TN3]);
TN2: sends e_TREP to TN1;
```

Figure 5-7 describes in detail the TREP message format. Our goal behind this step is to allow each creator to get IP addresses of the possible future tunnel nodes.



`Type`: Type of the AODV message
`Flg`: |R|A| same flags of RREP massage in AODV
`Reserved, Rsv`: Not used
`Hop Count`: Hop count between tunnel nodes
`Prefix Sz`: Prefix size used in RREP in AODV
`t_ID`: The temporary identifier sent by TN2 (or S)
`TN3 IP address`: IP address of TN3
`Destination IP Address`: IP address of the sender of TREP
`Destination Sequence Number`: Sequence Number of the sender of TREP
`Originator IP Address`: IP address of the sender of the received TREQ
`Lifetime`: Lifetime of TREP (in milliseconds)

Figure 5-7: Tunnel Reply (TREP) message format

b. Tunnel creation

After getting tunnel node identities, TN1 generates TID (Tunnel Identifier) for the tunnel [TN1, TN2 and TN3] and selects its type T_type, then it creates a new entry in its tunnel table (TunnelT) about this tunnel. TID is a unique identifier for each tunnel in the network.

At this stage, TN1 sends an encrypted tunnel creation message (TCRT) to TN2 using an asymmetric algorithm (figure 5-8).

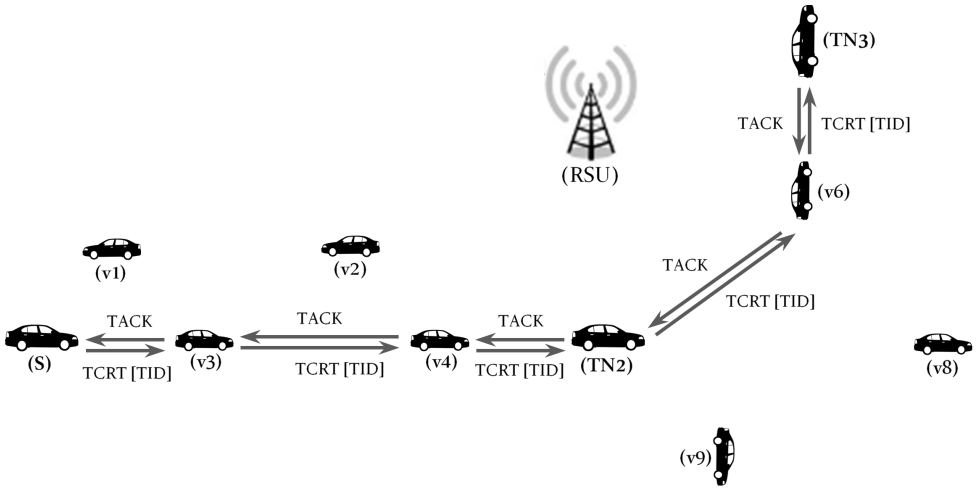
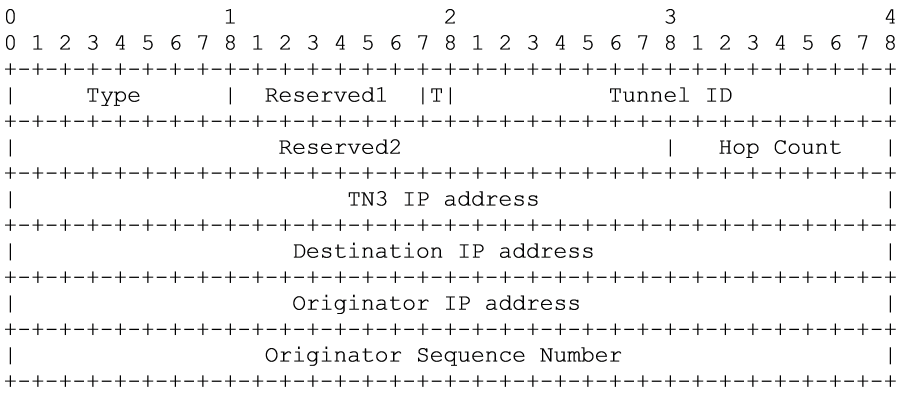


Figure 5-8: The tunnel creation process

TCRT (5-9) plays the role of a request to create the tunnel, according to T_type and TID sent.

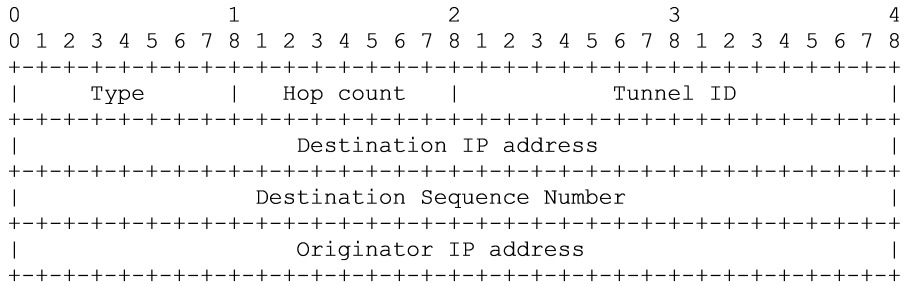


Type: Type of the AODV message
Reserved1, Reserved2: Not used
T: Tunnel type (outbound or inbound)
Tunnel ID: Identifier of the tunnel in the network
Hop Count: Hop count between tunnel nodes
TN3 IP address: IP address of TN3
Destination IP Address: IP address of the destination of TCRT
Originator IP Address: IP address of the source of TCRT
Originator Sequence Number: Sequence Number of the source of TCRT

Figure 5-9: Tunnel Creation (TCRT) message format

When receiving TCRT [TID, T_type, TN3], TN2 decrypts the message and creates a new entry in its TunnelT table containing TID, NxHp (next tunnel node in the tunnel) and T_type. Then TN2 sends an encrypted TCRT to TN3 using TN3's public key.

Similarly to TN2, when TN3 receives TCRT, it decrypts that message and creates a new entry in its TunnelT. Then, TN3 responds TN2 with an encrypted tunnel acknowledgement message (TACK [TID]). The format of TACK message is illustrated in figure 5-10.



Type: Type of the AODV message
Hop Count: Hop count between tunnel nodes
Tunnel ID: Identifier of the tunnel in the network
Destination IP Address: IP address of the sender of TACK
Destination Sequence Number: Sequence Number of the sender of TACK
Originator IP Address: IP address of the sender of the received TCRT

Figure 5-10: Tunnel Acknowledgement (TACK) message format

TACK message is encrypted by using TN2's public key as detailed in algorithm 17. TN2 decrypts TACK then encrypts it using TN1's public key and forwards the message to TN1 to confirm the reception of TCRT and consequently the creation of the tunnel. In this case, TN1 sends a LeaseSet to one of NetDB databases. A LeaseSet contains the identities of TN3s belonging to the outbound tunnels of TN1 (as represented in I2P chapter 3, section 3.7). After receiving LeaseSets, the NetDB database shares the information with other NetDBs in the network.

Algorithm 17 Algorithm of the tunnel creation process

```
TN1: TID ← generateTID (TN1, TN2, TN3);
     if outbound tunnels > inbound tunnels then
        T_type ← inbound;
     else
        T_type ← outbound;
     end if
     TunnelT.addTunnel (TID);
     e_TCRT ← e_Pk(TN2) (TCRT [TID, T_type, TN3]);
TN1 → TN2: {e_TCRT};
TN2: TCRT [TID, T_type, TN3] ← de_Prk(TN2) (e_TCRT);
     if T_type = outbound then
        NxHp ← TN3;
     else
        NxHp ← TN1;
     end if
     TunnelT.addTunnel (TID);
     e_TCRT ← e_Pk(TN3) (TCRT [TID, T_type]);
TN2 → TN3: {e_TCRT};
TN3: TCRT [TID, T_type] ← de_Prk(TN3) (e_TCRT);
     if T_type = outbound then
        NxHp ← NULL;
     else
        NhHp ← TN2;
     end if
     TunnelT.addTunnel (TID);
     e_TACK ← e_Pk(TN2) (TACK [TID]);
TN3 → TN2: {e_TACK};
TN2: TACK [TID] ← de_Prk(TN2) (e_TACK);
     e_TACK ← e_Pk(TN1) (TACK [TID]);
TN2 → TN1: {e_TACK};
TN1: TACK [TID] ← de_Prk(TN1) (e_TACK);
     if TunnelT.T_type (TID) = inbound then
        Send TN3 to NetDb;
     end if
NetDB: save and share this information with other NetDbs.
```

5.4.2.3 Communication using tunnels

After creating tunnels, each node can start communication by using one of its outbound tunnels to send messages and then received by the destination through one oh its inbound tunnels (figure 5-11). To be received by the inbound tunnel of a destination (dst), the source node (src) should know TN3_dst that was registered in the NetDB database by the destination after creating its inbound tunnels. Src and dst nodes play the role of TN1_src (source) and TN1_dst (destination) respectively.

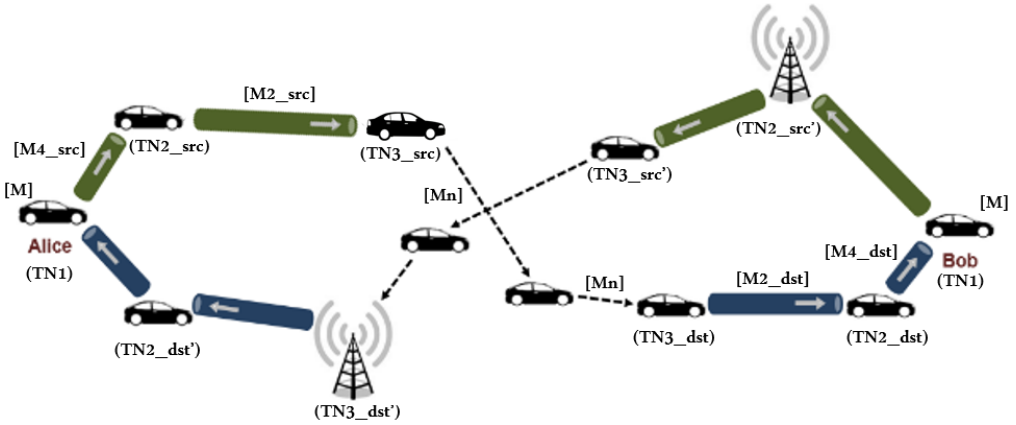


Figure 5-11: Communication phase

At this stage, TN1_src encrypts the message M using an asymmetric encryption algorithm to obtain Mge (a garlic encrypted message) as detailed in algorithm 18. Then it iteratively encrypts (asymmetrically) the message according to its outbound tunnel nodes (tunnel encryption). For each one of them, TN1_src adds TID_src (Outbound tunnel ID of Source) as a header to the message and encrypts the whole message. Except for TN3_src, the header contains TID_dst (Inbound tunnel ID of destination) and @TN3_dst.

At the reception, each outbound tunnel node decrypts the message to extract TID_src. According to that, the node selects the next hop using its TunnelT. TN3_src decrypts the message to obtain an encrypted message Mge, @TN3_dst and TID_dst, then it sends a message Mn to TN3_dst including Mge and TID_dst as a header. Mn is routed in the network till reaching TN3_dst.

At the reception, each inbound tunnel node reads TID_dst and selects the next hop using its TunnelT table, then encrypts the message (the encryption is done by decrypting the unencrypted message) and adds TID_dst as a header to send it. Except for TN1_dst, it iteratively decrypts the message according to its inbound tunnel nodes identified by TID_dst to get Mge (the decryption is done by encrypting the message). TN1_dst decrypts Mge using an asymmetric encryption algorithm to obtain the message M.

If an acknowledgement is needed, TN1_dst resends Ack message by following the same process as before knowing that TN1_src and TN1_dst will be the destination and the source nodes respectively.

Algorithm 18 Communication algorithm

TN1_src: $Mge \leftarrow eg_{Pk(TN1\_dst)}(M)$; // Garlic encryption
$M1\_src \leftarrow [@TN3\_dst+TID\_dst] + Mge$;
$M2\_src \leftarrow et_{Pk(TN3\_src)}(M1\_src)$; // Tunnel encryption
$M3\_src \leftarrow [TID\_src] + M2\_src$;
$M4\_src \leftarrow et_{Pk(TN2\_src)}(M3\_src)$;
TN1_src → *TN2_src:* {*M4_src*};
TN2_src: $M3\_src \leftarrow dt_{Prk(TN2\_src)}(M4\_src)$; // Tunnel decryption
$TN3\_src \leftarrow TunnelT.getNxtHp(TID\_src)$;
TN2_src → *TN3_src:* {*M2_src*};
TN3_src: $M1\_src \leftarrow dt_{Prk(TN3\_src)}(M2\_src)$
$Mn \leftarrow [TID\_dst] + Mge$;
TN3_src → *TN3_dst:* {*Mn*};
TN3_dst: $TN2\_dst \leftarrow TunnelT.getNxtHp(TID\_dst)$;
$M1\_dst \leftarrow dt_{Prk(TN3\_dst)}(Mge)$;
$M2\_dst \leftarrow [TID\_dst] + M1\_dst$;
TN3_dst → *TN2_dst:* {*M2_dst*};
TN2_dst: $TN1\_dst \leftarrow TunnelT.getNxtHp(TID\_dst)$;
$M3\_dst \leftarrow dt_{Prk(TN2\_dst)}(M1\_dst)$
$M4\_dst \leftarrow [TId\_dst] + M3\_dst$;
TN2_dst → *TN1_dst:* {*M4_dst*};
TN1_dst: $M1\_dst \leftarrow et_{Pk(TN2\_dst)}(M3\_dst)$;
$Mge \leftarrow et_{Pk(TN3\_dst)}(M1\_dst)$;
$M \leftarrow dg_{Prk(TN1\_dst)}(Mge)$; // garlic decryption

The section below presents the analysis of security and performance of the proposed protocol during the communication.

5.4.3 Performance and security analysis

To show the security of the proposed protocol, we analyze the anonymity issue in different stages during its execution. We show the effectiveness and lightness of this protocol compared to AODV by analyzing some performance results.

5.4.3.1 Security analysis

The proposed protocol uses mechanisms based on tunnels and encryption algorithms, in which identities presented in the exchanged messages have no relation to the communicant nodes. At the reception, every tunnel node changes these identities according to the next-hop defined in its tunnel table. The attack treated in this work is to disclose the identities of the

source and destination nodes. After that, the attacker will be able to launch malicious activities in the network according to its purpose.

To succeed this attack in the proposed protocol, the only thing the attacker can do is to know different created tunnels (with their TIDs) and then during the communication, it must retrieve the used TIDs to get the tunnel nodes and then it can know the destinations' identities. To do so, it must be present during the tunnel creation step, in which it can get TIDs and know different created tunnels. After that, it must eavesdrop the communication to get the used TIDs to know the tunnel nodes, and then know the source and destination nodes.

Facing this attack during the tunnel creation, the protocol puts an encryption mechanism, in which the tunnel information (TIDs and node identities) is hidden. It uses encrypted messages TREP, TCRT and TACK between each two tunnel nodes except TREQ that includes i and j indexes. In wireless environments, these indexes will not be significant to know the tunnel nodes. At this stage, it is ensured that the attacker does not have any information about the created tunnels.

During the communication, data messages are encrypted inside and between tunnels. For that, the attacker cannot detect which tunnels are used even if it retrieves TIDs. Between tunnels, destination inbound tunnel TID is not encrypted, but it is not significant since the attacker could not know the different created tunnels during the tunnel creation step.

5.4.3.2 Performance analysis

In this work, the performance results are shown by simulating AODV and the proposed protocol in NS3 platform [21]. Table 5.4.3.2 summarizes the settings and parameters used during the simulation.

Area size	1280*480m
Number of nodes	10 to 20
Number of communications	3
Number of messages per second	1
Simulation time	100s
Packet size	256
Routing protocol	AODV
Propagation model	Friis
Transmission power	0dbm
Frequency band	5GHz
Transmission bandwidth	5MHz

Table 5.2: Simulation settings 1

We compare the proposed protocol with the original version of AODV in different scenarios to show the effectiveness and lightness of the proposed

algorithms regarding to the network performances. This section shows the simulation results based on the packet delivery ratio (PDR), end-to-end delay and overhead. These parameters are defined in chapter 2, section 2.2.2.

We show the results of our simulations by using box plots. NS3 simulator uses random functions that influence its operational mode, in which causes the generation of results with different values even for the same scenario with the same settings. For that, we launch the simulation six times for each scenario to get a set of different values. After that, we show the results for the different sets by using box plots.

Figure 5-12 shows that the packet delivery ratio in the proposed protocol is less than in AODV, in which in our protocol, most of the values in different scenarios vary between 30% and 80% while in AODV, they vary between 60% and 90%. The PDR is lower in our protocol due to the security mechanisms and processes that are developed in the protocol.

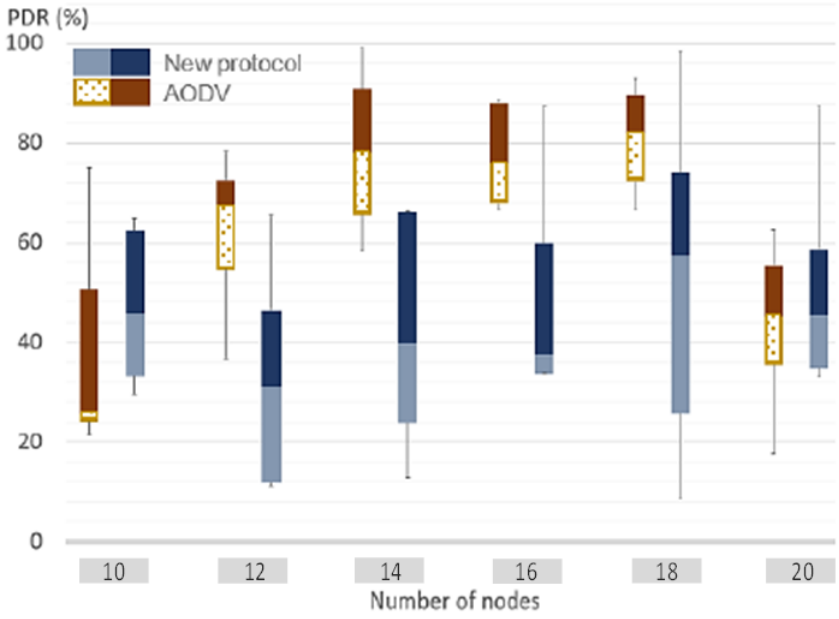


Figure 5-12: Packet delivery ratio of the proposed protocol version 1 and AODV

Figure 5-13 shows that the proposed protocol is better than AODV in term of end-to-end delay and its stability in different scenarios from the routing point of view. Delay values in AODV typically vary between 0.003s and 0.2s while in the proposed protocol, they are between 0.007 and 0.03s.

The on-demand process to discover routes in AODV takes time during the communication. Before sending the data messages, the node launches this process if it has not the route to the destination. It broadcasts RREQ messages and waits for the RREP from the destination, which delays the communication. In the proposed protocol, each node launches the tunnel

creation process once it enters in the network. This process uses TREQ, TREP, TCRT and TACK messages, which help to discover routes between the different nodes inside the created tunnels during this process. After that, each node has its own tunnels, where routes are already created. In this case, data messages can be sent directly without launching the on-demand discovery process, which reduces the delay and speeds up the communication.

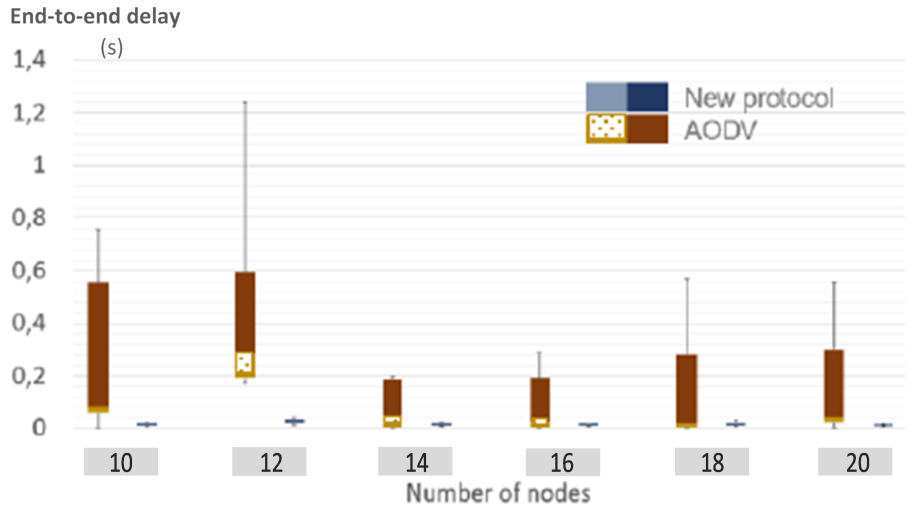


Figure 5-13: End-to-end delay of the proposed protocol version 1 and AODV

As represented in figure 5-14, the protocol generates an additional overhead comparing to AODV, in which most of the values for our protocol vary between in 100 (*100%) and 450 (*100%) contrary to AODV that generates between 25 and 50 overhead. This can be justified by the implementation of security mechanisms added to ensure anonymity.

This overhead is generated with high level just first time when starting the tunnel creation process. After that, it can be reduced considerably when there is no frequent change of routes.

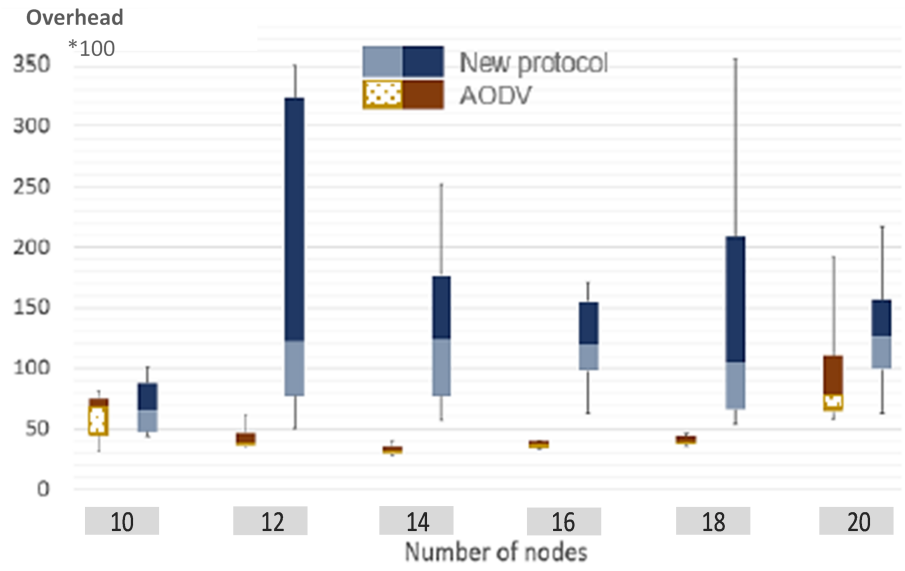


Figure 5-14: Overhead of the proposed protocol version 1 and AODV

Based on the simulations of this version of the proposed protocol and AODV in a static network, we notice that the proposed protocol has a low packet delivery ratio compared to AODV. This difference refers to the security algorithms implemented in this protocol. However, it has a better end-to-end delay than AODV thanks to the additional messages of security that provide information about routes in the network, which help to avoid the on-demand process to discover new routes and save time.

The results show that the main problem that our protocol suffers from is the overhead. Besides using AODV messages for routing, the proposed protocol uses the additional messages of security, which involves a high overhead compared to AODV.

The following section presents in detail the second part of the second contribution.

5.5 Part 2: A new proposed protocol based on I2P to anonymize communication in VANet

5.5.1 Model overview

In the previous work, a secure model is developed to ensure the anonymity of the communication in vehicular ad hoc networks. This model is inspired by the protocol I2P, which is mainly based on using tunnels and implementing encryption and signature mechanisms.

The proposed protocol uses the same principle in VANet by creating tunnels and secure communication by using encryption and signature al-

gorithms. This protocol is developed according to two versions; the first version is represented in the previous contribution, in which the tunnels are created and used to exchange data but not maintained. Once a tunnel is created, it is supposed to exist for a long time. This version is implemented using static scenarios where no movement in the network and each vehicle has the same position during the time of the simulation.

In this contribution, we propose the second version of the protocol. We continue the previous work by developing an algorithm of maintenance that keeps the existence of tunnels and provides the tunnel nodes with the current information about the tunnel during the communication. This algorithm is based on exchanging messages between the tunnel nodes to provide them information about the status of the tunnel. At any time, if a breakage occurred or the tunnel becomes long, tunnel nodes launches the maintenance process to repair the tunnel, in other cases when the tunnel cannot be repaired, a tunnel creation process is launched to create a new tunnel.

In this work, the proposed protocol is improved by developing a tunnel maintenance algorithm, which allows using this protocol in real mobile scenarios of VANet.

The proposed tunnel maintenance process is detailed in the following section.

5.5.2 Tunnel maintenance process

Vehicular networks have special characteristics comparing to other kinds of networks. High vehicle velocities and fast change of the topology make maintenance of tunnels and even its creation very difficult in some situations. For that, the developed mechanisms used for the creation and maintenance of tunnels should consider these characteristics and face these limitations.

After the tunnel creation process, tunnel nodes have to maintain consistently their existence in the network. In case of breaking of such links between tunnel nodes, these nodes can detect this breakage and know which process of reparation can be launched. In this contribution, a tunnel is a set of tunnel nodes (3 nodes max) related to each other by normal nodes (2 intermediate nodes max) as it is supposed in the previous work.

5.5.2.1 THELLO exchange

After creating tunnels, tunnel nodes exchange beacons between each other to keep information about the state of the tunnel, which is based on the state of the link between tunnel nodes. This approach considers that the state of the link depends on the number of hops between the two tunnel nodes. Each two successive tunnel nodes exchange periodically encrypted

Tunnel HELLO (THELLO) messages between each other through intermediate nodes; TN1 to TN2, TN2 to TN1, TN2 to TN3 and TN3 to TN2 (algorithm 19).

Algorithm 19 Algorithm of the THELLO exchange

For each time interval:

TN1 → *TN2* $\{e_{Pk(TN2)}$ *(THELLO [TID, HP, Flag])};*

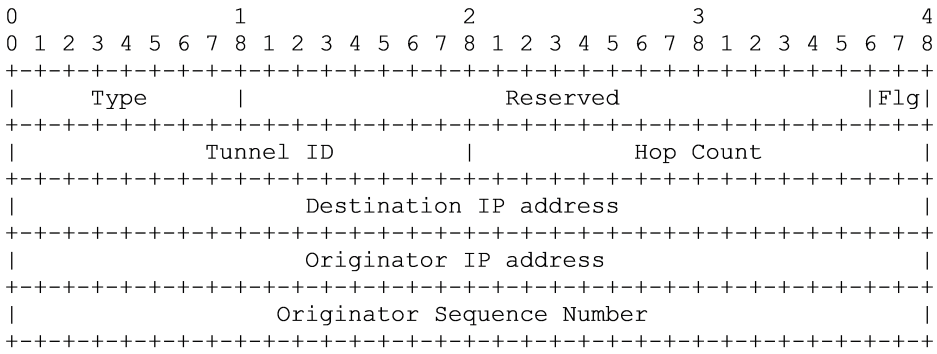
TN2 → *TN1* $\{e_{Pk(TN1)}$ *(THELLO [TID, HP, Flag])};*

TN2 → *TN3* $\{e_{Pk(TN3)}$ *(THELLO [TID, HP, Flag])};*

TN3 → *TN2* $\{e_{Pk(TN2)}$ *(THELLO [TID, HP, Flag])};*

A THELLO message is encrypted using an asymmetric encryption algorithm. The sender encrypts it using the public key of the receiver which uses its private key for the decryption. THELLO includes the tunnel ID, the hop count between tunnel nodes and other fields shown in figure 5-15.

After receiving TACK from TN3 during the tunnel creation, TN2 start exchanging encrypted THELLO messages each interval of time with TN3. The same process is launched between TN1 and TN2 after receiving TACK from TN2.



`Type`: Type of the AODV message
`Reserved`: Not used
`Flg`: Flags about distance (NORMAL, LONG, CONFIRMED_LONG)
`Tunnel ID`: Identifier of the tunnel in the network
`Hop Count`: Hop count between tunnel nodes
`Destination IP Address`: IP address of the destination of THELLO
`Originator IP Address`: IP address of the sender of THELLO
`Originator Sequence Number`: Sequence Number of the sender of THELLO

Figure 5-15: THELLO message format

THELLO message indicates the distance between tunnel nodes. It is used to inform tunnel nodes about the distance between each other. When THELLO is sent to the next tunnel node, each intermediate node increments the number of hops (Hp) and then forwards the message to the next

node until being received by the next tunnel node, which notifies the tunnel nodes when the number of intermediate nodes is increased.

When receiving a THELLO message, the tunnel node checks the number of hops to decide if the route is long or not. If Hp is under than a predefined threshold, the link is considered as normal (using the NORMAL flag in THELLO message) even if it was declared as long (LONG flag) by the sender tunnel node. If Hp is more than the threshold, the tunnel node checks the state of the link indicated by the sender:

- If it is declared as LONG, the node confirms this declaration and marks the state flag as CONFIRMED_LONG.

- If it is declared as NORMAL, it marks the state as LONG for the first time.

During communication, tunnels can be broken in many cases. In this part of the contribution, a maintenance algorithm is presented to maintain the existence of tunnels. This work is presented in the following section.

5.5.2.2 The tunnel breakage and maintenance cases

The tunnel maintenance process is launched differently from a tunnel node to another, in which each one has a special role in this process. According to the link that is broken, the maintenance process can be launched in such a manner. In the proposed algorithm, two tunnel nodes can detect the breakage between each other; TN1 and TN2. A breakage means the long-distance or the link break between tunnel nodes.

Each of the two tunnel nodes acts in the maintenance process according to its position in the tunnel. The creator maintains the existence of TN2 that maintains the existence of TN3. Two cases of breakage can occur during the communication: a breakage between TN1 and TN2, or/and breakage between TN2 and TN3. For that, two algorithms are created to maintain the existence of the tunnel; one algorithm launched by TN2 and the second launched by TN1.

a. Maintenance of TN2

After sending TACK to TN2, TN3 waits for the THELLO message from TN2. If no THELLO is received after a long interval of time (T1), TN3 considers that the link TN2-TN3 is broken and it deletes the entry of the tunnel from its TunnelT table (figure 5-16).

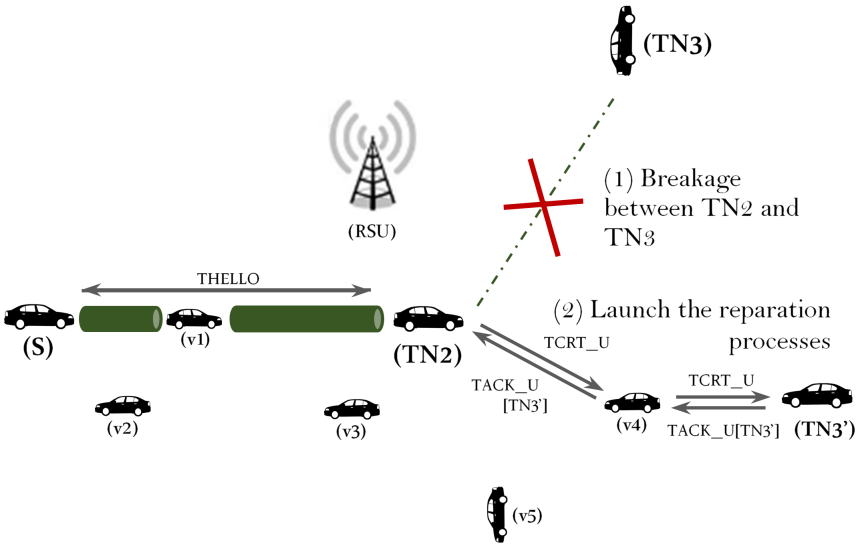


Figure 5-16: Tunnel maintenance of TN2

At the reception of THELLO, TN3 checks the state of the route declared by TN2 and the number of hops. According to these two parameters, TN3 decides as detailed in the algorithm 20.

- If the distance is accepted, TN3 resends an encrypted THELLO message to TN2 indicating the state of the route as NORMAL in the THELLO message;

- Otherwise, it checks the state declared by the sender: If it is declared as NORMAL, it indicates that the route is long (LONG flag) for the first time. If it is declared as LONG, the node confirms this state by CONFIRMED_LONG flag, then resends the encrypted THELLO message to TN2. If the sender has already confirmed that the distance is long (by CONFIRMED_LONG), TN3 stops sending encrypted THELLO to TN2, then deletes the tunnel entry from its TunnelT table.

Algorithm 20 The maintenance algorithm of TN2

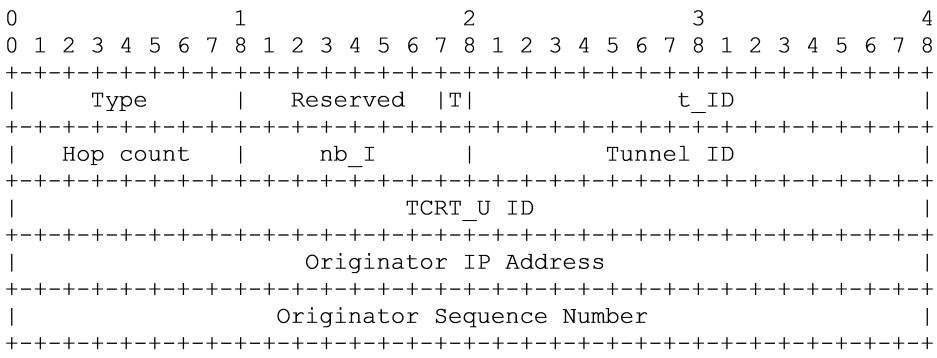
```
if the sender is TN3 then
    if FLAG = C_LONG then
      tunnelTable.FLAG ← IN_REPATION;
      TN2: delete TN3;
      TN2 => TN'3 {TCRT_U [TID', T_type, i]};
      TN'3 => TN2 {e_Pk(TN2) (TACK_U [TID", @TN'3])};
      TN2 => TN1 { e_Pk(TN1) (TACK_U [TID", @TN'3])};
    else if nb_hop <= Top then
            FLAG ← NORMAL;
            Send e_Pk(TN3) (THELLO) to TN3;
    else if FLAG = NORMAL then
            FLAG ← LONG;
            Send e_Pk(TN3) (THELLO) to TN3;
     else
            FLAG ← C_LONG;
            Send e_Pk(TN3) (THELLO) to TN3;     Delete the entry;
     end if
 end if
if the sender is S then
    if FLAG = C_LONG then
            Delete the entry;
            Send e_Pk(TN3) (THELLO) with FLAG = C_LONG to TN3;
    else if nb_hop <= Top then
            FLAG ← NORMAL;
            Send e_Pk(S) (THELLO) to S;
    else if FLAG = NORMAL then
            FLAG ← LONG;
            Send e_Pk(S) (THELLO) to S;
    else
            FLAG ← C_LONG;
            Send e_Pk(S) (THELLO) to S;    Delete the entry;
    end if
end if
```

When TN2 receives THELLO from TN3, the same process is repeated, except when the distance is long and THELLO sent with the flag CONFIRMED_LONG (or when no THELLO received from TN3 after an interval of time), TN2 checks in its TunnelT table if it has already the entry

about the tunnel; If so, it starts a repair process (figure 5-16), otherwise, it ignores the received THELLO.

The repair process is based on two messages: Tunnel Creation Update (TCRT_U) and Tunnel Acknowledgement (TACK_U) messages. After receiving THELLO with CONFIRMED LONG flag and long distance from TN3, TN2 makes the entry flag of the tunnel as IN_REPARATION then broadcasts TCRT_U message in the network to select another tunnel node TN3' nearby.

TCRT_U is treated similarly as the TREQ message and it contains almost the same fields as shown in figure 5-17.

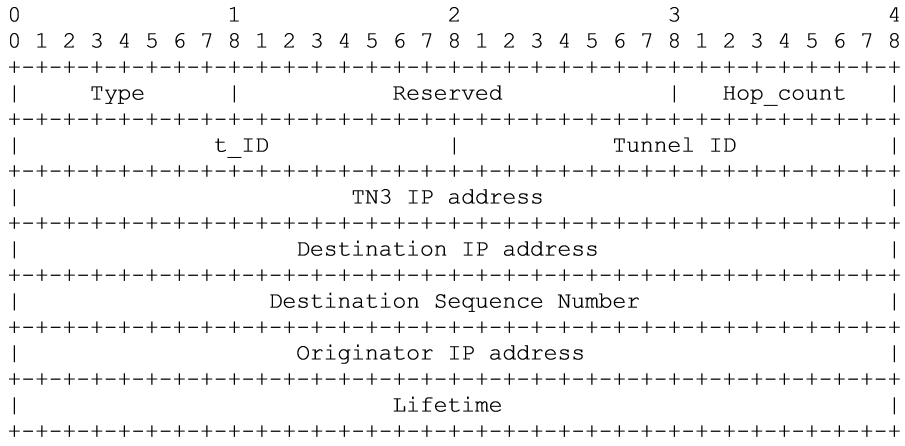


Type: Type of the AODV message
Reserved: Not used
T: Tunnel type (outbound or inbound)
t_ID: Temporary identifier generated by TN2 to identify its address
Hop Count: Hop count between tunnel nodes
nb_I: Number of intermediate nodes between TN2 and the new TN3
Tunnel ID: Identifier of the tunnel in the network
TCRT_U ID: Identifier of the TCRT_U message
Originator IP Address: IP address of TN2
Originator Sequence Number: Sequence Number of TN2

Figure 5-17: Tunnel Creation Update (TCRT_U) message format

The selection of TN3' is based on a random number of hops chosen by TN2. At each reception, the node decreases the number of hops until reaching the distance required between TN2 and the new third tunnel node (TN3'). TN2 indicates in TCRT_U the type of the created tunnel already existed, in addition to a new TID different from the real TID of the tunnel.

After reaching the number of hops required, the concerned node (TN3') creates an entry in its TunnelT table about the existed tunnel with a new TID generated by TN3'. Then, it replies with an encrypted TACK_U to TN2 including its address and the new TID as shows figure 5-18.



Type: Type of the AODV message
Reserved: Not used
Hop Count: Hop count between tunnel nodes
t_ID: The temporary identifier sent by TN2
Tunnel ID: Identifier of the tunnel in the network
TN3 IP address: IP address of TN3
Destination IP Address: IP address of the sender of TACK_U
Destination Sequence Number: Sequence Number of the sender of TACK_U
Originator IP Address: IP address of the receiver tunnel node of TACK_U
Lifetime: Lifetime of TACK_U (in milliseconds)

Figure 5-18: Tunnel Acknowledgement Update (TACK_U) format

When TN2 receives TACK_U, it places the address of TN3' and the new TID in the entry of the tunnel in its tunnel table and makes the entry as CREATED. Then TN2 forwards TACK_U to TN1, which updates the entry about the tunnel in its TunnelT table. If this tunnel is inbound, TN1 sends the new TN3's identity and the new TID to the NetDB database to update the previous TN3 identity and TID.

b. Maintenance of TN1

After sending a TACK message to TN1 (during the tunnel creation process), TN2 waits for a period of time (T2), if no THELLO had been received during this interval, it considers that the route to TN1 is ruptured (figure 5-19). In this case, TN2 deletes the entry of the tunnel from its TunnelT table, then sends an encrypted THELLO message with CONFIRMED_LONG flag and with a high number of hops to TN3.

When TN3 receives the THELLO message, it deletes the entry about the tunnel from its TunnelT table.

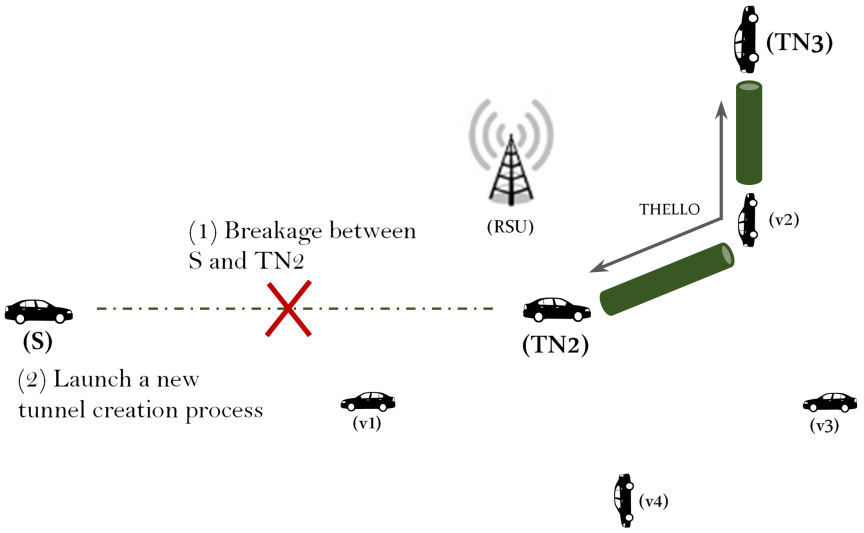


Figure 5-19: Tunnel maintenance of TN1

When TN1 receives TACK from TN2, it starts sending THELLO messages (encrypted) to TN2 for each interval of time (T3). Similarly to the previous algorithm of the tunnel maintenance, when TN2 receives THELLO, it checks the state declared by TN1 and the number of hops.

> - If the distance is acceptable, TN2 resends an encrypted THELLO message to TN1 indicating that the state of the route is NORMAL.
> - If the distance to TN1 is long (more than the threshold) and the state of the route was indicated as NORMAL, TN2 replies TN1 by an encrypted THELLO message with LONG flag, if TN1 has declared the state as CONFIRMED_LONG, TN2 deletes the entry of the tunnel from its TunnelT table, then sends an encrypted THELLO to TN3 with CONFIRMED_LONG flag and a high number of hops as described in the previous algorithm ("Maintenance of TN2").

Similarly, at reception of THELLO from TN2, TN1 does the same verification about the two parameters the state and the number of hops. It processes like TN2 except when the state was declared by TN2 as CONFIRMED_LONG and the distance is long. In this case or when no THELLO received after an interval of time, TN1 considers that the tunnel is broken. Thus, it deletes the tunnel from its TunnelT table and sends an encrypted THELLO message to TN2 with CONFIRMED_LONG flag and a high number of hops. TN2 deletes the tunnel from its TunnelT table and forwards the message to TN3. Similarly, TN3 deletes the tunnel from its TunnelT table. At the same time, TN1 launches the tunnel creation process to create a new tunnel as represents figure 5-19. All this process is detailed in the algorithm 21.

Algorithm 21 The maintenance algorithm of TN1

```
if FLAG = C_LONG then
    Delete the entry;
    // Launch the tunnel creation process;
else if nb_hop <= Top then
    FLAG ← NORMAL;
    Send e_Pk(TN2) (THELLO) to TN2;
else if FLAG = NORMAL then
    FLAG ← LONG;
    Send e_Pk(TN2) (THELLO) to TN2;
else
    FLAG ← C_LONG;
    Send e_Pk(TN2) (THELLO) to TN2;
    // Launch the tunnel creation process
end if
```

The followed section analyzes the security and performance of the second version of the proposed protocol during the communication.

5.5.3 Performance and security analysis

In this section, we show the anonymity and security of the enhanced version of the proposed protocol compared to the previous version and AODV protocol. Similarly to part 1 of this contribution, the comparison is based on three QoS parameters: packet delivery ratio, end-to-end delay and overhead.

5.5.3.1 Security analysis

The improved version of the proposed protocol uses the same mechanisms and algorithms to provide anonymity of communication and secure messages. It is based on using tunnels and encryption algorithms inspired by the I2P protocol (as detailed in the previous part of the contribution in section 5.4). Besides, it uses a tunnel maintenance algorithm to maintain the existence of tunnels in the network. The tunnel creation part is discussed in the previous work in section 5.4.3.1. In this section, we analyze the added part related to the tunnel maintenance process. This process is detailed into three phases: the THELLO exchange (section 5.5.2.1), maintenance of TN2 (section 5.5.2.2) and maintenance of TN1 (which represents the tunnel creation process developed in the previous part of the contribution).

The THELLO exchange is based on using THELLO messages. These messages include the tunnel ID, which must be secret between the tunnel

nodes during the communication. For that, THELLO is encrypted between each successive nodes in the tunnel (using an asymmetric encryption algorithm). Thus, only the communicating tunnel nodes can know the content of THELLO messages.

During the maintenance process of TN2, two messages are used: Tunnel Creation Update (TCRT_U) and Tunnel acknowledgement update (TACK_U) messages (in addition to the encrypted THELLO message). TCRT_U is not encrypted, however, its content is not critical to the point where it threatens the anonymity of the communication when an intermediate node read this content. TCRT_U includes the TID used before the breakage. When the new TN3 is selected, it ignores the TID sent in the TCRT_U message and generates a new TID for the tunnel. After that, it sends it in the TACK_U message to TN2. TACK_U is encrypted using an asymmetric algorithm, which prevents intermediate nodes to know this TID sent in the TACK_U message.

For the tunnel node TN1, the maintenance of the tunnel consists on sending an encrypted THELLO message to TN2 to delete the tunnel, then launching the tunnel creation process from the beginning as detailed in the previous part of the contribution in section 5.4.2.2. the security analysis of this process is discussed in section 5.4.3.1.

5.5.3.2 Performance analysis

In this part of the contribution, we present a simulation of three protocols: AODV, the first version and the new version of the proposed protocol. We launch the simulation in the NS3 platform and we show the results using box plots. Similarly to the previous part, the comparison between these protocols is based on three QoS parameters: the packet delivery ratio, end-to-end delay and overhead.

The simulation is launched for six scenarios with a different number of nodes and for a duration of 150s. We use scenarios with different number of nodes to see the impact of this number on the proposed protocol. The speeds of vehicles are chosen between 20 and 35 km/h as average speeds in urban environments where the limited speed is up to 50km/h. Table 5.5.3.2 presents the settings used in the simulation.

The use of box plots to show the simulation results refers to the way of generation of random values in NS3. These values are based on the "seed" value, which causes differences in the result values even for the same scenario. For that, we launch the simulation many times each scenario to have a set with different possible values during the simulation. Then, we use box plots for each set (each scenario) to show the distribution and concentration of values.

Area size	3320*2035m
Number of nodes	10 to 20
Speed of nodes	7m/s, 9m/s
Number of communications	3
Number of messages per 2 seconds	1
Simulation time	150s
Packet size	256
Routing protocol	AODV
Propagation model	Friis
Transmission power	0dbm
Frequency band	5GHz
Transmission bandwidth	5MHz

Table 5.3: Simulation settings 2

As represents the figure 5-20, the packet delivery ratio in the previous version of the proposed protocol is low, in which, the values are mainly between 0% and 5%. That makes the first version inappropriate to VANet due to the mobility of vehicles.

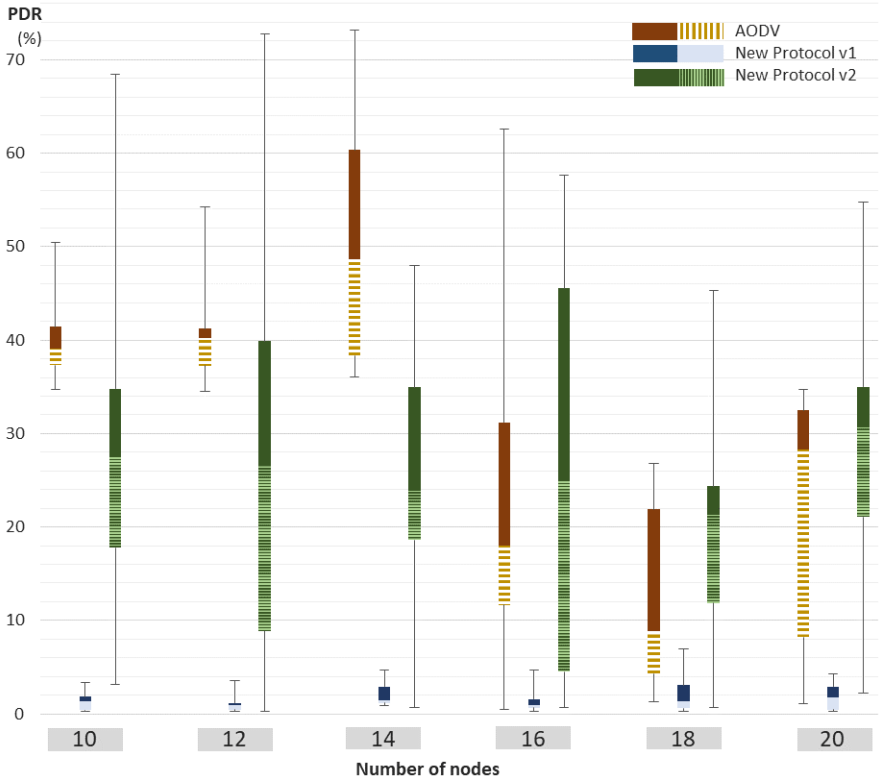


Figure 5-20: Packet Delivery Ratio of the proposed protocol (version 1 and 2) and AODV

In version 1 of the proposed protocol, the mobility of vehicles is not taken into account by that version. Besides, no maintenance process is integrated to maintain the existence of tunnels. This version is dedicated to being implemented statically where no mobility, thus, tunnels can be established for a long time.

In real VANet, tunnels can be broken at any time, which can cut the communication and requires to launch the tunnel creation process many times.

In the new version, the PDR is high compared to the previous version of the protocol thanks to the proposed algorithm of maintenance of tunnels, in which, the majority of PDR values are generally between 10% and 40%, which are significant compared to the previous version (between 0% and 5%).

Compared to AODV, the new version of the protocol has a PDR with low values in small scenarios (10, 12 and 14 nodes), and with relatively high values in other scenarios (with high number of nodes), which shows the ability of the proposed protocol to be implemented in real VANet with high number of vehicles.

Concerning the end-to-end delay, it is noticed in figure 5-21 that the proposed protocol (version 1 and 2) is better than the AODV protocol from the routing point of view.

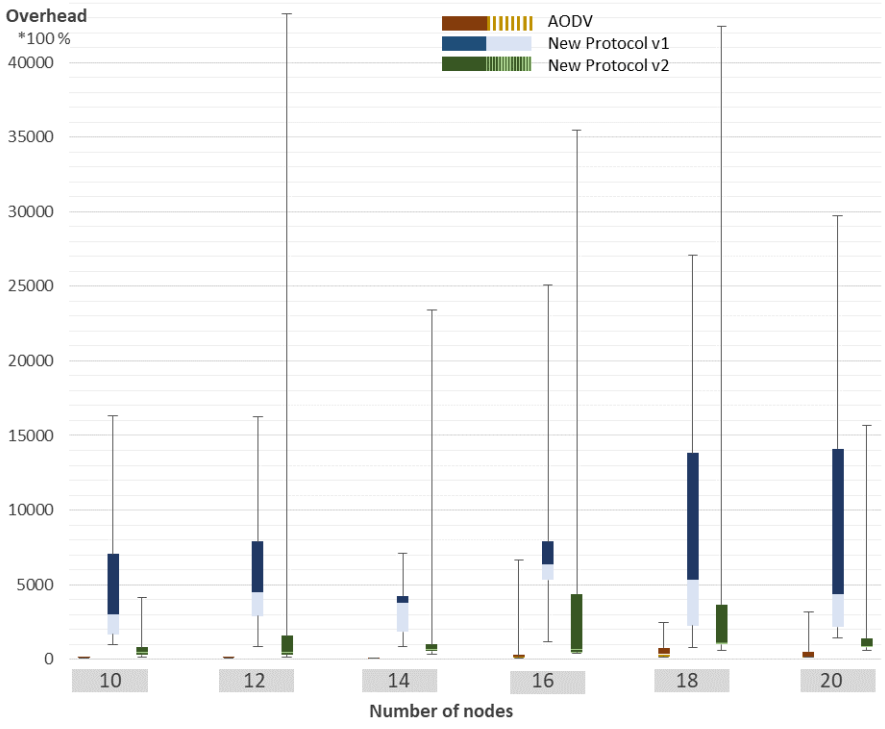


Figure 5-21: End-to-end delay of the proposed protocol (version 1 and 2) and AODV

In AODV, the end-to-end delay values are mostly between 0.1s and 0.6s, which make the difference with the proposed protocol that has an end-to-end delay with values between 0.004s and 0.02s.

The AODV protocol has an end-to-end delay with high values compared to the proposed protocol because of the on-demand process to discover routes between the communicant nodes. This process can be launched frequently because of the mobility of vehicles. However, in the proposed protocol, the on-demand process is launched during the tunnel creation and before the communication, which helps to discover several routes between nodes in the network. The discovered routes can exist thanks to the periodic exchanged messages used in the creation and maintenance of tunnels (version 2). As a result, the communicant nodes can use these routes without launching the on-demand process (the problem with AODV).

Adding security algorithms in VANet can cause a degradation in some QoS parameters. In the proposed protocol, we integrate algorithms for creating and maintaining tunnels and encrypting data, which use messages more than AODV. The first version of the protocol uses TREQ, TREP, TCRT and TACK, and the second version uses more three messages: THELLO, TCRT_U and TACK_U. That is why our protocol has high values for the overhead compared to the AODV protocol as represents figure 5-22.

The second version of the proposed protocol is better than the first one in term of overhead. This improvement refers to the integrated tunnel maintenance process. When a tunnel is broken (very frequently), the protocol (version 1) requires to relaunch the tunnel creation process from the beginning, which generates a high number of messages. However, in the second version, the tunnels are maintained most of the time using THELLO messages and (TCRT_U and TACK_U messages when breakage occurs). For that, the number of messages used in the second version is less than the first one.

As a conclusion for this discussion, we can say that the second version of the proposed protocol is better than the first one thanks to the integrated tunnel maintenance algorithm. Even by providing security and anonymity in VANe, this protocol can allow an acceptable level of performance compared to the AODV protocol, while mentioning the high values of the overhead, which refers to the additional messages used in this protocol to provide security and anonymity of the communication.

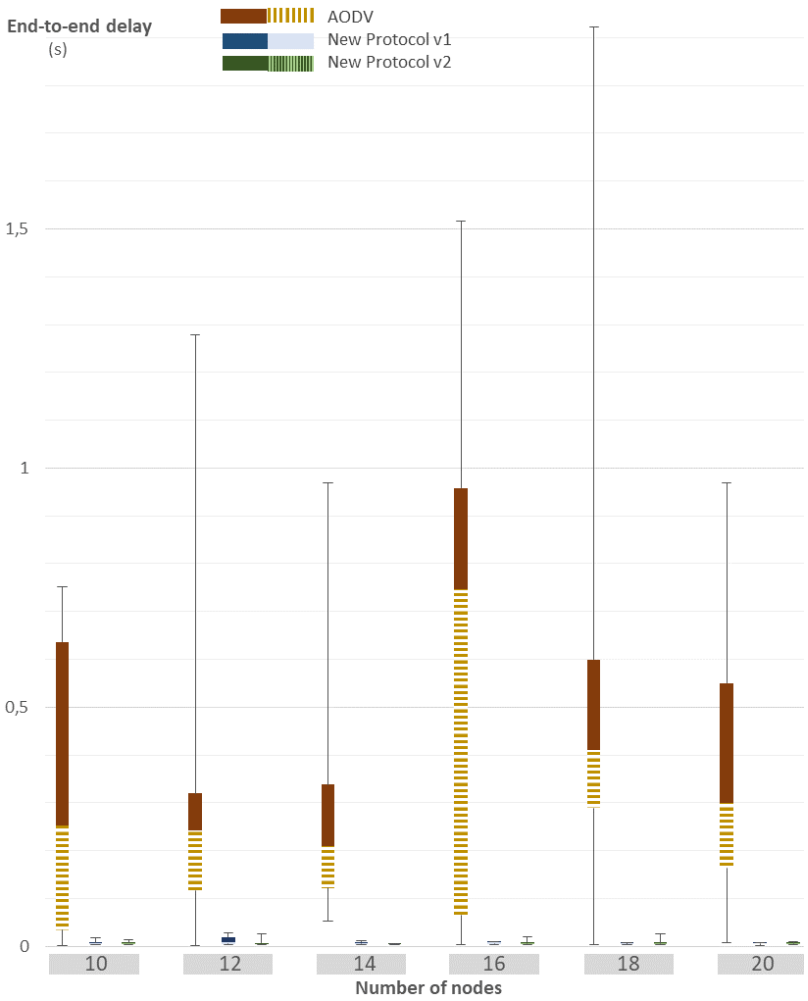


Figure 5-22: Overhead of the proposed protocol (version 1 and 2) and AODV

5.6 Conclusion

Due to the high level of security and anonymity in the I2P protocol, it can be used as a reference model to provide security and anonymity in VANet. I2P mechanisms are designed for the internet, which has different characteristics from VANet. Implementation of I2P protocol in VANet needs an adaptation of its algorithms and mechanisms according to VANet requirements.

In this chapter, the second contribution is proposed to continue our project to adapt the I2P model to VANet. This contribution is divided into two parts:

The part 1 treats the second part of the work. We develop a tunnel creation mechanism and encryption algorithms inspired by the I2P protocol. As the first version of our protocol, the tunnel creation mechanism is

simplified by creating tunnels statically in the network and without maintenance. We assumed that tunnels have a long lifetime. The encryption of messages is provided using the tunnel and garlic encryption layers used in I2P, in which each layer is represented by an asymmetric encryption algorithm.

In Part 2, we create the final version of the proposed protocol by developing the tunnel maintenance algorithm. This algorithm is integrated into the last protocol version to provide the creation and maintenance of tunnels in VANet. In this case, tunnels can be created and maintained in real mobile scenarios.

To compare between the two versions of the proposed protocol and AODV, we launch simulations of some scenarios with different settings. The results show that the second version is better than the first one. This improvement refers to the integrated tunnel maintenance algorithm.

After analysing the results, we can say that version 2 is acceptable from the PDR and end-to-end delay point of view. However, it has a high overhead compared to AODV. Which makes this version discussable to be improved and has good results.

In the end, we become able to implement all important I2P mechanisms and algorithms providing security and anonymity in the network. Therefore, we have achieved our objective to secure and anonymize the communication in the vehicular ad hoc network.

CONCLUSION AND PERSPECTIVES

Conclusion

The research work in the field of vehicular ad hoc networks aims to improve their operation and make the transport safer, more secure, more efficient, more reliable and ecological. This type of networks constitutes an area where a lot of challenges and issues facing its evolution. This requires more studies and research to propose new solutions and realize network services by guaranteeing a certain level of security and quality of service at a low cost.

The field of research in this thesis concerns the security of VANet, specifically, it is about the anonymity of communication. Our goal is to establish a secure and anonymous communication in VANet. To achieve that, we have proposed a secure protocol inspired by the I2P protocol, where we adapt its algorithms and mechanisms to several requirements of VANet.

This thesis is organized into five chapters:

- the first three chapters are focused on the theoretical part, speaking in general about VANet and security in this type of networks, as well as a presentation of the I2P protocol.

- The last two chapters concern the practical part, where we present the contributions realized during this thesis.

In the first part, chapter 1, we present generally different axes of VANet, in which we talk about characteristics, applications, architectures and components of VANet. We present some known mobility models and routing protocols, in addition to standards and technologies implemented in this network. Then, we show some challenging issues focusing on the security field in that kind of networks.

In chapter 2, we continue the theoretical part by talking about security services and QoS parameters, followed by a classification of different known attacks in VANet and proposed solutions for some of these attacks in the literature.

The last chapter in the theoretical part is focused on the anonymity in VANet using the I2P protocol. We detail the anonymity concept by

presenting I2P, Freenet and Tor as the most known protocols providing security and anonymity on the Internet. After that, we argue our choice to I2P as a model of reference to develop a protocol that provides secure and anonymous communication in VANet. Then, We present in detail different I2P concepts and mechanisms related to our work. We describe the tunnel creation process and mechanism of exchanging messages inside and between tunnels. In addition to present the encryption layers implemented in this protocol. Then, we discuss the difference between VANet and I2P network and how we applied I2P in VANet.

Moving to the practical part, in chapter 4, we explain the first contribution, in which we implement the I2P transport encryption layer. We propose a secure model based on encryption algorithms and digital signature mechanisms to provide a secure communication.

The last chapter in the practical part presents the second contribution. We continue the previous work by implementing the tunnel and garlic encryption layers of I2P and creating tunnels in part 1. We suppose that tunnels are created statically in VANet to simplify the tunnel creation process. Thereafter, we treat the mobility of the network in part 2. We create the complete version of the new protocol by developing a tunnel maintenance algorithm allowing to maintain the existence of tunnels during the communication.

During these contributions, we have found that I2P is a heavy protocol, which requires to choose adaptable algorithms in I2P to use or adapt them to the critical requirements of VANet, such as high mobility, rapid change of topology and variety of communication environments. In these contributions, we try to reduce the number of messages, resume some of the I2P mechanisms and so on to have acceptable QoS parameters like reducing the time needed to complete tasks and minimizing overhead during the communication.

The experimentations executed and the scenarios used in our contributions are relatively limited, in which they are not implemented in real VANet (outside experimentations), but they are based on simulations using NS3 simulator. For that, we show results based on these simulations.

Perspectives and future works

Our thesis is about the anonymity of communication in VANet. The desired work here is to develop algorithms and mechanisms to anonymize and secure communication. Contributing to this issue, we propose a secure protocol inspired by the I2P protocol. We develop a set of algorithms and mechanisms providing anonymity and security of communication and responds to different requirements of VANet. However, there are still some issues to explore. Each of these issues is a potential perspective in the continuity of this thesis.

The proposed protocol in this thesis is a version that needs to be improved to provide a high level of security and performance in VANet. Our perspectives for this work is to enhance the proposed algorithms in the proposed contributions to better respond to VANet characteristics while maintaining a satisfactory level of security and anonymity in VANet. Another aspect is to profile nodes where each node will be assigned by a profile about its capabilities such as the speed of calculation, range of communication and speed of the vehicle. Therefore, instead of selecting any node in the network to be in the tunnel, the creator checks the capacities of nodes existed then select the best potential tunnel nodes. That process helps to create tunnels with high performance and makes exchange and encryption of messages faster than before.

Another work to do, concerns the tunnel creation process. In our contributions, we create tunnels through two steps; "Getting tunnel nodes identities" then "Tunnel creation". As an improvement to this process, we will merge the two steps into one-step, in which we obtain the identities of tunnel nodes and create tunnels in the same time, which reduces the time needed to complete the tunnel creation process compared to the proposed protocol in this thesis.

In this thesis, the proposed protocol uses the AODV routing protocol and it is executed with a speed less than 35km/h. As future contributions, we aim to use other routing protocols, implement realistic propagation models and rise the speed of vehicles.

List of acronyms

VANet	Vehicular Ad hoc Network
MANET	Mobile Ad hoc Network
RSU	Road Side Unit
I2P	Invisible Internet Project
MITM	MAn-In-The-Middle attack
NetDB	the Network Database in I2P
GHG	Greenhouse Gas
ITS	Intelligent Transport System
NTIC	New Information and Communication Technology
V2V	Vehicle-to-Vehicle communication
IVC	Inter-Vehicular communication
V2I	Vehicle-to-roadside communication
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
DGPS	Differential Global Positioning System
RTK	Real-Time Kinematics
HMI	Human-Machine Interface
OBU	On-Board Unit
DSRC	Dedicated Short Range Communication
AODV	Ad hoc On-demand Distance Vector
ACB	Prediction Based Routing
MURU	Multi-hop Routing protocol for Urban VANet
OLSR	Optimized Link State Routing protocol
RBVT-P	Road-Based Vehicle Traffic Routing protocol
ZRP	Zone Routing Protocol
ETSI	European Telecommunications Standardization Institute
GPSR	Greedy Perimeter Stateless Routing
GPCR	Greedy Perimeter Coordinator Routing
COIN	Clustering for open IVC network
CAN	Control Area Network

ADAS	Advanced Driver Assistance System
ABS	Anti-lock Braking System
ASR	Acceleration Slip Regulation system
GSM	Global System for Mobile communication
GPRS	General Packet Radio Service
UMTS	Universal Mobile Telecommunication System
RDS/TMC	Radio Data System/Traffic Message Channel
DVB-T/DVB-H	Digital Video Broadcasting-Terrestrial/ Digital Video Broadcasting-Handheld
WiMAX	Worldwide Interoperability for Microwave Access
DSRC	Dedicated Short Range Communication
WAVE	Wireless Access for Vehicular Environments
WSM	WAVE Short Message
WSMP	WAVE Short Message Protocol
WSA	WAVE Service Advertisement
MI-interface	Interface between the management entity and the access layer in the ETSI TC ITS V2X Reference Architecture
MN-interface	Interface between the management entity and the networking and transport layer in the ETSI TC ITS V2X Reference Architecture
MF-interface	Interface between the management entity and the facilities layer in the ETSI TC ITS V2X Reference Architecture
AM-interface	Interface between management entity and ITS station applications in the ETSI TC ITS V2X Reference Architecture
MS-interface	Interface between the management entity and the security entity in the ETSI TC ITS V2X Reference Architecture
LDW	Local Danger Warning
SHA	Secure Hash Algorithm
MD5	Message Digest 5
TPD	Temper Proof Device
PDR	Packet Delivery Ratio
VoIP	Voice over Internet Protocol
DoS	Deny of Service
DDoS	Distributed Deny of Service
Tor	The Onion Router
TCP	Transmission Control Protocol

UDP	User Datagram Protocol
SSU	Secure User datagram protocol
NTCP	NIO-Based Transmission Control Protocol
I2NP	I2P Network Protocol
I2CP	I2P Client Protocol
TBM	Tunnel Build Message
VTBM	Variable Tunnel Build Message
TBRM	Tunnel Build Reply Message
VTBRM	Variable Tunnel Build Reply Message
DSM	Database Store Messages
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
DSA	Digital Signature Algorithm
TA	Trusted Authority
NS3	Network Simulator 3

Bibliography

[1] automotive intelligent transport systems (its). https://www.etsi.org/technologies/automotive-intelligent-transport. Accessed: 2019.

[2] Bob - basic open bridge. https://geti2p.net/fr/docs/api/bob. Accessed: 2019.

[3] Browse privately. explore freely. https://www.torproject.org/. Accessed: 2019.

[4] Datagram specification. https://geti2p.net/spec/datagrams. Accessed: 2019.

[5] Elgamal/aes + sessiontag encryption. https://geti2p.net/fr/docs/how/elga- malaes. Accessed: 2019.

[6] End-to-end delay. https://ipfs.io/ipfs/QmXoypizjW3WknFiJnKLwHCnL72vedxjQkDDP1mXWo6uco/wiki/End-to-end_delay.html. Accessed: 2019.

[7] The european telecommunications standards institute. https://www.etsi.org/. Accessed: 2019.

[8] Flexibility (engineering). https://en.wikipedia.org/wiki/Flexibility_(engineering). Accessed: 2019.

[9] Freenet. https://freenetproject.org/fr/index.html. Accessed: 2019.

[10] A gentle introduction to how i2p works. https://geti2p.net/en/docs/how/intro. Accessed: 2019.

[11] I2np specification. https://geti2p.net/spec/i2np. Accessed: 2019.

[12] I2ptunnel. https://geti2p.net/fr/docs/api/i2ptunnel. Accessed: 2019.

[13] Intelligent transport systems (its); communications architecture. https://www.etsi.org/deliver/etsi_en/302600_302699/302665/01.01.01_60/en_302665v010101p.pdf. Accessed: 2019.

[14] Intelligent transport systems (its); osi cross-layer topics; part 3: Interface between management entity and access layer. https://www.etsi.org/deliver/etsi_ts/102700_102799/10272303/01.01.01_60/ts_10272303v010101p.pdf. Accessed: 2019.

[15] Intelligent transport systems (its); osi cross-layer topics; part 4: Interface between management entity and networking transport layer. https://www.etsi.org/deliver/etsi_ts/102700_102799/10272304/01.01.01_60/ts_10272304v010101p.pdf. Accessed: 2019.

[16] Intelligent transport systems (its); osi cross-layer topics; part 5: Interface between management entity and facilities layer technical specification. https://www.etsi.org/deliver/etsi_ts/102700_102799/10272305/01.01.01_60/ts_10272305v010101p.pdf. Accessed: 2019.

[17] Intelligent transport systems (its); security; security header and certificate formats. https://www.etsi.org/deliver/etsi_ts/103000_103099/103097/01.03.01_60/ts_103097v010301p.pdf. Accessed: 2019.

[18] Intelligent transport systems; osi cross-layer topics; part 6: Interface between management entity and security entity. Technical report.

[19] International telegraph and telephone consultative committee. https://uia.org/s/or/en/1100050019. Accessed: 2019.

[20] The invisible internet project. https://geti2p.net/en/. Accessed: 2019.

[21] ns-3 tutorial. https://www.nsnam.org/docs/tutorial/html/index.html. Accessed: 2018.

[22] Protocol stack. https://geti2p.net/fr/docs/protocol. Accessed: 2019.

[23] Sam v1 specification. https://geti2p.net/fr/docs/api/sam. Accessed: 2019.

[24] Sam v2 specification. https://geti2p.net/fr/docs/api/samv2. Accessed: 2019.

[25] Sam v3. https://geti2p.net/fr/docs/api/samv3. Accessed: 2019.

[26] Socks. https://geti2p.net/fr/docs/api/socks. Accessed: 2019.

[27] Streaming library. https://geti2p.net/fr/docs/api/streaming. Accessed: 2019.

[28] Tor: Overview. https://2019.www.torproject.org/about/overview.html.en. Accessed: 2019.

[29] Traceability. https://en.wikipedia.org/wiki/Traceability. Accessed: 2019.

[30] Tunnel routing. https://geti2p.net/fr/docs/how/tunnel-routing. Accessed: 2019.

[31] Velocity. https://en.wikipedia.org/wiki/Velocity. Accessed: 2019.

[32] What is non-repudiation? https://www.cryptomathic.com/products/authentication-signing/digital-signatures-faqs/what-is-non-repudiation. Accessed: 2019.

[33] Fips 180-2, secure hash stand ard, federal information processing stand ard (fips). *National Institute of Stand ards and Technology*, 2002.

[34] Information technology - security techniques - evaluation criteria for it security. ISO/IEC, 2005.

[35] E.800 : Definitions of terms related to quality of service. ITU-T Rec, 2008.

[36] Imad Aad, Jean-Pierre Hubaux, and Edward W Knightly. Denial of service resilience in ad hoc networks. In *Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 202–215. ACM, 2004.

[37] M. Tahar abbes. *Proposition d'un protocole à économie d'énergie dans un réseau hybride GSM et AD HOC*. PhD dissertation, University of Oran, 2012.

[38] S. ALLAL. *Optimisation des échanges dans le routage géocast pour les réseaux de véhicules Ad hoc VANETs*. PhD dissertation, University of Paris 13, Villetaneuse, France, 2014.

[39] A. A. Aouiz, S. H. Boukli, P. Lorenz, and M. Gilg. Network life time maximization of the aomdv protocol using nodes energy variation. *Network Protocols and Algorithms*, 10(2), 2018.

[40] M. Azees, P. Vijayakumar, and D. L. Jegatha. Eaap: Efficient anonymous authentication with conditional privacy-preserving scheme for vehicular ad hoc networks. *IEEE Transactions on Intelligent Transportation Systems*, 18:2467–2476, 2017.

[41] G. Bellikar, A. Bhatia, R. Hansdah, and S. Singh. 3taav: A three-tier architecture for pseudonym-based anonymous authentication in vanets. pages 420–425. International Conference on Information Networking, 2018.

[42] S. BITAM and A. MELLOUK, editors. ISTE Ltd and John Wiley Sons, Inc, 27-37 St George's Road, London SW19 4EU, UK. 111 River Street, Hoboken, NJ 07030, USA, first edition, 2014.

[43] R. Braden. Requirements for internet hosts – communication layers. RFC 1122, October 1989.

[44] T.H. Clausen and P. Jacquet. Optimized link state routing (olsr). RFC 3626, October 2003.

[45] T. Diab, M. Gilg, F. Drouhin, and P. Lorenz. Anonymizing communication in vanets by applying i2p mechanisms. IEEE Globecom'19, Waikoloa, USA, 2019.

[46] T. Diab, M. Gilg, and P. Lorenz. A secure communication model using lightweight diffie-hellman method in vehicular ad hoc networks. *International Journal of Security and Networks*, 14(2):61–77, 2019.

[47] Xia Feng, Chun yan Li, De xin Chen, and Jin Tang. A method for defensing against multi-source sybil attacks in vanet. *Peer-to-Peer Networking and Applications*, 10(2):305–314, 2017.

[48] T. Gao, X. Deng, N. Guo, , and X. Wang. An anonymous authentication scheme based on pmipv6 for vanets. *IEEE Access*, 6:14686–14698, 2018.

[49] Pengwenlong Gu, Rida Khatoun, Youcef Begriche, and Ahmed Serhrouchni. k-nearest neighbours classification based sybil attack detection in vehicular networks. Mobile and Secure Services (MOBISECSERV), 2017.

[50] Rejab HAJLAOUI. *Résolution à base d'heuristiques du problème de routage dans les réseaux ad hoc de véhicules.* PhD dissertation, University of Franche-Comté, 2018.

[51] Clarke I., Sand berg O., Wiley B., and Hong T.W. Freenet: A distributed anonymous information storage and retrieval system. *Federrath H. (eds) Designing Privacy Enhancing Technologies. Lecture Notes in Computer Science*, 2009:46–66, 2001.

[52] Kang J., Lin D., Jiang W., and Bertino E. Highly efficient rand omized authentication in vanets. *Pervasive and Mobile Computing, Elsevier*, 44:31–44, 2018.

[53] David Johnson, Charles Perkins, Jari Arkko, et al. Rfc 3775: Mobility support in ipv6. *IETF, June*, pages 1–165, 2004.

[54] A. A. Kahina. *Modélisation et étude de performances dans les réseaux VANET*. PhD dissertation, University of Technology of Belfort-Montbeliard, 2012.

[55] Chaker Abdelaziz Kerrache, Nasreddine Lagraa, Carlos T.Calafateb, and Abderrahmane Lakas. Tfdd: A trust-based framework for reliable data delivery and dos defense in vanets. *Vehicular Communications, Elsevier*, 9:254–267, 2017.

[56] Diyar Khairi and Amine Berqia. Survey on qos and security in vehicular ad hoc networks. *International Journal*, 5(5), 2015.

[57] Ahmed Korichi, Abderrahmane Lakas, and Mohammed El Amine Fekair. An efficient qos-compliant routing scheme for vanet. In *2016 5th International conference on electronic devices, systems and applications (ICEDSA)*, pages 1–4. IEEE, 2016.

[58] Deepak Kushwaha, Piyush Kumar Shukla, and Raju Baraskar. A survey on sybil attack in vehicular ad-hoc network. *International Journal of Computer Applications*, 98(15), 2014.

[59] J. Li, K. R. Choo K, W. Zhang, S. Kumari, J. Rodrigues, K. Khan, and D. Hogrefe. Epa-cppa: An efficient, provably-secure and anonymous conditional privacy-preserving authentication scheme for vehicular ad hoc networks. *Vehicular Communications*, pages 104–113, 2018.

[60] Jonathan Loo, Jaime Lloret Mauri, and Jesús Hamilton Ortiz, editors. ISTE Ltd et John Wiley Sons, Inc, Boca Raton, Florida, United States, first edition, 2016.

[61] Abir Mchergui, Tarek Moulahi, Bechir Alaya, and Salem Nasri. A survey and comparative study of qos aware broadcasting techniques in vanet. *Telecommunication Systems*, 66(2):253–281, 2017.

[62] Preeti Nagrath and Bhawna Gupta. Wormhole attacks in wireless adhoc networks and their counter measurements: A survey. In *2011 3rd International Conference on Electronics Computer Technology*, volume 6, pages 245–250. IEEE, 2011.

[63] Hassan Noura and Wassim Znaidi. Security of cooperative intelligent transport systems: Standards, threats analysis and cryptographic countermeasures. *Electronics*, 4, 07 2015.

[64] C. Perkins. Ad hoc on-demand distance vector (aodv) routing. RFC 3561, July 2003.

[65] J. Petit. Surcoût de l'authentification et du consensus dans la sécurité des réseaux sans fil véhiculaires.

[66] S.M. Pournaghi, , B. Zahednejad, , M. Bayat, and Y. Farjami. Necppa: a novel and efficient conditional privacy-preserving authentication scheme for vanet. *Computer Networks, Elsevier*, 134:78–92, 2018.

[67] Rivest R. The md5 message digest algorithm. RFC 1321, April 1992.

[68] Varsha Raghuwanshi and Simmi Jain. Denial of service attack in vanet: A survey. *International Journal of Engineering Trends and Technology (IJETT)*, 28(1):15–20, 2015.

[69] M. Saddiki, H. S. Boukli, P. Lorenz, and M. Gilg. Black hole attack detection and ignoring in olsr protocol. *International Journal of Trust Management in Computing and Communications*, 4(1), 2017.

[70] F. Sakiz and S. Sen. A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov. *Ad hoc Networks, Elsevier*, 61:33–50, 2017.

[71] Fatih Sakiz and Sevil Sen. A survey of attacks and detection mechanisms on intelligent transportation systems: Vanets and iov. *Ad Hoc Networks, Elsevier*, 61:33–50, 2017.

[72] B. T. Sharef, R. A. Alsaqour, and M. Ismail. Vehicular communication ad hoc routing protocols: A survey. *Journal of Network and Computer Applications, Elsevier*, 2013.

[73] Muhammad Sheikh and Jun Liang. A comprehensive survey on vanet security services in traffic management system. *Wireless Communications and Mobile Computing*, 2019:1–23, 09 2019.

[74] M. L. SICHITIU and M. KIHL. Inter-vehicule communication systems : A survey. *IEEE Communications Surveys Tutorials*, 2008.

[75] J. Soryal and T. Saadawi. Dos attack detection in internet-connected vehicles. pages 7–13. International Conference on Connected Vehicles and Expo (ICCVE).

[76] R. Sugumar, A. Rengarajan, and C. Jayakumar. Trust based authentication technique for cluster based vehicular ad hoc networks (vanet). *Wireless Networks, Springer Science + Business Media*, 24:373–382, 2018.

[77] Gao T., Deng X., Guo N., and X Wang. An anonymous authentication scheme based on pimpv6 for vanets. *IEEE Access*, 6:14686–14698, 2016.

[78] Fan-Hsun Tseng, Li-Der Chou, and Han-Chieh Chao. A survey of black hole attacks in wireless mobile ad hoc networks. *Human-centric Computing and Information Sciences*, 1(1):4, 2011.

[79] Diffie W. and Hellman E.M. New directions in cryptography. *IEEE Transactions On Information theory*, 22(2):644–654, 1976.

[80] S. Wang and N. Yao. Liap: a local identity-based anonymous message authentication protocol in vanets. *Computer Communications, Elsevier*, 112:154–164, 2017.

[81] Guo X., Chen C., Gong C., and Leu F. A secure official vehicle communication protocol for vanet. page 482–485. 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2016.

[82] Li X., Li S., Hao J., Feng Z., and An B. Optimal personalized defense strategy against man-in-the-middle attack. page 593–599. Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI-17), 2017.

[83] C. Xu, X. Huang, M. Ma, and H. Bao. An anonymous hand over authentication scheme based on lte-a for vehicular networks. *Wireless Communications and Mobile Computing*, pages 1–15, 2018.

[84] H. Yiliang, L. Xi, J. Di, and F. Dingyi. Attribute-based authenticated protocol for secure communication of vanet. page 4078–4081. 29th Chinese Control and Decision Conference (CCDC), 2017.

[85] B. Ying and A. Nayak. Anonymous and lightweight authentication for secure vehicular networks. *IEEE Transactions on Vehicular Technology*, 66(12):10626–10636, 2015.

[86] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan. Vehicular ad hoc networks (vanets) : status, results and challenges. *Telecommunication Systems journal (Springer Science Media, LLC 2010)*, 50(4):217–241, 2012.

[87] L. Zhang, Q. Wu, J. Domingo-Ferrer, B. Qin, and C. Hu. Distributed aggregate privacy-preserving authentication in vanets. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):516–526, 2016.

[88] H. Zhong, B. Huang, J. Cui, Y. Xu, and L. Liu. Conditional privacy-preserving authentication using registration list in vehicular ad hoc networks'. *IEEE Access*, 6:2241–2250, 2018.