UNIVERSIDADE FEDERAL DE LAVRAS

DIEGO VINICIUS NATIVIDADE

DVNAT:

A DEDICATED VEHICULAR NETWORK ARCHITECTURE AGAINST INCONSISTENCY AND BAD-MOUTHING ATTACKS THROUGH A REPUTATION SYSTEM

LAVRAS – MG

2020

DIEGO VINICIUS NATIVIDADE

DVNAT:
A DEDICATED VEHICULAR NETWORK ARCHITECTURE AGAINST INCONSISTENCY AND BAD-MOUTHING ATTACKS THROUGH A REPUTATION SYSTEM

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

Prof. DSc. Luiz Henrique Andrade Correia
Orientador

LAVRAS – MG
2020

Ficha catalográfica elaborada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Natividade, Diego Vinicius
DVNAT : a Dedicated Vehicular Network ArchiTecture against inconsistency and bad-mouthing attacks through a reputation system / Diego Vinicius Natividade. - 2020.
78 p. : il.

Orientador(a): Prof. DSc. Luiz Henrique Andrade Correia.

Dissertação (mestrado acadêmico)- Universidade Federal de Lavras, 2020.
Bibliografia.

1. VANET. 2. reputation system. 3. Ed25519. I. Correia, Luiz Henrique Andrade. II. Título.

DIEGO VINICIUS NATIVIDADE

DVNAT: A DEDICATED VEHICULAR NETWORK ARCHITECTURE AGAINST INCONSISTENCY AND BAD-MOUTHING ATTACKS THROUGH A REPUTATION SYSTEM

Dissertação apresentada à Universidade Federal de Lavras, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, área de concentração em Redes de Computadores e Sistemas Embarcados, para a obtenção do título de Mestre.

APROVADA em 24 de Novembro de 2020.

Prof. DSc. Raphael Winckler de Bettio	UFLA
Prof. DSc. Aldri Luiz dos Santos	UFPR
Prof. DSc. Daniel Ludovico Guidoni	UFSJ

Prof. DSc. Luiz Henrique Andrade Correia
Orientador

LAVRAS – MG
2020

I dedicate this work to my wife and my children Letícia and João.

ACKNOWLEDGMENT

First of all, I thank the Federal University of Lavras and my teacher, Ph. D Luiz Henrique, for the work's teachings and conduction, my family for the support and patience, and the friends for the incentive. In particular, I thank my Aunt Rita for funding me at the beginning of my academic life and for giving me all the support I needed for all these years. I thank and dedicate this work to CAPES for funding my master's degree with the scholarship that helped me to commit to my researches.

Special thanks to my friend Ph. D Jandinho for encouraging me to take a master's degree and to my classmates for exchanging information. I also thank the company CONNECTIVA REDES for the support before and after the master's scholarship and all the knowledge I acquired there. I could not fail to thank a person without whom today I would not be in the information technology area, much less working with computer networks. Thank you, Mateus do Nascimento, for giving me a chance and opportunity to work with you and enter this wonderful world of computing in 2002.

Finally, even during so many difficulties in 2018, I thank the Universe that always conspired in my favor.

My sincere thanks.

"Cogito, ergo sum"
(René Descartes)

ABSTRACT

The VANETs (Vehicular Ad hoc Network) brought more safety and comfort to traffic, allowing the exchange of traffic messages and entertainment content between vehicles. However, several types of attacks are known on vehicle networks, causing significant problems for drivers. Inconsistency and collusion attacks by bad-mouthing, for example, can disturb the correct functioning of the network. This paper presents DVNAT (Dedicated Vehicular Network ArchiTecture), which is capable of handling these types of attacks on vehicular networks. It uses a digital signature with the Ed25519 algorithm and a centralized reputation system with the LETICIA (Lightweight and EfficienT Information exChange In Ad-hoc network) algorithm developed to mitigate malicious vehicle attacks on the network. Simulation results show that DVNAT with the LETICIA algorithm effectively reduced the reputation of the malicious vehicle against inconsistency attacks while maintaining the reputation of the vehicle honest against bad-mouthing collusion attacks when compared to the ARS algorithms, BYOR, BYOR-LF, and IDES algorithms.

Keywords: VANET, reputation system, Ed25519.

RESUMO

As VANETs (*Vehicular Ad hoc Network*) trouxeram mais segurança e conforto para o trânsito, possibilitando a troca de mensagens de tráfego e conteúdos de entretenimento entre os veículos. Entretanto, existem vários tipos de ataques conhecidos em redes veiculares, que podem trazer grandes problemas para os motoristas. Ataques de inconsistência e conluio por *bad-mouthing*, por exemplo, são capazes de perturbar o correto funcionamento da rede. Este trabalho apresenta o DVNAT (*Dedicated Vehicular Network ArchiTecture*), que é capaz de lidar com esses tipos de ataques em redes veiculares. Ele utilizando assinatura digital com o algoritmo Ed25519 e um sistema de reputação centralizado com o algoritmo LETICIA (*Lightweight and EfficienT Information exChange In Ad-hoc network*), que foi desenvolvido para mitigar ataques de veículos maliciosos em VANET. Resultados de simulação mostram que o DVNAT com o algoritmo LETICIA reduziu efetivamente a reputação do veículo malicioso contra ataques de inconsistência, ao mesmo tempo que manteve a reputação do veículo honesto contra ataques de conluio por *bad-mouthing*, quando comparado aos algoritmos ARS, BYOR, BYOR-LF e IDES.

Palavras-chave: VANET, sistema de reputação, Ed25519.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS

4G Fourth Generation of mobile network

5G Fifth Generation of mobile network

API Application Programming Interface

ARS Anonymous Reputation System for Vehicular Ad hoc Networks

BYOR Bring Your Own Reputation

BYOR-LF Bring Your Own Reputation with Logevity Factor

CA Certification Authority

CRL Certificate Revocation List

DB Data Base server

DBMS Data Base Management System

DoS Denial of Service

DDoS Distributed Denial of Service

DSRC Dedicated Short-Range Communications

DVNAT Dedicated Vehicular Network ArchiTecture

ECC Elliptical Curve Cryptography

ECDLP Elliptical Curve Discrete Logarithm Problem

ECDSA Elliptic Curve Digital Signature Algorithm

Ed25519 Edwards curve 25519

EDCA Enhanced Distributed Channel Access

EMAP Expedite Message Authentication Protocol

EPD EtherType Protocol Description

FCC Feedback Computation Center

GPS Global Positioning System

HMAC Hash Message Authentication Code

HMM Hidden Markov Model

HTMF Hybrid Trust Management Framework

IDE Integrated Development Environment

IDES Instant Data Evaluation Scheme

IoT Internet of Things

IEEE Institute of Electrical and Electronics Engineers

IPv6 Internet Protocol Version 6

JSON JavaScript Object Notation

LETICIA Lightweight and EfficienT Information exChange In Ad-hoc network

LLC Logic Link Control

LTC Long-term Certificate

MAC layer Medium Access Control Layer

MAC Message Authentication Code

MITM Man-In-The-Middle

OBU On-board Unit

P2P Peer-to-Peer

RCRL RSU Certificate Revocation List

RGTE Reputation-based Global Trust Establishment

RH Request Handler

RSA Rivest, Shamir and Adleman

RSU Road Side Unit

STC Short-term Certificate

SUMO Simulation of Urban Mobility

TCP Transmission Control Protocol

TCP/IP Transmission Control Protocol/Internet Protocol

TESLA Timed Efficient Stream Loss-tolerant Authentication

WAVE Wireless Access in Vehicular Environments

WSA WAVE Service Advertisement

WSMP WAVE Short Message Protocol

V2I Vehicle to Infrastructure

V2V Vehicle to Vehicle

VANET Vehicular Ad Hoc Network

VARS Vehicle Ad-Hoc Network Reputation System

VCRL Vehicle Certificate Revocation List

VEINS Vehicles in Network Simulation

SUMMARY

1 INTRODUCTION

Due to the miniaturization of the embedded devices, the vehicles incorporated in their on-board systems equipment can create a communication network. This network was named Vehicular Ad Hoc Network (VANET), which deals specifically with vehicular communication, where standards and protocols were also developed for this purpose. In VANET, vehicles behave like the nodes of a network, communicating directly with each other or with intermediate infrastructure devices, called Road Side Unit (RSU). The VANETs bring more comfort and especially safety for drivers. Several applications go beyond the use of VANET, from communication and accident registration, information about the state of the road, points of interest, and even entertainment services, such as sharing multimedia files.

In some applications, the information exchanged between nodes is critical, such as road safety information, for example. It is estimated that every year, 1.35 million people die in traffic accidents, (World Heath Organization, 2020). Many solutions for communication in VANETs have been proposed in recent years to improve traffic safety, but just implementing a system for exchanging messages in traffic does not make traffic safer. It shows the need to create more and attack-tolerant vehicle networks.

The sending of false information in critical scenarios can cause accidents, divert the driver from his route, create false traffic situations, and even qualify a vehicle from the network unfairly or imprecisely. The nodes of a network must have confidence in the information exchanged with their neighbors, as disseminating false messages between the nodes of a network or an application can have disastrous consequences (PEDROSO et al., 2019; Su et al., 2020). One way to contain the dissemination of this message type is to use a network architecture that contemplates: (a) an efficient vehicle reputation system, capable of following the behavior of the vehicle; (b) a digital signature algorithm that is fast and compatible with vehicular networks' reality to ensure that messages sent are not repudiated.

A robust reputation system for vehicle networks must be able to judge the vehicles involved in the communication and react to possible attacks, reducing malicious vehicles' reputation to mitigate their effects. Critical applications in VANET must vigorously punish malicious vehicles and considerably decrease their reputation. Some systems found in the current literature smoothly reduce the malicious vehicle's reputation and still allow its rapid recovery. Others reduce it drastically, not allowing it to resume if it returns to behaving ethically. Many of these works fail because they do not present simulations or experiments, do not evaluate the network under malicious attacks, or evaluate these systems with a reduced number of vehicles or even from the perspective of just one type of attack.

Some attacks against vehicular networks are very ordinary, such as the inconsistency attack when a vehicle remains unstable by sending false and true messages (ZHANG, 2011), and bad-mouthing attack, when a malicious vehicle opines negatively on the messages received, in order to harm another vehicle (BANKOVIĆ et al., 2011; WANG et al., 2016). This attack can also be enhanced when multiple vehicles come together to attack a specific vehicle.

1.1 Motivation

The implementation of vehicular communication allows bringing comfort and tranquility to the drivers. Some safety problems arise, such as false messages, outdated or out of context, electronic rumors, and SPAM. It can lead the driver to make incorrect decisions or be attacked by malicious people. The correct implementation of a secure communication method, but with an acceptable overhead, is a challenge because, in VANET, time constraints for message exchange and high mobility between vehicles are restrictions that must be taken into account (BAO et al., 2017).

Malicious vehicles can poison the network with fake messages, alternating their behavior, sending real and fake messages to confuse neighbors. Several vehicles can also join together to make a group attack against a specific node's reputation to gain some advantage. Therefore, the guarantee of receiving a valid and reliable message can be achieved by observing the reputation of the nodes that send the messages. It gives drivers all the benefits that applications in VANET can offer.

High-priority false messages circulating on the network, such as accident information and mandatory detours, can be as dangerous as a real accident or cause significant disruption to the correct traffic flow. For example, a misleading message about an accident stating an alternative route for vehicles can cause drivers to be directed to even more dangerous locations and give rise to new types of thefts, kidnappings, among other problems. This work's accomplishment contributes to improving security in VANETs and brings users more utility to the network, reducing the number of false positives.

1.2 Problem definition

There are many reputation systems and architectures proposed in the literature to combat attacks on vehicular networks. The problem is the lack of a system that can handle and mitigate both inconsistency attacks and bad-mouthing collusion attacks, using a single algorithm capable of finding a balance when reputing vehicles according to their behavior on the network.

1.3 Goals

This work's main objective is to create a vehicular network architecture capable of guaranteeing confidence in the messages received and safety for the driver when making decisions. For this, the proposed architecture makes use of (a) a digital signature algorithm that is fast and compatible with the reality of a VANET; (b) a lightweight and efficient algorithm against inconsistency and collusion by bad-mouthing attacks; (c) an architecture to simulate reputation in VANET.

1.3.1 Specific goals

In addition to the proposed goals, the specific objectives include:

- to prove the privacy of the vehicle on the network, using a short-term certificate to sign the messages;
- to ensure the authenticity and non-repudiate of the messages, through the use of digital signature with Ed25519 Edwards curve 25519 (Ed25519) digital signature algorithm. It is a lightweight algorithm and has been widely used in several computational systems (IANIX, 2020);
- to implement a centralized reputation system that contains an algorithm capable of following the behavior of vehicles and reputing them fairly. This system must be resistant to attacks of inconsistency and bad-mouthing collusion;
- to create an architecture containing dedicated entities and services at the core of its infrastructure, to control the entire reputation system and distribution of certificates;
- to increase VANET reliability to reduce the incorrect decisions by drivers due to the traffic of false messages on the network;

1.4 Contributions

The main contributions of this work were the development of Dedicated Vehicular Network ArchiTecture (DVNAT) and Lightweight and EfficienT Information exChange In Ad-hoc network (LETICIA). DVNAT is a complete and functional approach to the vehicular network, composed of dedicated services, taking into account aspects of the real world through the Veins framework (SOMMER; GERMAN; DRESSLER, 2011), widely used in VANET simulations. This architecture uses a digital signature to guarantee the authenticity and non-repudiation of the messages in VANET. The second, LETICIA, is a lightweight reputation algorithm that uses simple calculations to maintain the vehicle's reputation against inconsistency and bad-mouthing collusion attacks.

LETICIA is a lightweight reputation algorithm that uses simple calculations to maintain the vehicle's reputation against inconsistency and bad-mouthing collusion attacks.

The contributions of this work also resulted in the following publications:

- NATIVIDADE, D.; CORREIA, L.; SANTOS, A. Um algoritmo de reputação centralizado para redes veiculares contra ataques de inconsistência e bad-mouthing. In: SBSeg - Simpósio Brasileiro de Segurança da Informação e Sistemas Computacionais. 2020.
- NATIVIDADE, D. V.; CORREIA, L. H. A. Avaliação de algoritmos de assinatura digital em redes veiculares utilizando ambiente emulado. In: Workshop de Gerência e Operação de Redes e Serviços. 2020.

1.5 Work organization

This work is organized as follows: Chapter 2 shows the background for understanding this work, as VANET concepts, protocols, and kinds of attacks. Some related work is presented and compared in Chapter 3. The proposed architecture and the used reputation algorithm is presents in Chapter 4 and Chapter 5, respectively. The methodology used and the metrics and some preliminary tests are shown in Chapter 6. Chapter 7 presents the test results of this work. The Chapter 8 concludes the work with the final considerations.

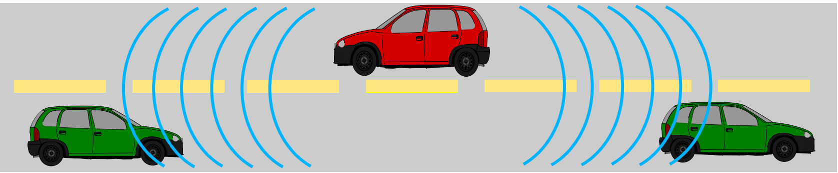
2 BACKGROUND

This chapter presents the background for understanding this work, giving a brief description of VANET, the types of authentication protocols used, the Institute of Electrical and Electronics Engineers (IEEE) 1609 standard, kinds of attacks in vehicular networks, and some cryptography and authentication protocols.

2.1 VANET

VANET is the term used to refer to vehicular networks. *A priori* VANET refers to how is established the communication between the network vehicles: from Vehicle to Vehicle (V2V). However, since there is an exchange of messages between vehicles and the network infrastructure (RSU) — Vehicle to Infrastructure (V2I) communication, the term VANET is controversial among researchers, (HARTENSTEIN; LABERTEAUX, 2008). The Figure 2.1 illustrates V2V communication, where the vehicles send and receive messages directly from each other. Figure 2.2 shows a V2I communication, in which vehicles communicate with the infrastructure through RSU.

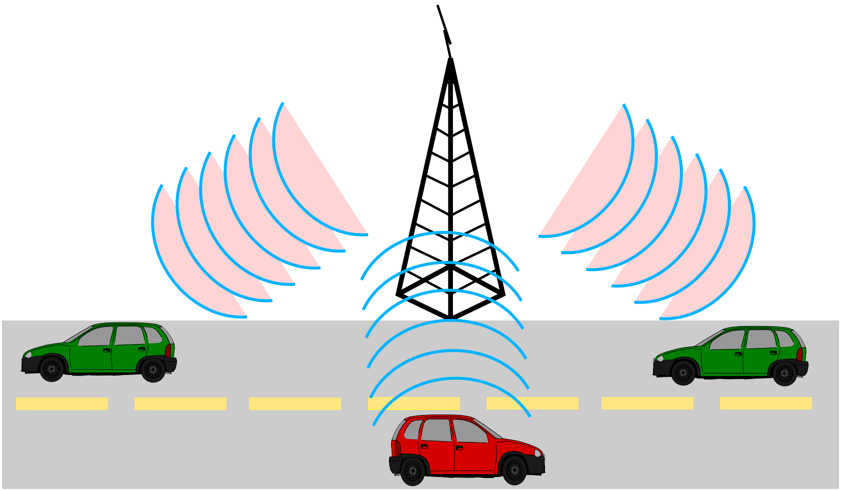
Figure 2.1 – V2V communication



Source: author's own (2020)

A vehicle network's primary purpose is to bring safety to the driver by sharing relevant information about road conditions and traffic (Brendha; Prakash, 2017). Nevertheless, it can also exchange entertainment information, such as multimedia files, through an Internet connection (MISHRA; SINGH; KUMAR, 2016). Unlike other network topologies, in a VANET, nodes move at high speeds (in equal or opposite directions) and deal with broadcast messages from a large number of nodes. For a network of this length to work correctly, a standard was designed to support communication between vehicles: IEEE 1609.

Figure 2.2 – V2I communication



Source: author's own (2020)

2.2 IEEE 1609 Standard

In 2006, the IEEE created the first version of a network communication standard called Wireless Access in Vehicular Environments (WAVE), which was based on the IEEE 802.11p and IEEE 1609 standards, for physical layer modeling and definition of architecture and security services, respectively (AHMED; ARIFFIN; FISAL, 2013). The IEEE 1609 standard has several subdivisions, where each defines a specific layer of the network. The same has undergone updates in 2010, 2013, 2016, and 2017.

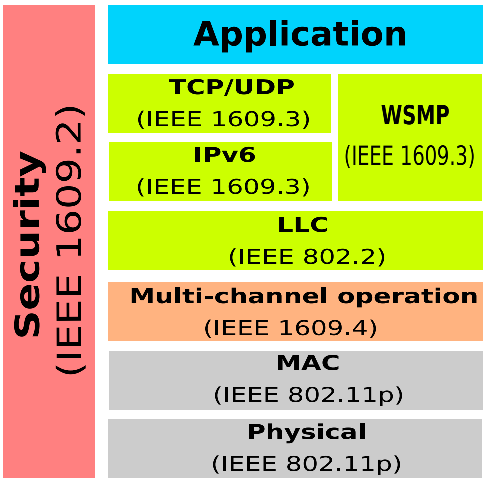
The physical layer operates on the 5GHz band and has seven channels of 10MHz each, being service, communication, and security channels (EICHLER, 2007). The IEEE 1609 standard describes the Medium Access Control Layer (MAC layer) and its access mechanisms and the network layer, with Internet Protocol Version 6 (IPv6) and WAVE Short Message Protocol (WSMP) addressing, for sending high-priority and common messages respectively (AHMED; ARIFFIN; FISAL, 2013).

Application security services and message management services are defined in the IEEE 1609.2 standards, which was last revised in 2017 on IEEE 1609.2.a (IEEE1609.2a, 2017). This model uses a cross-layered approach (implemented in multiple layers). The Figure 2.3 shows the main norms and protocols that make up the WAVE standard. These are explained below.

IEEE 1609.4-2016

Specifies extensions of the IEEE 802.11 layer (MAC layer) such as multichannel Operations, channel coordination, and routing, multichannel synchronization, Enhanced

Figure 2.3 – WAVE: layers, standards and protocols



Source: author's own (2020)

Distributed Channel Access (EDCA), use of the IEEE 802.11 Timing Announcement table, Message Authentication Code (MAC) layer readdressing with pseudonym support (IEEE1609.0, 2017).

IEEE 1609.3-2016

This standard establishes the network layer services and the following functions: WAVE Service Advertisement (WSA) monitoring and transmission, channel access assignment, use of the Logic Link Control (LLC) sub-layer, and EtherType Protocol Description (EPD) and simplified IPv6 configuration (IEEE1609.0, 2017).

802.11p

It is an OFDM-based standard that establishes the physical and MAC layer. It is a standard 802.11 patch that establishes communication between vehicles in a frequency range of 5.9GHz (5,850-5,925 GHz) with a range of 10 MHz (GRäFLING; MäHöNEN; RIIHIJäRVI, 2010). It uses Dedicated Short-Range Communications (DSRC) and allows a distance of up to 1Km. The application layer is proprietary and can be deployed as needed. The security layer is presented as follows.

2.2.1 IEEE 1609.2/IEEE 1609.2.a - Security services

According to IEEE1609.2a (2017), initially, in order to establish security standards WAVE, the IEEE P1556 standard was created, which was later renamed to IEEE Std 1609.2.

This standard defines message formats and structures, including protection and management methods for services using this protocol. This model aims to offer a minimum of security in communication, protecting the exchanged messages with low overhead, both in the transmission and processing (IEEE1609.2a, 2017).

However, this standard only addresses the protection of messages against privacy and integrity attacks, not specifying methods to protect against false messages or contain and identify vehicles with bad behavior on the network. The current literature describes many attacks on vehicular networks.

2.3 The attacks in VANET

The malfunction of sensors in a vehicle or even a malicious driver can impair a vehicle network's proper functioning through several techniques known. Many of these threats were inherited from wireless sensor networks. The primary known forms of attacks can affect: the availability of the network, the integrity and confidentiality of data, the authenticity, and reputation of the vehicles. Among the various types of attacks at VANET are the following:

- **Denial of Service (DoS)/Distributed Denial of Service (DDoS)**: attacks against availability; a node (DoS) or several nodes (DDoS) that trigger many messages against the On-board Unit (OBU) or RSU to deplete its resources (ROSELINMARY; MAHESHWARI; THAMARAISELVAN, 2013);

- **Jamming**: attacks against availability, an intruder interferes in the electromagnetic spectrum by increasing latency or even blocking communication (WANG et al., 2019);

- **Blackhole/Greyhole**: attacks against availability; a malicious node do not relay the received packets (blackhole) or retransmit only those of its interest (Greyhole) (ALNASSER; SUN; JIANG, 2019);

- **Impersonation**: attacks against authenticity; an attacker goes through another vehicle or entity in order to deceive other nodes of the network (MOKHTAR; AZAB, 2015);

- **Eavesdropping**: attack against confidentiality; an attacker monitors another vehicle for insider information (HASROUNY et al., 2017);

- **Global Positioning System (GPS) Spoofing**: attacks against integrity; an attacker changes routes by manipulating GPS signal within his coverage area to send a vehicle the wrong way, for instance, (BITTL et al., 2015);

- **Bogus Information**: attacks against integrity; an attacker sends false messages to compromise the network and change the behavior of drivers on the road (HASROUNY et al., 2017);

- **Man-In-The-Middle (MITM)**: attacks against integrity and confidentiality; an attacker remains in the middle of the communication path, intercepts the messages sent by a vehicle, modifies and forwards them with the violated content (AHMAD et al., 2018);

- **Sybil**: attacks against authenticity and reputation; an adversary forges the identity of several others at the same time, creating the illusion that there are several vehicles in the network, thus confusing the other vehicles in the network (ROSELINMARY; MAHESHWARI; THAMARAISELVAN, 2013);

- **Newcomer**: attacks against authenticity and reputation; an attacker enters in the network with an identity, and after a while, it leaves the network and enters again with another identity to restart its reputation (TRČEK, 2017);

- **Betrayal**: attacks against integrity; an opponent of the network suddenly changes its behavior by sending fake messages, becoming a malicious node (ZHANG, 2011);

- **Inconsistency**: attacks against integrity; an attacker behaves in an unstable way, informing true and false messages alternately, thus compromising the correct functioning of the network (ZHANG, 2011). In this attack, unlike Betrayal, the vehicles do not maintain a constant attack but behave honestly or dishonestly arbitrarily;

- **Collusion**: attacks against integrity and reputation; several vehicles in the network can ally to achieve a common goal, such as incorrectly reporting a vehicle's reputation, for example, (ZHANG, 2011).

- **Bad-mouthing**: attacks against reputation; malicious users can arbitrarily give positive or negative feedback about a vehicle on the network, in order to change their reputation so that other vehicles can make incorrect decisions (BANKOVIĆ et al., 2011). Various malicious vehicles can conspire against a target to reduce its reputation. This attack usually happens between competing nodes that provide similar services. In this case, this attack is used by one or more vehicles, to slander your competitor (WANG et al., 2016).

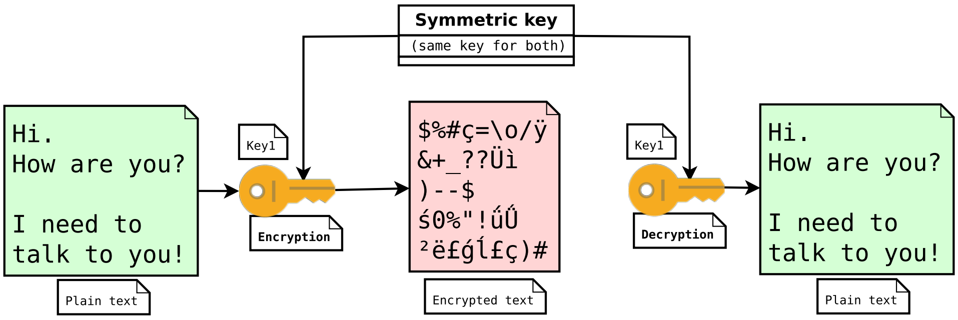
2.4 Cryptography and digital signature

This section presents the symmetric and asymmetric cryptography types and the digital signature processes, with some examples of algorithms.

2.4.1 Symmetric cryptography

Symmetric encryption, private key cryptography, or secret key cryptography are synonyms of an encryption scheme in which everyone involved in secure communication receives the same key. It is a widely used scheme in vehicular networks for being lightweight. This key is used for both encryption and decryption, and everyone involved in the communication must have a copy of it (AYUSHI, 2010). The Figure 2.4 illustrates the symmetric cryptography process, where a same key (`Key1`) is used for encryption and decryption of messages.

Figure 2.4 – Symmetric encryption process



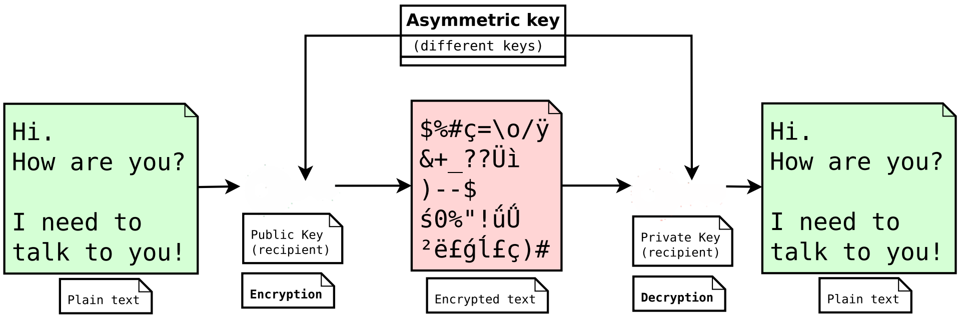
Source: author's own (2020)

For message authentication, a widely discussed algorithm is MAC. MAC is a tag generated from a message and a secret key to authenticate a message (RSA LABORATORIES, 1993). A mathematical function receives the message and the secret key to generate the tag. The key is shared between the sender and receiver of the message to verify the authenticity of the same. An implementation of MAC is Timed Efficient Stream Loss-tolerant Authentication (TESLA). It provides a message's integrity and authenticity by delaying key disclosure. This algorithm also contributes to low processing overhead and low network communication (PERRIG et al., 2005).

2.4.2 Asymmetric cryptography

Asymmetric or public key cryptography is a framework widely used in encryption and digital signature processes, where a pair of mathematically related keys are used: public and private key. While the public key is distributed to all, the private key must belong exclusively to the owner. The public key of asymmetric cryptography should not be confused with the secret key of symmetric cryptography. In asymmetric cryptography, the private key must always be kept safe (AYUSHI, 2010). Figure 2.5 shows the asymmetric cryptography procedure, where the recipient's key pair is used in the process. The sender uses the recipient's public key to encrypt the message, and the recipient uses its private key to decrypts the same message.

Figure 2.5 – Asymmetric encryption process



Source: author's own (2020)

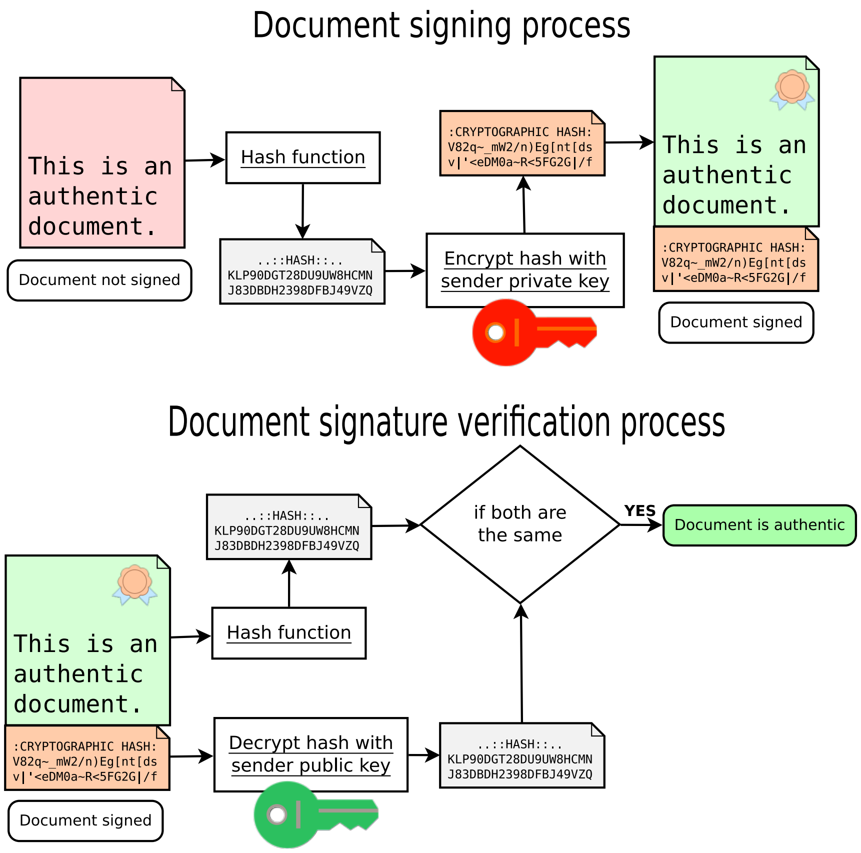
2.4.3 Digital signature

Digital signature schemes provide validation of a document when it is signed, giving the same legal validity as a handwritten document. Public key digital signature schemes (scope of this work) should be verifiable, using the private key to sign and the public key to verify the validity. A trusted third party is used to solve some problems in the case of an attempt of forgery of keys (JOHNSON; MENEZES; VANSTONE, 2001). This third party is responsible for issuing the certificates with the public and private keys of all those involved in the communication.

The Figure 2.6 illustrates the digital sign procedure, where the sender's key pair is used in the process. The sender generates a hash of the message and encrypts that hash with its private key. The receiver uses the sender's public key to verify the digital signature as follows: decrypts the received hash; generates a new hash of the message (using the same hash generation algorithm used in the sender); finally, it compares the generated hash with the decrypted hash, if both are equal, the signature is considered valid.

In follow are presented some of the main asymmetric cryptographic algorithms: Rivest, Shamir and Adleman (RSA), Elliptic Curve Digital Signature Algorithm (ECDSA) and Ed25519.

Figure 2.6 – Digital signature process



Source: author’s own (2020)

RSA

The public key cryptography algorithm called RSA was created in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman, that offer encryption and digital signatures (RSA LABORATORIES, 2000). The size of the keys used is generally 1024 or 2048 bits, and its security is based on the product factorization of two large prime numbers, (PERBAWA; AFRYANSYAH; SARI, 2017). To encrypt a message, raise it to a power e power (the message must be converted to a numeric value), which is publicly known (public key). The result is divided by the product of two large prime secret numbers, p and q. To decipher the message, raised it to a secret power d (private key), and the result is also divided by the product of the two large secret primes, p and q. The security of the system lies in the difficulty of factoring large numbers (RIVEST; SHAMIR; ADLEMAN, 1978).

ECDSA

ECDSA is an algorithm widely used in digital signature processes, created in 2000. It is characterized by being lighter than other algorithms of the same category, offering the same security, with smaller keys (PERBAWA; AFRYANSYAH; SARI, 2017). Its security is based on the Elliptical Curve Discrete Logarithm Problem (ECDLP). This makes the force per key bit significantly larger in an algorithm that uses elliptic curves, (JOHNSON; MENEZES; VANSTONE, 2001).

Ed25519

Ed25519 is a relatively new and lighter cryptography algorithm than RSA and ECDSA. Ed25519 has this name because it is based on the Edwards-curves and reduces the field q to $2^{255} - 19$. The algorithm is efficient in short messages due to the way it was constructed, (BERNSTEIN et al., 2012). The Ed25519 has fast execution of high-security elliptic curve encryption and can be used for high-performance applications and low-computing hardware (TURAN; VERBAUWHEDE, 2019).

An encryption system's security is proportional to the relative complexity of the mathematical problem it addresses (KALRA; SOOD, 2011). The security provided by a 160-bit Elliptical Curve Cryptography (ECC) cryptographic key is equivalent to a 1024-bit key of the RSA algorithm (MANVI; KAKKASAGERI; ADIGA, 2009). According to Kumar (2006), Bernstein et al. (2012), a 256-bit ECC key is equivalent to an approximately 3072-bit RSA key and a 128-bit security level. Table X presents a table comparing the three algorithms presented and their key sizes for a security level of 128 bits.

Table 2.1 – Key size of algorithms

Algorithm	Key size
RSA	3072-bit
ECDSA	256-bit
Ed25519	256-bit

Source: author's own (2020)

2.5 Reputation systems

Reputation systems aim to establish trust and give credibility to information exchanged between nodes in the network. According to Engoulou et al. (2019), reputation refers to observations of an entity's past behavior, which may indicate its future behavior. Entities that have a history of good behavior tend to remain with good behavior. The same can be said for individuals with bad behavior. This same concept, can be applied to vehicular networks. For example, so that a vehicle decides whether or not to trust the information sent by an unknown vehicle, its reputation can be used as a behavior parameter. For network nodes to exchange messages with their neighbors, they must have confidence in the information exchanged. Reputation systems aim to establish trust and give credibility to information exchanged between nodes in the network.

These systems can be: (a) centralized, when there is a central entity that computes and controls the participants' reputation; (b) distributed, when the participants themselves store and distribute their opinions about others, without a controlling entity (JøSANG; ISMAIL; BOYD, 2007). However, for Su et al. (2020), these systems can still be: (a) centered on the entity when reputations reflect the behavior of the entities that send messages on the network; or (b) message-centered, when forwarded messages carry the reputation independent of the entity that created them. A specific entity's reputation is given by other entities' opinions or feedback with the previous contact. This feedback is subsequently aggregated, and a reputation value is calculated for each entity. This value can be expressed in binary, in intervals (0, 1 or -1, 1), by a positive integer, or even in a textual form such as bad/regular/good/excellent (Ruohomaa; Kutvonen; Koutrouli, 2007). Therefore, for network nodes to exchange messages with their neighbors, they must have confidence in the information exchanged, and this can be achieved through a reputation system.

3 RELATED WORKS

This session analyzes some related works to security in the information exchanged and the received messages' reliability. This covered works that deal only with authentication schemes to jobs related to Fifth Generation of mobile network (5G), reputation, trust in VANET, and Blockchain.

Authentication in VANET

The authors Manvi, Kakkasageri e Adiga (2009) proposed a method of message authentication in vehicular networks using the ECDSA algorithm. Each vehicle has a pair of keys (public and private). This type of encryption is called public (or asymmetric) key cryptography, where it is possible to guarantee the authenticity and non-repudiation of the messages transmitted on the network. However, this method alone does not guarantee the message's reliability, but it is possible to identify the vehicle. The ECDSA algorithm is based on the ECC that has a reduced overhead compared to RSA.

In Wasef e Shen (2013), the Expedite Message Authentication Protocol (EMAP) was suggested for VANET, focusing on the process of checking the Certificate Revocation List (CRL) through a more efficient verification process. To comply with the proposal, EMAP uses the Hash Message Authentication Code (HMAC) algorithm for message authentication and a probabilistic key allocation system for the distribution of keys between vehicles. In the tests performed by the authors, the protocol proved to be safe and efficient, significantly reducing the message loss rate compared to other models. But this way, a large number of keys needs to be created and distributed among the vehicle, increase the complexity of the keys management and also do not ensure the reliability of the message.

Another scheme of message authentication using ECDSA was shown in Ravi e Kulkarni (2013). The authors used a Peer-to-Peer (P2P) network in which vehicles organize themselves in a highly dynamic way and development a framework to achieves high scalability, security, and speed in communication. According to the authors, the tests were performed in a real environment, with different message sizes, but used only two vehicles.

Due to VANET connectivity constraints and to reduce the overhead inserted by ECDSA, Sakhreliya e Pandya (2014) authors invoked a hybrid system that uses public key infrastructure with symmetric cryptography. Thus, while a pure asymmetric encryption mechanism has a time of two milliseconds for message generation and five milliseconds for verifying the same, using the proposed system, it is reduced to 26 microseconds each of the previous operations (SAKHRELIYA; PANDYA, 2014).

In Wang e Yao (2017), the authors presented a two-step authentication model: authentication with asymmetric encryption through a long-term digital certificate and symmetric cryptography with master key exchanged with the RSU. This paper also addresses the issue of referring to Vehicle Certificate Revocation List (VCRL) and RSU Certificate Revocation List (RCRL). According to the authors, the simulations performed achieve the proposed objective, although it creates an overhead for checking revoked certificates.

In Singh et al. (2015), is approached the authentication and preservation of privacy in the exchange of messages between vehicles. It also addresses the use of a leading cluster, which receives messages from vehicles and only communicates with RSU to reduce message traffic. The IEEE 802.11p standard and ECC encryption are used for communication. This paper also does not address false messages or VANET attacks.

Reputation systems in VANET

Dotzer, Fischer e Magiera (2005) have created the Vehicle Ad-Hoc Network Reputation System (VARS) that uses direct and indirect opinions on the messages sent. Feedback on the messages' reliability is attached by vehicles during forwarding, and the sender's reputation interferes with these opinions. When the vehicle enters the decision area, it evaluates the opinions and decides whether or not to accept the message. According to the authors, the system can handle sophisticated attacks and has proved to be efficient in general. However, according to the authors, this model may be susceptible to collusion attacks, and practical tests have not been carried out.

Li et al. (2013) proposed Reputation-based Global Trust Establishment (RGTE) scheme, where the network nodes inform the reputation management center about the trust they have over the other nodes in the network, and this communication occurs through RSUs. The server stores a table with the reputation of one node over another one. Reputations in a node have time to live, and nodes with good behavior slowly increase their reputation, while a bad reputation decreases quickly. The nodes do not receive the recommendations of trust directly from the other nodes but through the central reputation. However, the authors did not conduct experiments to test the proposed mechanism.

The Hidden Markov Model (HMM) for reputation computing in VANET was proposed by Shrivastava, Sharma e Chaurasia (2016) as a lightweight reputation calculation mechanism. The proposed scoring system assesses the messages received' reliability and legitimacy, taking into account the time restrictions in a vehicular network. However, reputation calculations were tested in the Matlab[1] mathematical calculation system. No tests in real or simulated scenarios were performed to prove their efficiency, taking into account the vehicle mobility.

In Kchaou, Abassi e Guemara (2018), the authors propose a scheme where vehicles are grouped into clusters, and messages are stored using BlockChain. A cluster head is responsible for maintaining a table with cluster members' reputation, and the mining node decides the message validation using fuzzy logic. After validation, a block is constructed containing several messages validated by the miners. The authors did not perform practical tests. The time for the validation of the safety messages seems very high, making it impossible to use them in accident situations, where the decision must be made quickly.

3.1 VANET reputation algorithms

In this section, presents the reputation algorithms in vehicular networks that we use to compare this work. Some are not about reputation algorithms but complete frameworks that use their reputation algorithm. In comparison, this work use only the reputation algorithms previously presented and not their complete frameworks. So, for simplicity, we named the algorithm its framework. All the algorithms presented here were adapted to work in the architecture proposed in this work.

3.1.1 ARS

The authors Jaimes, Ullah e Moreira (2016) proposed the Anonymous Reputation System for Vehicular Ad hoc Networks (ARS). It uses a reputation system in which a centrally-reputed server evaluates the vehicles that generate the messages and those that forward them to the destination. Vehicle privacy is achieved through the use of short-term certificates as pseudonyms. The messages are attached to the pseudonym of vehicles that create them and forward them.

In the reputation system used in ARS, vehicles have a reputation ρ between $]-1,1[$. Two counters are used to store the feedback received from other vehicles about the messages

[1] https://www.mathworks.com/products/matlab.html

sent, in which F^+ stores the positive feedback and F^- the negative feedback. Positive feedback are worth +1, and negative feedback are worth -1. The reputation calculation is performed according to Equation 3.1. The α factor is used to give different weights between the old reputation ρ_0 and the aggregated feedback. Finally, a normalization function was applied to leave the reputation between $]0,1[$, as shows the Equation 3.2.

$$\rho = \rho_0 * (1-\alpha) + (F^+ + F^-) * \alpha \tag{3.1}$$

$$\rho = Norm(\rho) \tag{3.2}$$

3.1.2 BYOR and BYOR-LF

In (MüHLBAUER; KLEINSCHMIDT, 2018), the authors proposed Bring Your Own Reputation (BYOR), a reputation system for vehicular networks. At BYOR, vehicles receive their digitally signed reputation when they are within the infrastructure coverage area (RSU). The system operates in a partially decentralized manner, where contact with the RSU does not need to be permanent to operate but only sporadic. RSU also addresses vehicle privacy through short-term digital certificates, in which all messages sent are digitally signed. For reputation calculation, the authors used a simple summation algorithm of the feedback received and a Bayesian inference algorithm. However, the authors evaluated their algorithms in simulations with a limited number of vehicles. Although they stated that the model is robust against bad-mouthing attacks, they did not evaluate this type of attack.

In BYOR, the authors used some reputation algorithms to test the proposed model. The tests in this work use the Bayesian inference reputation algorithm, using the beta probability distribution function. A value gives the vehicles' reputation between $]0,1[$ and two counters are used to store the feedback received, in which α stores the positive feedback and β the negative feedback, as shown in Equation 3.3. The term ρ_{fb_i} refers to the vehicle's reputation that sent the feedback, and n is the number of vehicles that sent feedback.

$$\begin{cases} \alpha = \sum_{i=1}^{n} \rho_{fb_i} \\ \beta = \sum_{i=1}^{n} \rho_{fb_i} \end{cases} \tag{3.3}$$

All feedback remains stored and is used during reputation calculations. In each reputation calculation, they have aggregated all feedback, as shown by Equation 3.4. Finally, it is calculated the new reputation ρ by averaging the old reputation ρ_0 and aggregating feedback, as shown by Equation 3.5.

$$F = \frac{\alpha}{(\alpha + \beta)} \tag{3.4}$$

$$\frac{\rho = \rho_0 + F}{2} \tag{3.5}$$

The authors made a variation of BYOR using a longevity factor. This factor defines that only the last `f` feedback will be stored for reputation calculation. In this work, we call this variation of Bring Your Own Reputation with Logevity Factor (BYOR-LF). It considers a longevity factor (LF), which functions as a sliding window, allowing feedback received after a certain point, are "forgotten". The main difference to BYOR is that only the f last feedback received is taken into account.

3.1.3 IDES

In the paper (Su et al., 2020), the authors presented a reputation management scheme for identifying malicious vehicles in VANET. The proposal of Instant Data Evaluation Scheme (IDES) is to collect the global reputation of vehicles and enable instant recognition of unreliable messages. The authors assume that a high-performance 5G network provides end-to-end communication between vehicles. The reputation system is centralized, and the vehicle's reputation that generated the data contributes to the confidence in the messages received. The receiving vehicle validates the received data and contributes to updating the sending vehicle's reputation. A self-developed emulator evaluated the behavior of IDES in the presence of malicious attacks in the ways of spreading false messages (Bogus and Secret) and collusion. IDES was compared only to the Hybrid Trust Management Framework (HTMF) framework created for vehicular social networks (HUSSAIN et al., 2016).

In IDES, the vehicles' reputation increases linearly and falls exponentially, depending on the vehicle's behavior on the network. The reputation algorithm used in this work has no upper and lower limits. In theory, the reputation could range from $]-\infty, +\infty[$. To adapt to our infrastructure and be able to compare with the other algorithms, the reputation was normalized so that it reached values only between $]0, 1[$.

The IDES calculates the reputation, taking into account the number of positive and negative feedback received and the vehicle's reputation that sent the feedback. A feedback counter F_m is used for each message sent, as shown by Equation 3.6. In this equation, feedback is the sum of the reputations of the vehicles that sent the message, ρ_{fb_i}. Positive feedback has a positive sign, and negative feedback has a negative one. The infrastructure evaluates each message's feedback to calculate the new reputation, whether positive or negative. If the result is negative, the message is assumed to be false. Otherwise, the message is supposed to be true.

$$F_m = \sum_{i=1}^{n} \rho_{fb_i} \tag{3.6}$$

For each identified true message, the reputation is increased, as shown by Equation 3.7, in which c is a constant with values between $]0,1[$, which serves to amortize the increase in reputation. ρ_0 is the old reputation. When it is identified that the vehicle has sent false messages, Equation 3.8 is used to reduce its reputation. n is the number of false messages identified, ρ_0 is the old reputation, and F_m is the feedback with a negative sign.

$$\rho = \rho_0 + c * F_m \tag{3.7}$$

$$\rho = \rho_0 + 2^n * F_m \tag{3.8}$$

As mentioned, a normalization function is used for both cases so that the values are between $]0,1[$, as shown in Equation 3.9. IDES had good results against inconsistency attacks. However, the authors did not consider bad-mouthing attacks on the network.

$$\rho = Norm(\rho) \tag{3.9}$$

3.2 Related works comparison

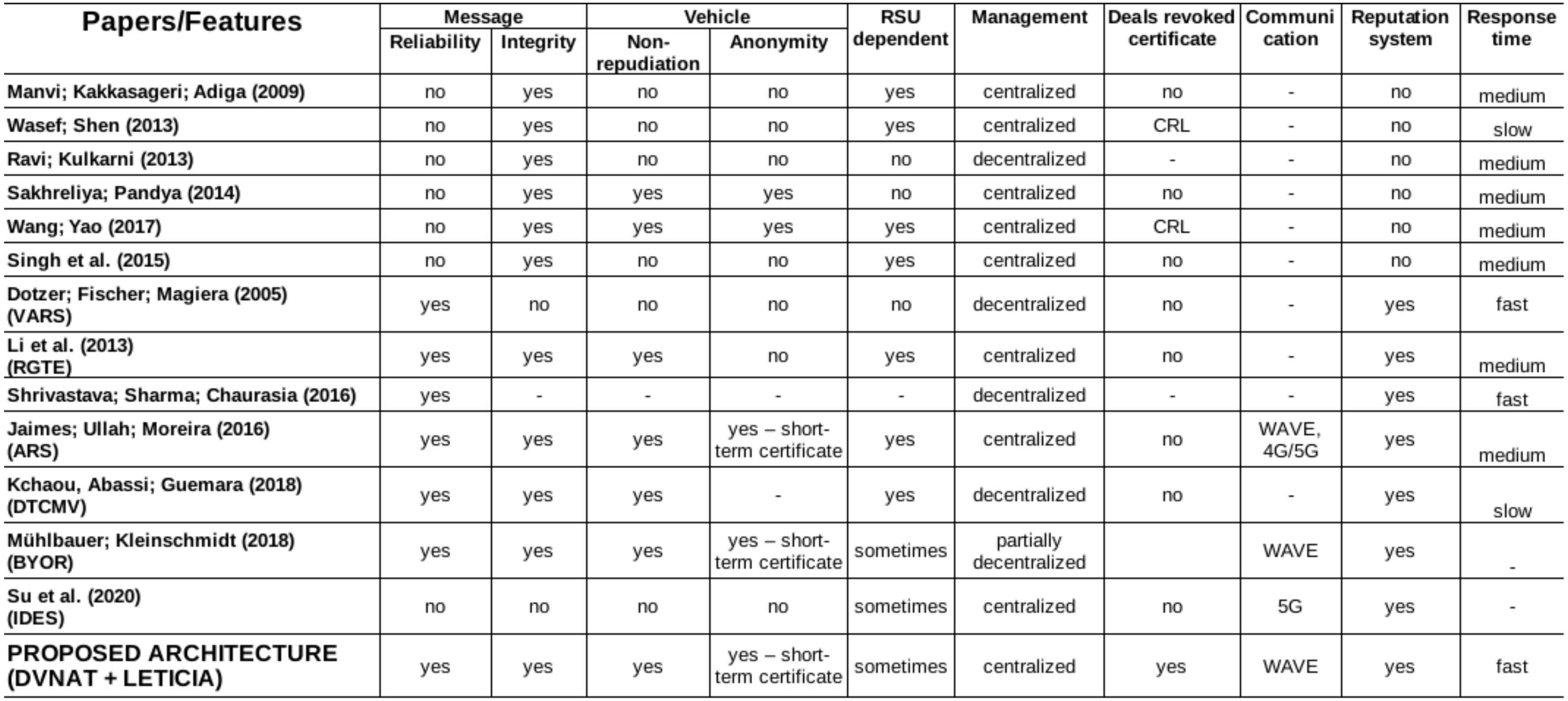
Following, Table 3.1 shows the comparison between the related works mentioned above and the proposed architecture, containing its main features. In short, about the reliability of the messages, the non-repudiation, and the anonymity of the vehicles, only the works that address some method of digital signature take into account these issues. In general, older papers did not have this concern. As for the integrity of the messages, practically all the works cited address this subject, except for Dotzer, Fischer e Magiera (2005) and Su et al. (2020). Shrivastava, Sharma e Chaurasia (2016) makes no mention of this. All works depend on the use of RSUs, except for Ravi e Kulkarni (2013) and Sakhreliya e Pandya (2014) and Dotzer, Fischer e Magiera (2005). Shrivastava, Sharma e Chaurasia (2016) doesn't mention that either.

The vast majority of the works are centrally managed, only Ravi e Kulkarni (2013), Dotzer, Fischer e Magiera (2005), Shrivastava, Sharma e Chaurasia (2016) and Kchaou, Abassi e Guemara (2018) are decentralized. Most do not deal with revoked certificates, except for Wasef e Shen (2013) and Wang e Yao (2017) which use CRL. Regarding the type of communication used, few authors mentioned: Jaimes, Ullah e Moreira (2016) is capable of using WAVE, Fourth Generation of mobile network (4G) or 5G; Mühlbauer e Kleinschmidt (2018) uses WAVE; Su et al. (2020) uses 5G. Most papers address some reputation system, except for the first six jobs shown in the Table 3.1.

About response time, most papers can be considered fast or medium, except for: Wasef e Shen (2013), due to the use of large lists of revoked certificates and complex key distribution management; Kchaou, Abassi e Guemara (2018), for using blockchain, there is a delay in mining the blocks. Mühlbauer e Kleinschmidt (2018) and Su et al. (2020) do not have enough data to assess their response times.

Finally, the proposed architecture, composed of DVNAT and the LETICIA reputation algorithm: addresses all the issues discussed; has a centralized management; handles revoked certificates without using CRLs; has a response time considered fast concerning the others.

Table 3.1 – Comparison of related works



Papers/Features	Message		Vehicle		RSU dependent	Management	Deals revoked certificate	Communi cation	Reputation system	Response time
	Reliability	Integrity	Non-repudiation	Anonymity						
Manvi; Kakkasageri; Adiga (2009)	no	yes	no	no	yes	centralized	no	-	no	medium
Wasef; Shen (2013)	no	yes	no	no	yes	centralized	CRL	-	no	slow
Ravi; Kulkarni (2013)	no	yes	no	no	no	decentralized	-	-	no	medium
Sakhreliya; Pandya (2014)	no	yes	yes	yes	no	centralized	no	-	no	medium
Wang; Yao (2017)	no	yes	yes	yes	yes	centralized	CRL	-	no	medium
Singh et al. (2015)	no	yes	no	no	yes	centralized	no	-	no	medium
Dotzer; Fischer; Magiera (2005) (VARS)	yes	no	no	no	no	decentralized	no	-	yes	fast
Li et al. (2013) (RGTE)	yes	yes	yes	no	yes	centralized	no	-	yes	medium
Shrivastava; Sharma; Chaurasia (2016)	yes	-	-	-	-	decentralized	-	-	yes	fast
Jaimes; Ullah; Moreira (2016) (ARS)	yes	yes	yes	yes – short-term certificate	yes	centralized	no	WAVE, 4G/5G	yes	medium
Kchaou, Abassi; Guemara (2018) (DTCMV)	yes	yes	yes	-	yes	decentralized	no	-	yes	slow
Mühlbauer; Kleinschmidt (2018) (BYOR)	yes	yes	yes	yes – short-term certificate	sometimes	partially decentralized		WAVE	yes	-
Su et al. (2020) (IDES)	no	no	no	no	sometimes	centralized	no	5G	yes	-
PROPOSED ARCHITECTURE (DVNAT + LETICIA)	yes	yes	yes	yes – short-term certificate	sometimes	centralized	yes	WAVE	yes	fast

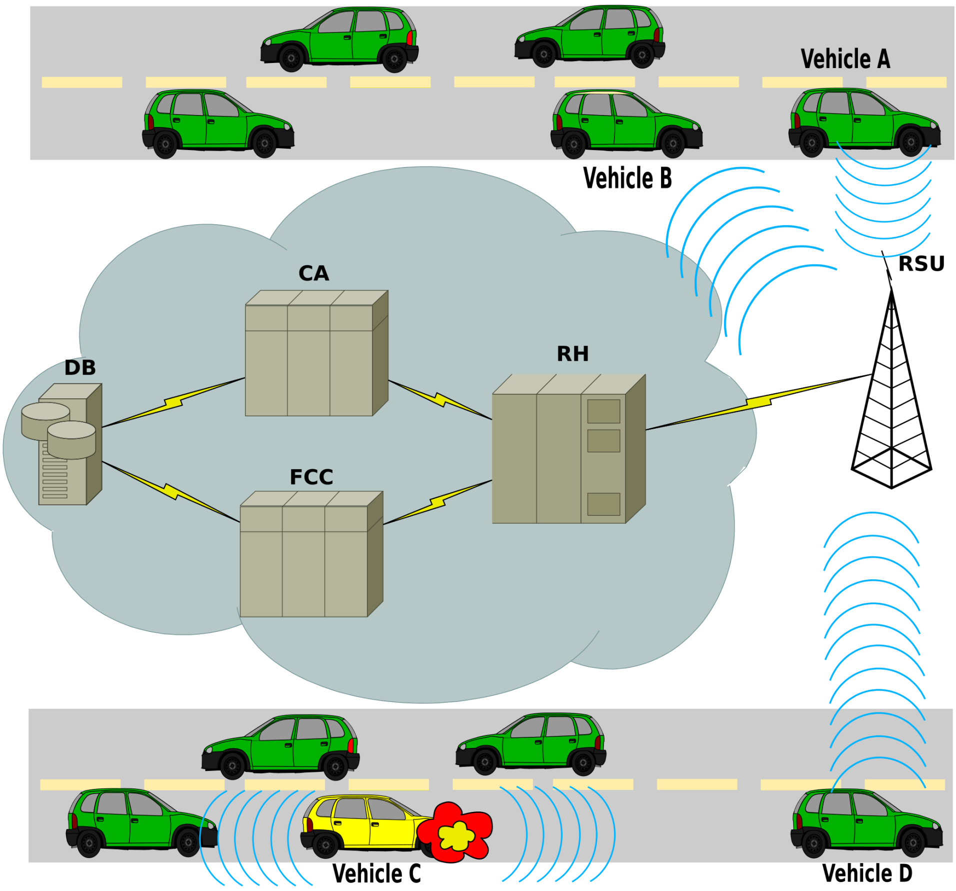
Source: author’s own (2020)

4 DVNAT - DEDICATED VEHICULAR NETWORK ARCHITECTURE

The purpose of this work is to create the DVNAT, a complete approach of a vehicular network, aiming to provide safety and a reputation system. A complete infrastructure has been created to provide digital signature services, distribution of digital certificates, and a centralized reputation system that is calculated based on driver feedback. At the core of the architecture is what we call **servers infrastructure**, which comprises four entities that play dedicated roles.

Figure Figure 4.1 shows an overview of the proposed architecture and the entities involved and their relationships. Session 4.2 will present how the architecture works. The following are all the elements and entities that make up this architecture.

Figure 4.1 – Architecture communications



Source: author's own (2020)

4.1 Architecture elements and assumptions

The following are the definitions of all entities used in this model and their preliminary specifications.

Vehicular network

The vehicles are the main elements of the network and are equipped with an OBU that has a radio frequency communication using WAVE for V2V and V2I communication. Besides, the vehicles have two digital certificates issued by a valid Certification Authority (CA): (a) a long-term digital certificate, used for authentication and identification of the vehicle by the transit authorities; (b) a short-term digital certificate, renewed periodically to sign messages that are transmitted by the vehicle. This period can be parametrized as necessary.

The objectives of the vehicular network are: (a) to provide information on traffic safety (vehicles, cyclists, pedestrians, animals, among others); (b) provide entertainment to network users through the sharing of videos, images, Internet access, among others (Brendha; Prakash, 2017; MISHRA; SINGH; KUMAR, 2016). Any message can be transmitted and shared between vehicles, as long as there is an application for that.

Long-term Certificates

The long-term certificate issuance should be linked to the vehicle licensing document . It is possible to establish periods of 1 to 5 years for their renewal. The transit authorities can carry out this renewal procedure. The long-term certificate uniquely identifies the vehicle and is used only to request short-term certificates.

Short-term certificates

Short-term certificates are issued periodically depending on the region, network density or the network configuration. It has a short shelf life and, if it expires before the vehicle renews it, the vehicle will no longer be able to send digitally signed messages. However, if an attacker tries to do this, vehicles that receive an expired certificate message will discard the message. This certificate must be renewed before it expires. When the vehicle is in the area covered by the RSU, it will request its renewal.

The short-term certificate solves two security issues: (a) the vehicle privacy, as its identity on the network is changed from time to time, preventing a malicious vehicle from being able to identify and track another vehicle; (b) eliminates consultation with the long list of revoked certificates, since if a vehicle has its long-term certificate revoked, it can not renew its short-term certificate.

RSU

The RSUs are used to send and receive messages between vehicles and the infrastructure. Many of these devices can be used at different points on the road to provide access to vehicles and other entities that need to communicate with the VANET infrastructure. RSUs are responsible for forwarding messages between vehicles and infrastructure. They handle three types of messages: (a) they receive feedback that the vehicles give for each message received; (b) receive short-term certificates renewal requests; (c) send short-term certificates requested by the vehicles.

Servers infrastructure

The servers infrastructure is the VANET servers cloud that contains all services necessary for the operation of the proposed VANET architecture. The Figure 4.2 illustrates this cloud, and each of the entities is explained below:

- **Request Handler (RH)**: handles requests received by RSU and forwards them to the corresponding server. It also decouples infrastructure and RSU, allowing the system to use other telecommunications systems, such as cell towers, Wi-Fi access points, among others.

- **Feedback Computation Center (FCC)**: receives, aggregates, and stores the feedback received in Data Base server (DB);

- **CA**: computes the reputation based on reputation historic and feedback stored in the database, and issues vehicle certificates;

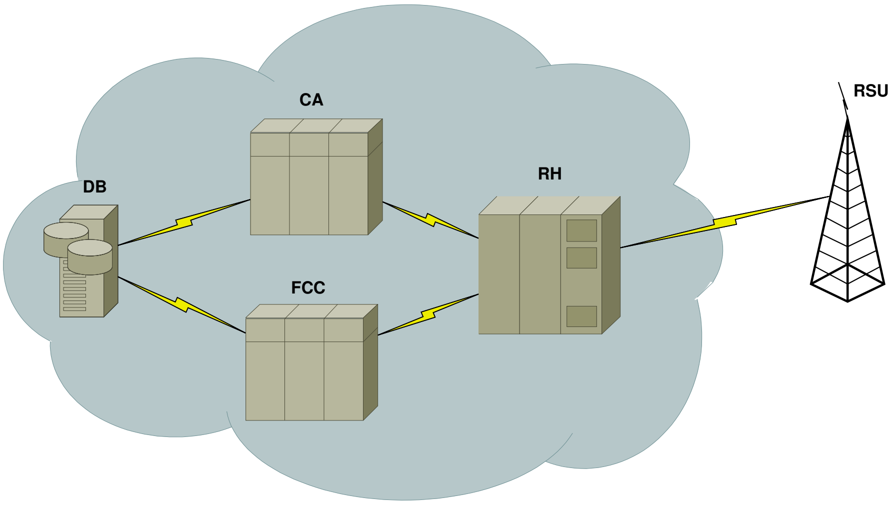
- **DB**: the database keeps the feedback received by FCC and vehicle reputations calculated by CA.

More details on the internal functioning of each of the infrastructure entities will be seen in Section 4.2.5.

Message types

There are two types of messages exchanged on the network: control and data messages. Control messages are small messages that manage the system. Data messages are the information that vehicles want to share with others. The vehicles can send a message in two

Figure 4.2 – Infrastructure entities



Source: author's own (2020)

ways: (a) directly, when the car's sensors capture an event, and the vehicle sends the alert message; (b) indirectly, when the vehicle is present in an event, and manually the driver informs the alert message. As these messages are of interest to everyone and need to be exchanged quickly, data messages are not encrypted, reducing overhead.

When a vehicle receives a message, it can check whether the message received is true or false. This verification can be done when passing through the incident site or by other means that will not be covered in this work. Upon entering the RSU coverage area, the vehicle sends an opinion about the message received. In this model, the opinions sent are called feedback. The feedback is control messages that only indicate the opinion of the driver (or vehicle) regarding the veracity of the received message.

The feedback will be computed later by the infrastructure to calculate the reputation of the vehicle that sent the message. The feedback is sent in encrypted form to RSU. The vehicle sends a control message making its request to request a new short-term certificate. When the certificate is ready, RSU delivers the certificate to the vehicle that is requested. This entire procedure is performed in an encrypted form.

Digital signature

All messages need to be digitally signed to guarantee authenticity and prevent the vehicle from changing its identity when its reputation is low or falsifying messages from other vehicles. The Ed25519 algorithm was used to sign digitally the messages in this architecture. It is a lightweight algorithm and proven to be viable in VANET, as shown in Natividade e Correia

(2020), which the authors compare its use in vehicular networks, testing its efficiency with the RSA and ECDSA, using the same security force for the three algorithms. The results of real hardware showed that the Ed25519 algorithm presented the best result against the others. It is about nine times faster than ECDSA and 42 times faster than RSA.

The digital signature can mitigate attacks such as impersonation, Sybil, newcomer, and betrayal. Data message encryption avoids attacks such as eavesdropping and MITM, although it is not used in this model.

Reputation

In this architecture, each vehicle has its reputation associated with its short-term certificate, as in Mühlbauer e Kleinschmidt (2018). The vehicle reputation level is given by a number between 0 and 1 (exclusive), as show the Equation 4.1. Each vehicle's reputation is recalculated with each request for a new short-term certificate and takes into account the vehicle's behavior when sending data messages on the network. Therefore, the reputation is attached to the short-term certificate and digitally signed by CA, and it is not possible to falsify it. It is up to the application or the specific scenario to define which levels are reliable or not for a vehicle. For example: in an environment, the reputation level considered reliable can be from 0.6 in others, from 0.4, and so on.

$$\{\rho \in \mathbb{R}/0 < \rho < 1\} \tag{4.1}$$

4.2 Architecture operation

The operations of sending and receiving messages, feedback, and certificates and the computation of vehicle reputation must be well established. The Figure 4.1 shows the entire procedure. `Vehicle A` represents a vehicle that is requesting for a new short-term certificate. `Vehicle B` shows receipt of a short-term certificate that was previously requested. `Vehicle C` represents a vehicle sending broadcast messages. On the other hand, `Vehicle D` shows the sending of feedback to the RSU regarding a previously received message. As for the requests/receipt of certificates and the sending of feedback, the vehicle must wait to be in a RSU coverage area. The RSU, in turn, forwards all messages received to the infrastructure to compute the new vehicle reputations and forward the requested certificate back. Below, each of these procedures will be presented in detail.

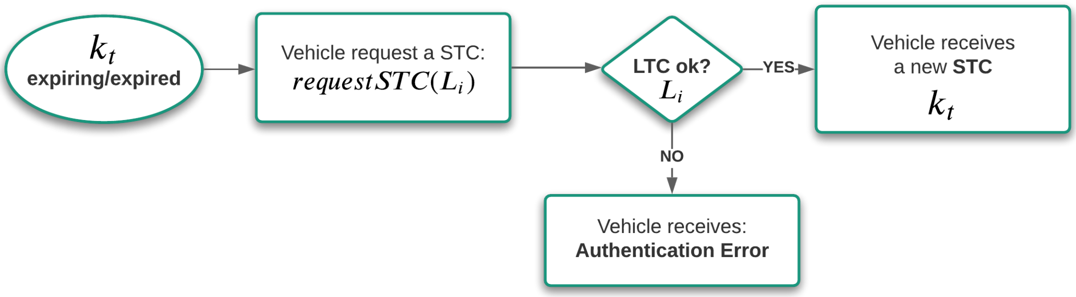
4.2.1 Obtaining a short-term certificate

Each vehicle must have a valid Long-term Certificate (LTC) L_i issued by a trust CA. At each predetermined period, the vehicle receives a temporary Short-term Certificate (STC) k_t from infrastructure. Figure 4.3 depicts the procedure for obtaining the short-term certificate from infrastructure.

(i) when the STC k_t is expiring, the vehicle makes a request to infrastructure informing its LTC L_i;

(ii) if the LTC is valid, the vehicle receives a new STC; otherwise, it receives an authentication error;

Figure 4.3 – Obtaining a short-term certificate



Source: author's own (2020)

4.2.2 Sending data messages

The Figure 4.4 shows the flowchart to sending data messages. The scheme works as follows:

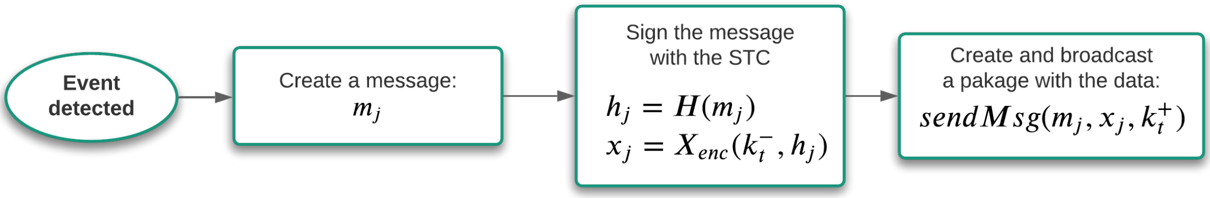
i) when an event is detected, the vehicle creates incident message m_j;

ii) it creates the message digital signature x_j;

iii) the vehicle creates a package containing: the message m_j, the message digital signature x_j, and the STC public key k_t^+;

iv) finally, the vehicle broadcasts the packet.

Figure 4.4 – Sending data messages


Event detected
Create a message: m_j
Sign the message with the STC
$h_j = H(m_j)$
$x_j = X_{enc}(k_t^-, h_j)$
Create and broadcast a pakage with the data:
$sendMsg(m_j, x_j, k_t^+)$

Source: author’s own (2020)

4.2.3 Receiving data messages

The Figure 4.5 shows the flowchart for receiving data messages and works as follows:

i) when a vehicle receives a data message, it needs to check some conditions before making a decision, as follows:

   (a) it checks the time of the message creation and its geographic localization, this is because ancient messages or that occurred in a very distant location can be discarded; times and distances must be parameterized according to the environment;

   (b) it verifies if the public key k_t^+ is not expired and it was issued by a trusted CA, using the CA public key K_{CA}^+, thus: if k_t^+ is not expired and $K_{CA}^+(k_t^+)$ is valid, then go to the next step; otherwise, the message is dropped;

   (c) it verifies the message authenticity: if $(H(m_j) == k_t^+(x_j))$, then go to the last step; otherwise, the message is dropped;

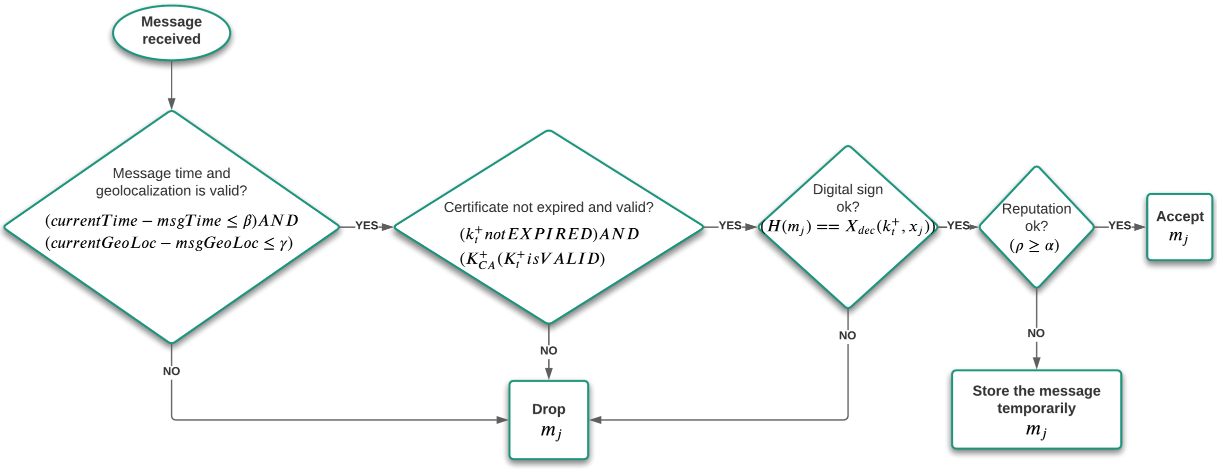
   (d) it verifies the reputation vehicle: if reputation ρ_i is greater than the threshold α, then the message is accepted; this threshold is parameterized according to the environment or application; otherwise, the message is stored temporally. This process depends on VANET application.

ii) all messages are stored (accepted or not) to the vehicle verifies the incident on-site and to send positive or negative feedback about the message.

4.2.4 Sending feedback

The scheme to send feedback about the received messages is present as follows:

Figure 4.5 – Receiving data message


Message received
Message time and geolocalization is valid?
$(currentTime - msgTime \leq \beta) AND$
$(currentGeoLoc - msgGeoLoc \leq \gamma)$
YES
Certificate not expired and valid?
$(k_t^+ notEXPIRED) AND$
$(K_{CA}^+ (K_t^+ isVALID)$
YES
Digital sign ok?
$(H(m_j) == X_{dec}(k_t^+, x_j))$
YES
Reputation ok?
$(\rho \geq \alpha)$
YES
Accept m_j
NO
NO
NO
NO
Drop m_j
Store the message temporarily m_j

Source: author’s own (2020)

i) when the vehicle V_i checks an incident on-site and verifies its truthfulness, it gives positive feedback to the vehicle that generated the message to increase its reputation;

ii) when V_i checks an incident on-site and proves that this is a false message, it gives negative feedback to the vehicle that generated the message to reduce its reputation;

iii) feedback is sent when the vehicle enters a RSU coverage area (V2I communication); the RSU, in turn, sends feedback to the infrastructure to be computed.

4.2.5 Operation of the servers infrastructure

The servers’ infrastructure internal working is shown in the cloud in Figure 2, and each entity’s details are described below. It should be noted that all services, applications, and entities contained in this infrastructure connects to RSU via Transmission Control Protocol/Internet Protocol (TCP/IP). As follows are described the infrastructure entities.

RH

RH is the server responsible for receiving and handling all requests from RSU through the OMNeT++ simulator. For this purpose, the MQTT[1] protocol was used through the broker Eclipse Mosquitto[2]. That is because MQTT is a lightweight and efficient publish/subscribe protocol, widely used in Internet of Things (IoT). Mosquitto is an open source broker for

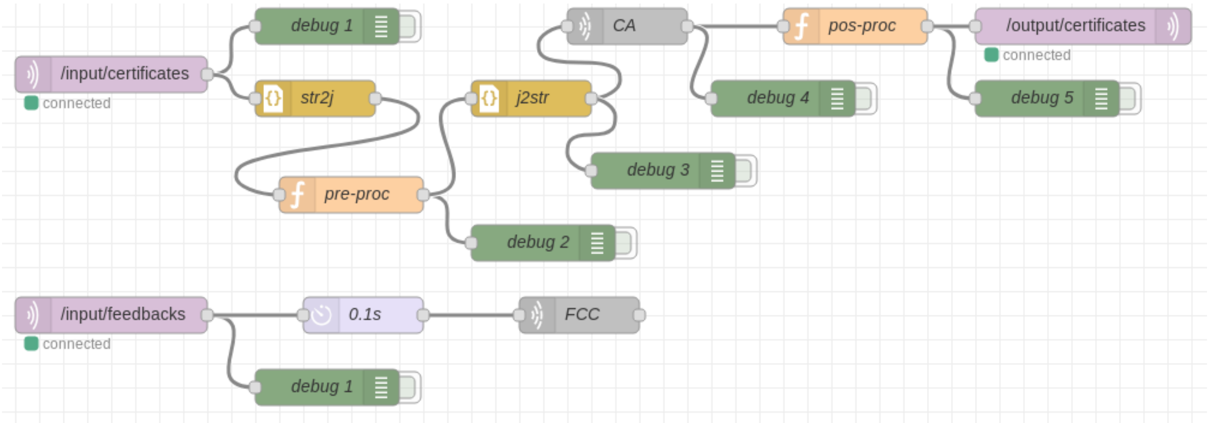
[1] MQTT is available at https://mqtt.org/

[2] Eclipse Mosquitto is available at https://mosquitto.org/

the Eclipse Foundation[3], widely supported by the community. For the implementation of communication via MQTT, the Paho[4] library was used, which is also developed by the Eclipse Foundation and has an open source implementation.

The request handler (RH) used is Node-RED[5], an open source flow-based programming tool for event-oriented applications. It is currently maintained by the OpenJS Foundation[6] and has been widely used by IoT applications. It has an easy configuration and is capable of handling various types of patterns and message formats. Figure 4.6 shows the flow created in Node-RED. New certificate requests are published in the MQTT `/input/certificates` directory, pre-processed, and delivered via socket to the CA server. The CA, in turn, does all the processing (seen below) and returns the certificate passing through post-processing and publishing it in the `/output/certificates` directory, which is received by RSU.

Figure 4.6 – Node-RED infrastructure



Source: author's own (2020)

The feedback is published in the `/input/feedback` directory, undergo a delay of 100 ms before being delivered to the FCC server. This delay was due to the implementation problem, which sometimes, there was a large feedback number arriving in a short time, creating a bottleneck in the FCC. All elements in green, seen in Figure 4.6, are "debugs"used in the implementation, but that remained disabled throughout the simulation.

[3] Eclipse Foundation website https://www.eclipse.org/

[4] Paho is available at https://www.eclipse.org/paho/

[5] Node-RED is available at https://nodered.org/

[6] OpenJS Foundation website https://openjsf.org/

FCC

FCC is a system made in C++, using the Crypo++ library, which remains listening on port 2775/Transmission Control Protocol (TCP), waiting for feedback from the simulation vehicles. The feedback is sent by the vehicles that received messages on the network to RSU. This process happens inside the simulator. RSU, in turn, forwards the request to RH, which directs the request to FCC, using the corresponding flow.

Feedback is received in packages in the format JavaScript Object Notation (JSON)[7], in which they can contain one or more feedbacks from the same vehicle about others vehicles, as the vehicle must store its feedbacks until it is in the coverage area of RSU to send them. The FCC then aggregates the feedbacks received, according to the selected reputation algorithm, and stores them in the DB database; FCC only sends feedback to DB, without making any other queries in the database.

CA

CA is an application developed in C++ with Crypto++, which is listening for connections on port 1393/TCP. It waits for requests for new short-term certificates in the format JSON. These requests are made by vehicles that have their certificate expiring (or expired) and sent to RSU. The RSU forwards the request to RH, which directs the request to CA through the corresponding flow. CA queries the DB for the vehicle's old reputation and received feedback, and computes the new reputation according to the selected reputation algorithm. Finally, CA should delete old feedback from the database if the reputation algorithm used does not use feedback histories for its calculations.

DB

The DB is the database server responsible for storing each vehicle's reputations in the simulation, and their feedback received. For simplification, the local database SQLite[8] version 3 was used in this model. However, any other Data Base Management System (DBMS) could be used, merely changing the application's corresponding communication driver. In this architecture only two tables are used in the database:

- **REPUTATION**: stores the reputations of each vehicle in the network;

[7] JSON is available at https://www.json.org/

[8] SQLite is available at https://www.sqlite.org/

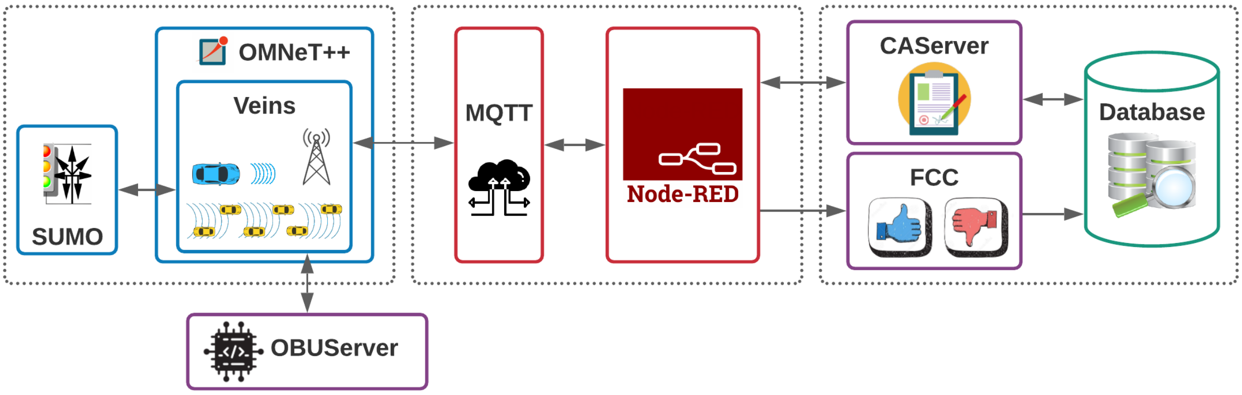
- **FEEDBACK**: stores the feedback received from each vehicle in the network; additional fields can be used, depending on the selected reputation algorithm.

4.3 Simulation operation

In a real environment, each vehicle would have its own OBU operating independently. However, in a simulated environment, some changes need to be made. Figure 4.7 synthesizes the entire operation of the infrastructure and its communications in the simulated environment. The first rectangle shows the iteration between the simulators used and the communication of the vehicles. The second and third rectangles represent the servers infrastructure. In this way, there is a decoupling between the servers infrastructure and the simulated environment, in which it can be replaced by a real VANET environment. OBUServer, the bottom of the first rectangle, works as a shared OBU for all vehicles in the simulations.

OBUServer was the alternative to compute in real-time the time spent by digital signature operations. OBUServer is a system made in C++ and Crypto++ library responsible for signing and checking all messages sent and received by all vehicles. The time spent in this process is added to the simulation time. The OBUServer can run on embedded hardware to obtain verification and digital signature times closer to a real OBU, as in Natividade e Correia (2020). In the second rectangle, it is possible to observe the RH, containing the MQTT and Node-RED services. They receive messages from the simulators and pass them on to the corresponding entity in the third rectangle. In this stage, there are the CAServer, for issuing certificates, the FCC, to receive feedback, and the DB to store the information permanently.

Figure 4.7 – Simulation infrastructure



Source: author’s own (2020)

5 LETICIA REPUTATION ALGORITHM

The LETICIA algorithm was developed to assess the reputation of vehicles that exchange messages on VANET. The algorithm works in a centralized way, which a server infrastructure processes and distributes the nodes' reputation according to the feedback received from other vehicles in the network. LETICIA was designed to be able to deal mainly with inconsistency and collusion attacks by bad-mouthing. This algorithm quickly reduces the reputation of a vehicle that sends fake messages on the network and gradually increases its reputation when sending true messages. This strategy aims to reduce inconsistency attacks, preventing the vehicle from rapidly increasing its reputation. For reputation calculation is considered: (a) the old reputation of the assessed vehicle; (b) feedback from neighboring vehicles and their reputation; (c) the time since creating the message and the moment when it was received by the vehicle that sent the feedback.

The infrastructure aggregates and stores the feedback sent about each vehicle in two counters: one to count the positive feedback and the other for the negative feedback. Each feedback received, whether positive or negative, is calculated according to Equation 5.1. The Table 5.1, shows all the variables used in calculating the aggregation of feedback and vehicle reputation in the LETICIA algorithm.

$$\begin{cases} F^{+} = \sum_{i=1}^{m} \frac{\rho_{fb_i} + (1 - \frac{T_{fb_i}}{\varepsilon})}{2} \\ F^{-} = \sum_{i=1}^{n} \frac{\rho_{fb_i} + (1 - \frac{T_{fb_i}}{\varepsilon})}{2} \end{cases} \tag{5.1}$$

In this equation, F^{+} is the counter that receives the sum of positive feedback, and F^{-} is the one that receives the sum of negative. ρ_{fb_i} is the vehicle reputation that sent the feedback, T_{fb_i} is the time since the message was created and the moment it was received by the vehicle which sent the feedback, and ε is the timeout for a message.

The aggregation of feedback calculation is given by Equation 5.2, which uses the Beta Probability distribution function, as in Jøsang, Ismail e Boyd (2007), Mühlbauer e Kleinschmidt (2018), Cervantes et al. (2014). A_{fb} is the aggregation of positive and negative feedback, α is the sum of positive feedback (F^{+}) + 1, and β, is the sum of negative feedback (F^{-}) + 1.

$$A_{fb} = \frac{\alpha}{(\alpha + \beta)} \tag{5.2}$$

Table 5.1 – Variables used in the LETICIA algorithm calculations

Variable	Description
F^{+}	summation of positive feedback
F^{-}	summation of negative feedback
m	number of vehicles that sent positive feedback
n	number of vehicles that sent negative feedback
ρ_{fb_i}	vehicle reputation that sent feedback
T_{fb_i}	message time to the vehicle that sent feedback (in seconds)
ε	the timeout of a message on the network
A_{fb}	aggregated feedback
α	positive feedback + 1
β	negative feedback + 1
ρ_0	the old reputation of the rated vehicle
ρ	the new reputation of the rated vehicle

Source: author's own (2020)

After aggregating the feedback, it is necessary to calculate the new reputation based on the old reputation and aggregated feedback. When the received feedback aggregation (A_{fb}) is greater than 0.5, it indicates that, in that iteration, most of the opinions about the vehicle were positive and that the value of its reputation should increase, as shown in the Equation 5.3. This increase is amortized by a quadratic factor that smoothly raises the vehicle's reputation.

$$\rho = \rho_0 + \rho_0 * A_{fb} - {\rho_0}^2 * A_{fb} \tag{5.3}$$

Suppose the feedback aggregation is less than or equal to 0.5. In that case, the value of A_{fb} indicates that the majority of opinions about the vehicle were negative and that its reputation value should be reduced, given by Equation 5.4. Its reduction is extreme since, in addition to the dependence on calculating the previous reputation and feedback aggregation, the value is still reduced by half.

$$\rho = \frac{\rho_0 + \rho_0 * A_{fb}}{2} \tag{5.4}$$

The feedback aggregation and reputation computation procedures can be exemplified by the following algorithms. Algorithm 1 shows the feedback aggregation process, in which line 1 increments the positive feedback counter, and line 5 increments the negative feedback counter.

Algorithm 1 – Feedback aggregation

```
if feedback = 1 then
    A = positiveFeedbackAggregation(feedback) {Equation 5.1 $F^{+}$}
else
    if feedback = -1 then
        A = negativeFeedbackAggregation(feedback) {Equation 5.1 $F^{-}$}
    end if
end if
```

Source: author's own (2020)

Algorithm 2 illustrates the reputation calculation procedure. Line 2 increases the vehicle's reputation if A (feedback aggregation) is greater than 0.5, while line 4 decreases the reputation. In both cases, the value of A and the old reputation (ρ_0) are taken into account.

Algorithm 2 – Reputation computation

```
1: if A > 0.5 then
2:     $\rho$ = increaseRho(A, $\rho_0$) {Equation 5.3}
3: else
4:     $\rho$ = decreaseRho(A, $\rho_0$) {Equation 5.4}
5: end if
```

Source: author's own (2020)

Equation 5.3 and Equation 5.4 were found empirically. A value between 0 and 1 was needed for the new reputation found, taking into account the old reputation and the aggregated feedback. For the calculation of reputation increase (EQUATION 5.3), it was thought to gradually increase, adding the old reputation by the product of the old reputation by the aggregation of feedback found: $\rho_0 + \rho_0 * A_{fb}$. It would result in a very high reputation, possibly exceeding the defined range of $]0,1[$. Therefore, to normalize and prevent the rapid increase in reputation, the negative term composed by the old reputation square's product by aggregating the feedback was added to the equation: $-{\rho_0}^2 * A_{fb}$. With this, it was possible to amortize the increase in reputation.

For the reputation reduction calculation (EQUATION 5.4), the reputation is reduced more quickly than the increase. We chose to calculate half the sum of the old reputation by the product of the old sum by aggregating feedback: $\frac{\rho_0+\rho_0*A_{fb}}{2}$. The maximum and minimum values found in reputation reduction calculations can be represented by Equation 5.5 and Equation 5.6, respectively. Thus, as in this equation, the value of A_{fb} is between $]0,0.5]$, the reputation values found are between $]0,0.75[$. That is, in any scenario, the reputation reduction would be at least 25%.

$$\rho = \lim_{\rho_0 \to 0} \frac{\rho_0 + \rho_0 * A_{fb}}{2} \tag{5.5}$$

$$\rho = \lim_{\rho_0 \to 1} \frac{(\rho_0 + \rho_0 * A_{fb})}{2} \tag{5.6}$$

6 METHODOLOGY

The purpose of this work is to create an architecture with a complete approach of a vehicular network, aiming to provide safety and comfort to the drivers through the sharing of accident messages.

6.1 Simulators and tools

In VANET, the use of real vehicles for testing is often impractical due to the high cost of hardware for vehicular communication and network deployment. In this way, the simulators are widely used for this purpose. These simulators should be as close as possible to the real world to obtain a satisfactory result (CAVIN; SASSON; SCHIPER, 2002). To perform simulations on vehicular networks, it is necessary: (a) a vehicle mobility simulator to simulate the route of the vehicles on the road; (b) a network simulator to create a system of communication and exchange of messages between the nodes of the network; (c) a middleware, which integrates the two previous simulators, (SPAHO et al., 2011). In this way, the network's mobile nodes become vehicles that communicate with each other, as expected in a vehicular network.

There are currently several vehicle mobility simulators, network simulators, and integration middlewares, as shown in the papers Cavin, Sasson e Schiper (2002) and Spaho et al. (2011). However, the open-source framework Vehicles in Network Simulation (VEINS)[1] was used to accomplish this work. VEINS integrates the SUMO[2] vehicle traffic simulator and the OMNeT++[3] network simulator (SOMMER; GERMAN; DRESSLER, 2011). According to the VEINS project website, more than 800 publications have been made in VANET using the VEINS framework over the last ten years, (SOMMER, 2019).

Next, a brief explanation of the simulators and tools used to accomplish this work.

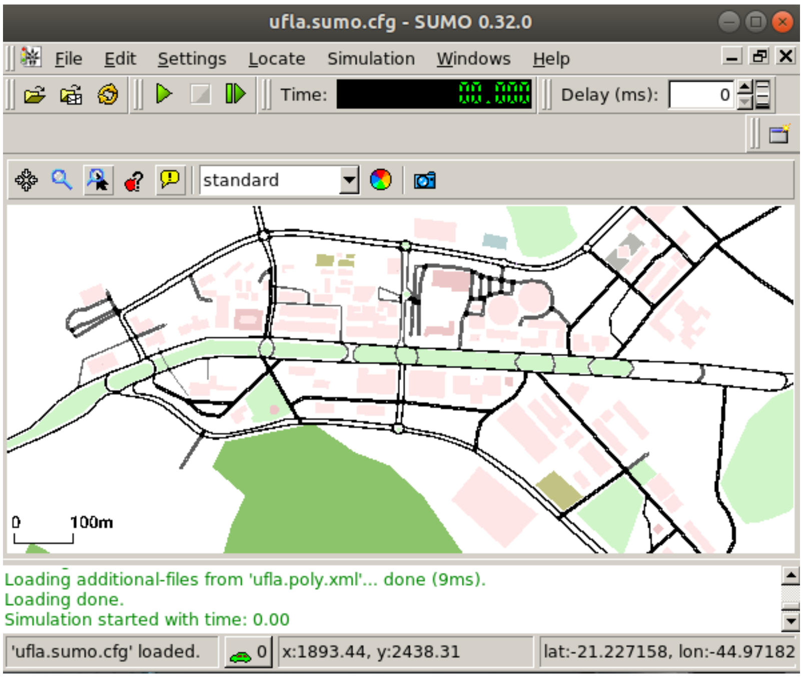
6.1.1 SUMO

The Simulation of Urban Mobility (SUMO) is an open source vehicle traffic simulator, which has a set of tools to help prepare and run the simulations. It is capable of generating outputs for each vehicle with diverse information, individual or aggregated. SUMO also has an Application Programming Interface (API) that allows integration with other systems,

[1] VEINS is available at https://veins.car2x.org
[2] SUMO is available at https://sumo.dlr.de
[3] OMNeT++ is available at https://omnetpp.org

(BEHRISCH et al., 2011). Second Lopez et al. (2018), SUMO is considered a microscopic traffic simulator because each vehicle and its dynamics can be modeled individually. The Figure 6.2 shows SUMO simulator screen.

Figure 6.1 – SUMO simulator



ufla.sumo.cfg - SUMO 0.32.0

Time: 00.000

Delay (ms): 0

standard

Loading additional-files from 'ufla.poly.xml'... done (9ms).
Loading done.
Simulation started with time: 0.00

'ufla.sumo.cfg' loaded. | 0 | x:1893.44, y:2438.31 | lat:-21.227158, lon:-44.97182

Source: author's own (2020)

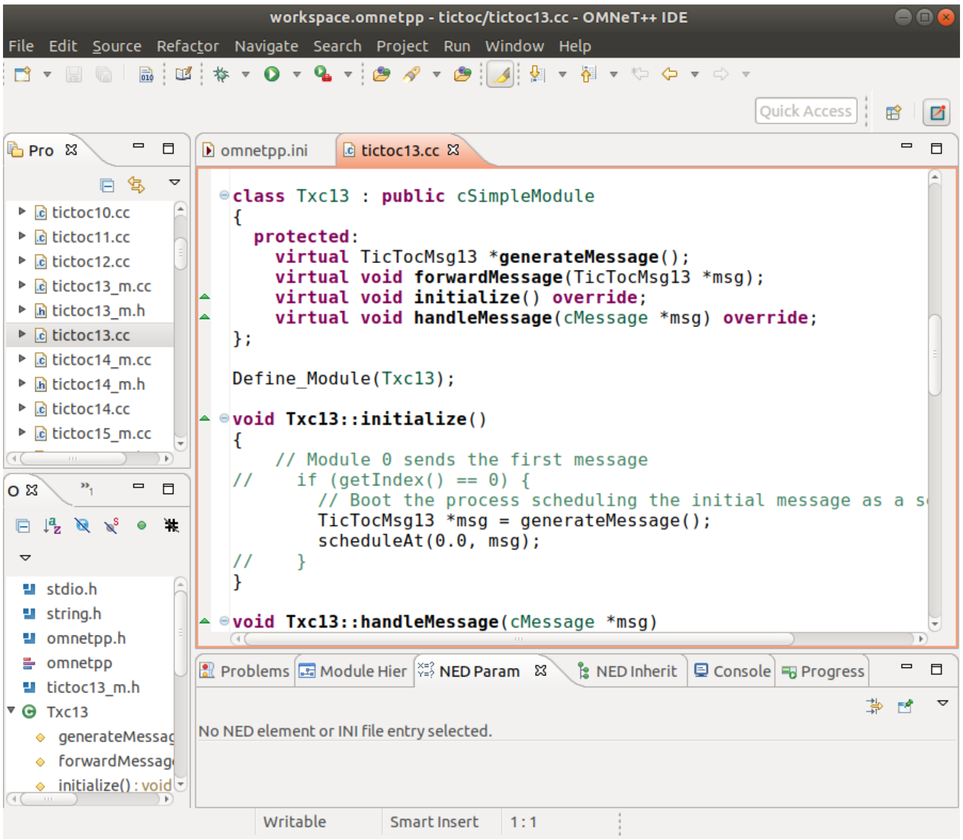
6.1.2 OMNeT++

The OMNeT++ is a modular, extensible, component-based C++ network simulation framework, that supports simulations of various types of networks. It has an Eclipse-based Integrated Development Environment (IDE) and simulations can be run in graphical or console mode. OMNeT++ can be used freely for academic purposes, (VARGA, 2001) and (VARGA, 2010). The Figure 6.2 shows OMNeT++ simulator screen.

6.1.3 Veins

Veins is an open-source framework for the simulation of vehicular networks, which integrates the OMNeT++ network simulator and the SUMO traffic mobility simulator, with two-way coupling between them. This coupling allows SUMO to send the vehicle's position to OMNeT++. It allows the OMNeT++ to react to received commands, creating, erasing, and changing the vehicles' position during the simulation (SOMMER; GERMAN; DRESSLER, 2011). According to the VEINS website, its main characteristics are:

Figure 6.2 – OMNeT++ simulator



```
class Txc13 : public cSimpleModule
{
  protected:
    virtual TicTocMsg13 *generateMessage();
    virtual void forwardMessage(TicTocMsg13 *msg);
    virtual void initialize() override;
    virtual void handleMessage(cMessage *msg) override;
};

Define_Module(Txc13);

void Txc13::initialize()
{
    // Module 0 sends the first message
//    if (getIndex() == 0) {
        // Boot the process scheduling the initial message as a s
        TicTocMsg13 *msg = generateMessage();
        scheduleAt(0.0, msg);
//    }
}

void Txc13::handleMessage(cMessage *msg)
```

No NED element or INI file entry selected.

Source: author’s own (2020)

i) 100% open source software;

ii) enables on-line reconfiguration and vehicle redirection in response to network packets;

iii) has on a reliable model of vehicle mobility;

iv) uses the IEEE 803.11p and IEEE 1609.4 standards;

v) uses models for the cellular network;

vi) can be used on a single machine or a cluster of distributed computers;

vii) able to import various elements of OpenStreetMap, such as buildings, speed limits, scrolling lanes, traffic lights, among others;

viii) able to use interference and shadowing patterns caused by buildings and vehicles.

6.1.4 Crypto++

Crypto++[4] is a free library that implements cryptography algorithms. Currently there are several cryptographic libraries such as Botan[5], Bouncy Castle[6], cryptlib[7], libgcrypt[8], NaCl[9], OpenSSL[10], wolfCrypt[11], etc. Crypto++ was chosen to implement digital signature in this work since:

i) it is easy to use;

ii) there are several examples available on the project website;

iii) implements the cryptography algorithm that will be used in this work, Ed25519;

iv) was recently used by researchers, as in Jaimes, Ullah e Moreira (2016);

v) it is updated continuously;

vi) its implementation is in C++, which is the language used in the used simulator.

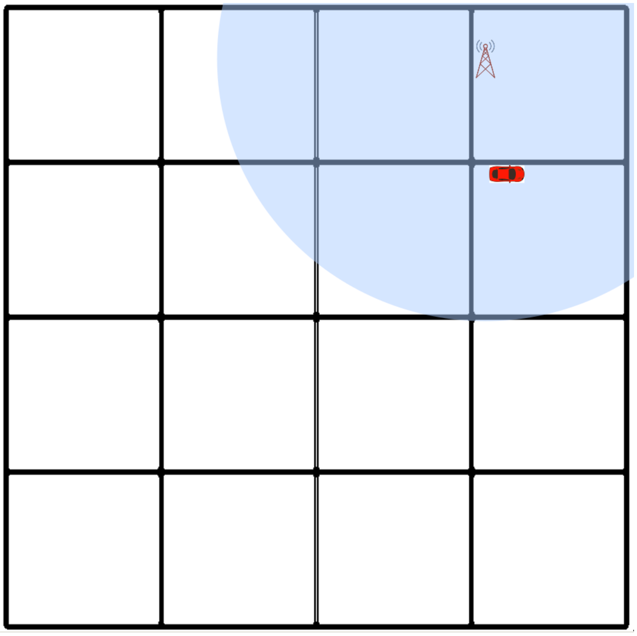
6.2 Scenario

The tests were performed using a Manhattan-type scenario, with five horizontal paths for five vertical ones of 200 meters each, as shown in Figure 6.3. In this model, a vehicle is determined to be sending messages at each time interval. The vehicle in question remains parked at a point on the road, over the RSU coverage (blue shadow), as shown in the image. The coverage area of the RSU represented in the image is approximate, as Veins does not use a fixed radius as the coverage area but performs mathematical calculations to approximate the real world.

As soon as the vehicle that sends the messages enters the simulation, it parks at the indicated point and remains there until the simulation's end, sending messages during the specified time. The other vehicles that receive messages circulate throughout the grid. However,

[4] Crypto++ is available at https://www.cryptopp.com
[5] Botan is available at https://botan.randombit.net
[6] Bouncy Castle is available at http://bouncycastle.org
[7] cryptlib is available at https://www.cs.auckland.ac.nz/ pgut001/cryptlib
[8] libgcrypt is available at https://dev.gnupg.org/source/libgcrypt
[9] NaCl is available at https://nacl.cr.yp.to
[10] OpenSSL is available at https://www.openssl.org
[11] wolfCrypt is available at https://www.wolfssl.com/

Figure 6.3 – Manhattan grid 5x5



Source: Source: author's own (2020)

the final destination is always the top right corner of the grid, forcing them to always pass through the coverage area of RSU so that they can deliver their feedback.

Simulation parameters can be seen in Table 6.1, in which the simulation time in the OMNeT++ and SUMO simulators is 1000 seconds. The parked vehicle starts with a reputation of 0.6, sends a message every two seconds, and renews its certificate every 10 seconds, consequently receiving its new reputation. The other vehicles on the road receive a random reputation, according to the seed used. The tests are repeated 33 times, with seeds from 660 to 692. The graphs are generated with a confidence level of 95%.

Table 6.1 – Simulation parameters

Simulations parameters	**Value**
Simulation time	1000s
Scenario size	800m x 800m
Vehicle initial reputation	0.6
Message sending interval	2s
Time to certificate renew	10s
Random reputation	from 0.1 to 0.99
Tests repetition	33
Seeds	from 660 to 692
Confidence level	95%

Source: author's own (2020)

6.3 Tests

In the simulations performed, two types of configured attacks: inconsistency attack, illustrated in Figure 6.4, and bad-mouthing collusion attack, illustrated in Figure 6.5. These attacks were divided into subgroups, using a taxonomy defined by the authors Natividade, Correia e Santos (2020): bipolar, restricted, and distributed. In simulations, the types, sub-types, and probabilities of attacks are parameterized, simply choosing them as desired.

Inconsistency attack

An attacker behaves inconsistently, alternately sending true and false messages, thus compromising the functioning of the network (ZHANG, 2011). In Figure 6.4, the red vehicle (`A`) performs the inconsistency attack by sending messages on the network, either true or false. The green vehicles that give their opinion (`1..n`), always emit a reliable feedback, that is, positive feedback for true messages and negative *feedback* for false messages. The subgroups for the inconsistency attack are:

i) **bipolar inconsistency attack**: the vehicle sends a true and a false message, alternately;

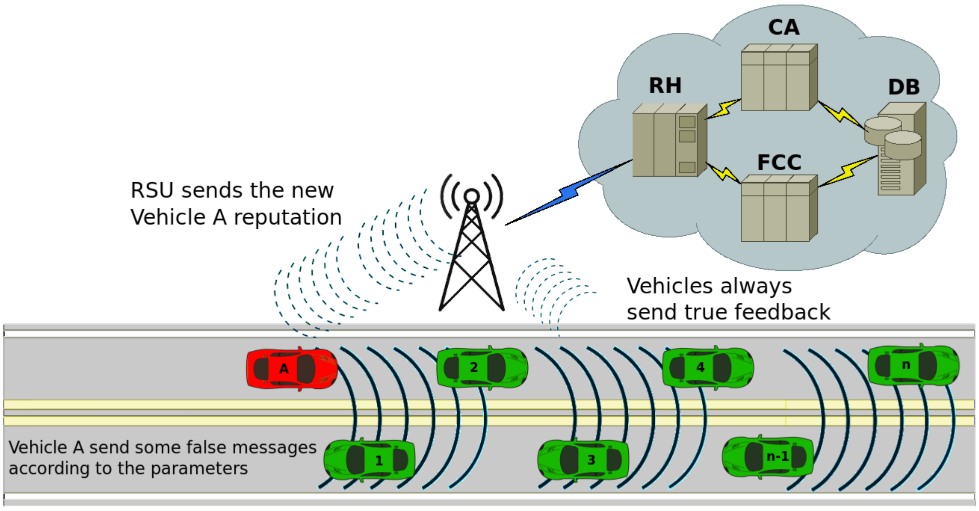
ii) **restricted inconsistency attack**: the vehicle sends true and false messages at a fixed rate. For example, it sends 20 true and 20 false messages, repeating this behavior until the end of the simulation;

iii) **distributed inconsistency attack**: throughout the simulation, the malicious vehicle is likely to send fake messages. For example, of all messages sent in the simulation, 40% are false.

Bad-mouthing collusion attack

Malicious attackers act in groups and arbitrarily issue a positive or negative feedback about a vehicle on the network to change its reputation so that other vehicles make incorrect decisions about it (BANKOVIĆ et al., 2011). In the attack shown in Figure 6.5, the blue vehicle (`A`) that sends messages on the network always does it suitably. That is, it always sends real messages on the network. The vehicles that give their feedback (`1..n`) are the ones who make the attacks against the reputation of the vehicle that sent the message, sometimes giving a positive opinion (green) and sometimes a negative one (red). The subgroups for these attacks are:

Figure 6.4 – Inconsistency attack operation

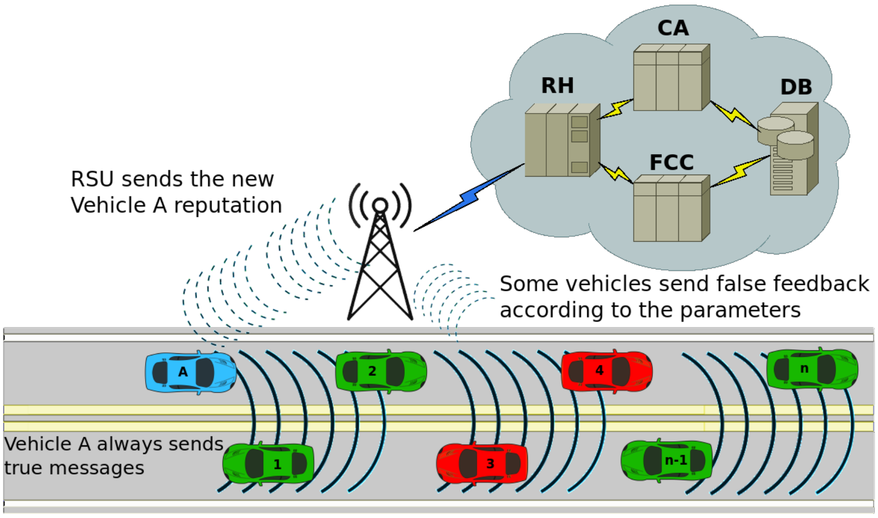

CA
RH
DB
FCC
RSU sends the new Vehicle A reputation
Vehicles always send true feedback
A
2
4
n
1
3
n-1
Vehicle A send some false messages according to the parameters

Source: adapted from Natividade, Correia e Santos (2020)

i) **restricted bad-mouthing collusion attack**: vehicles with a particular reputation below the specified rate always send negative feedback. For example, vehicles with a reputation below 0.3 always send negative feedback;

ii) **distributed bad-mouthing collusion attack**: in this attack throughout the simulation, vehicles that express their opinion are likely to send negative feedback incorrectly. For example: of all vehicles that express their opinion, 20% do so incorrectly, that is, sending negative feedback.

Figure 6.5 – Bad-mouthing collusion operation


CA
RH
DB
FCC
RSU sends the new Vehicle A reputation
Some vehicles send false feedback according to the parameters
A
2
4
n
1
3
n-1
Vehicle A always sends true messages

Source: adapted from Natividade, Correia e Santos (2020)

6.4 Metrics

This work compared five reputation algorithms in vehicular networks, ARS, BYOR, BYOR-LF, IDES, and LETICIA, using DVNAT. The metric used in the comparisons was the reaction of a vehicle's reputation over several iterations, taking into account its behavior and neighbors' behavior who evaluated its messages.

7 RESULTS AND DISCUSSION

The results of the reputation algorithms tests evaluated in the DVNAT are presented below, against attacks of inconsistency and collusion by bad-mouthing. The tested algorithms were ARS, BYOR, BYOR-LF, IDES, and the proposed algorithm, LETICIA.

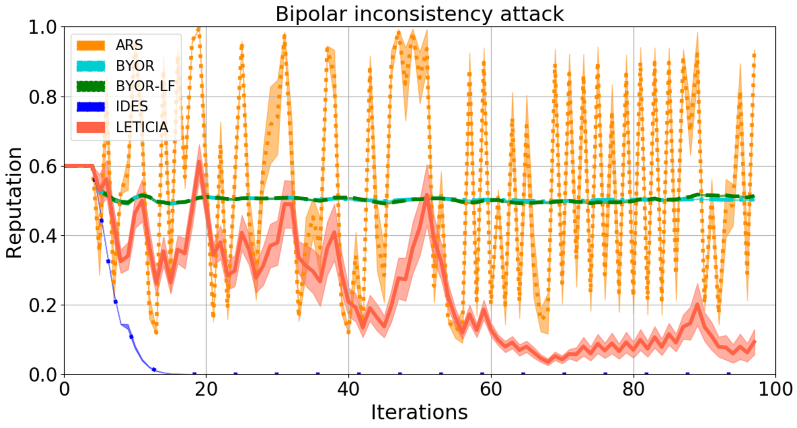
7.1 Inconsistency attack

The reputation algorithms have been evaluated in DVNAT against inconsistency attacks, in which an attacker alternates by sending true and false messages to the network's vehicles. The subgroups of attacks of bipolar inconsistency, restricted and distributed, were evaluated.

Bipolar inconsistency attack

In a bipolar inconsistency attack, the vehicle sends a true message, followed by a false one. It repeats this behavior until the end of the simulation. Figure 7.1 shows the result of the bipolar inconsistency attack for all algorithms.

Figure 7.1 – Bipolar inconsistency attack



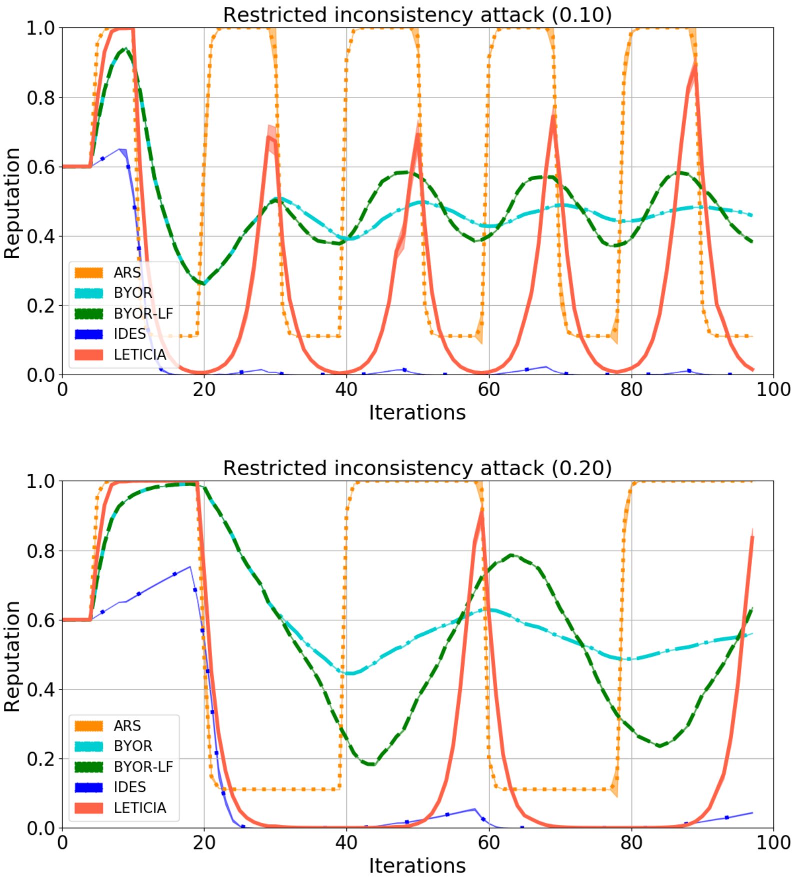
Source: author's own (2020)

The ARS algorithm was the most unstable, varying the reputation of the malicious vehicle between 0.15 to 0.99, classifying as a high reputation and low reputation, following the attacker's behavior. The BYOR and BYOR-LF algorithms had similar behaviors, which the vehicle's reputation was stabilized at 0.5. Although these algorithms negatively rated the vehicle, they reacted complacently to the attacks and did not reduce the vehicle's reputation as the number of iterations increased. IDES soon the beginning, it dropped its reputation close to zero, giving the vehicle no chance to resume a higher reputation. The LETICIA algorithm rated the malicious vehicle negatively, reacting more quickly against the attack and reducing its reputation as the number of iterations increased.

Restricted inconsistency attack

For this attack, the malicious vehicle alternates true and false messages at a fixed rate. In this simulation, for a total of 100 messages, the malicious vehicle sent 10 or 20 false messages, followed by 10 or 20 true messages, repeating this behavior until the end of the simulation. Figure 7.2 shows the result of the restricted inconsistency attack for all algorithms.

Figure 7.2 – Restricted inconsistency attack with vehicle alternately sends 10 and 20 false messages, after 10 and 20 true messages, respectively



Restricted inconsistency attack (0.10)

Restricted inconsistency attack (0.20)

Source: author’s own (2020)

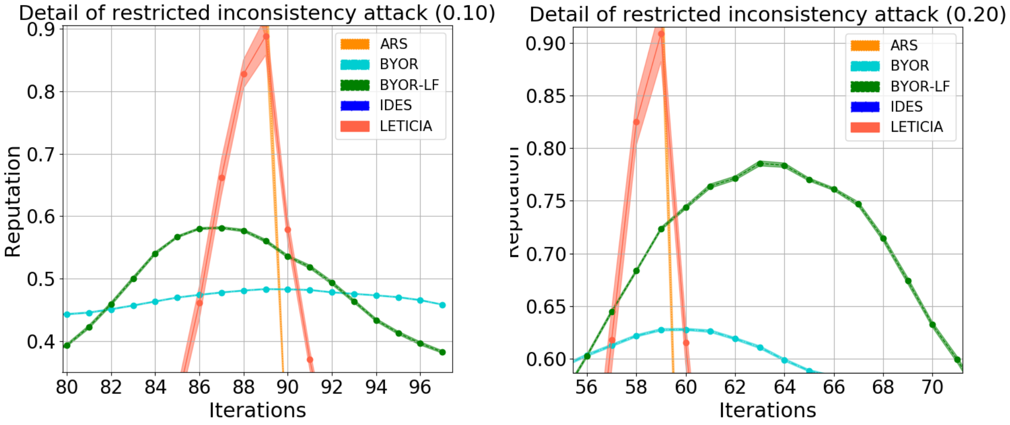
ARS follows the behavior of the attacking vehicle, alternating between negative and positive reputations according to the rate of sending false messages. The BYOR and BYOR-LF algorithms had similar behaviors for both rates, up to approximately the 30th iteration. After that, BYOR-LF becomes a little more aggressive in falls and resumes of reputation, due to its longevity factor. In general, both maintain the vehicle's reputation between 0.6 and 0.4.

IDES outlined a slight rise at the beginning. However, soon after the first false messages, it dropped its reputation close to zero, giving the vehicle no chance to resume a higher reputation. After 20 false messages in a row, the algorithm does not allow the vehicle's reputation to exceed 0.1.

There is a clear view for the 10 true/10 false attacks that LETICIA maintains the vehicle's reputation close to IDES. However, it has small recovery peaks that follow the vehicle's behavior, but it falls again, coming close to zero. In the attack of 20 true/20 false, the vehicle presents higher peaks. However, it remains with a relatively high reputation for a short time about the other algorithms, remaining most of the time with the reputation close to zero, which is desirable.

The Figure 7.3 shows in detail the recovery peaks from 10/10 and 20/20 attacks, respectively. Each point on the graph corresponds to an iteration. The first graph shows the detail between iterations 80 and 97, in which BYOR remains reputed above 0.4 for 18 iterations. BYOR-LF takes 17 iterations. While LETICIA, in just six iterations, the vehicle's reputation can rise and fall below this threshold. The second graph shows the peaks between iterations 56 and 71, in which BYOR-LF remains reputed above 0.6 for nine iterations while BYOR takes 16 iterations. On the other hand, LETICIA remains above 0.6 for just four iterations.

Figure 7.3 – Detail of restricted inconsistency attack with vehicle alternately sends 10 or 20 false messages, after 10 or 20 true messages, respectively


Detail of restricted inconsistency attack (0.10)

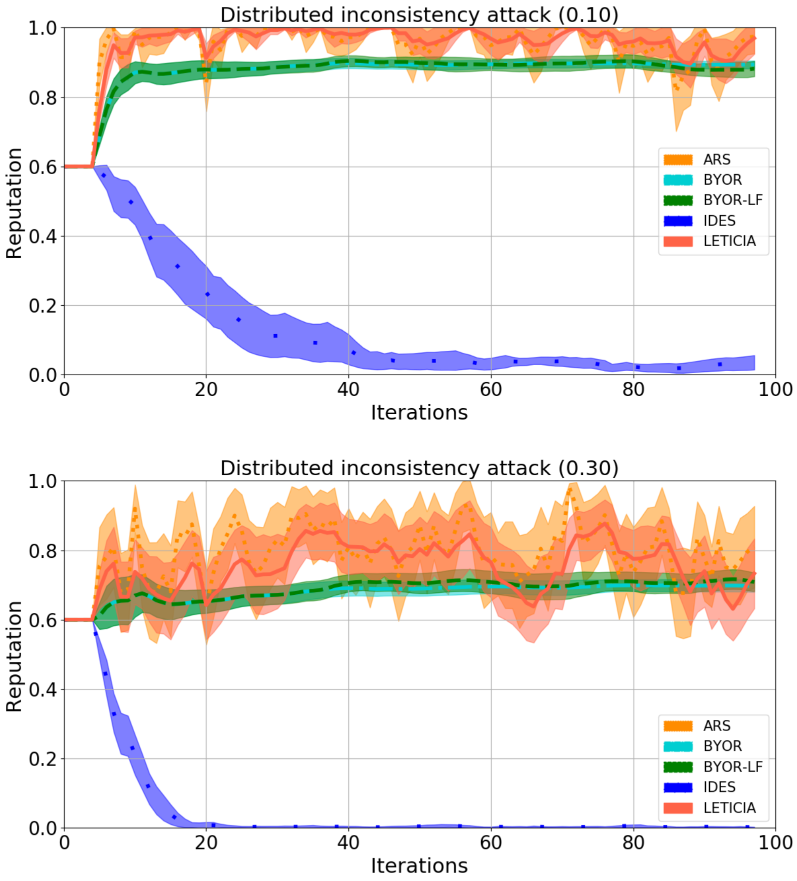
Detail of restricted inconsistency attack (0.20)

Source: author's own (2020)

Distributed inconsistency attack

The malicious vehicle probabilistically sends false messages in this attack, following distribution of 10%, 30%, 40%, 50%, and 70%. The Figure 7.4 shows the simulation results for all algorithms with 10% and 30% of attacks distribution. For the 10% distribution, the results are similar for LETICIA and ARS, which maintains the malicious vehicle's reputation at an average of 0.95. BYOR and BYOR-LF manage to reduce their reputation a little more, staying at approximately 0.90. However, IDES manages to more aggressively lower the vehicle's reputation to near zero, right in the 45th iteration. However, the attack percentage can still be considered low for such a dramatic reduction in reputation.

Figure 7.4 – Distributed inconsistency attack with vehicle sending false messages 10% and 30% of the time, respectively

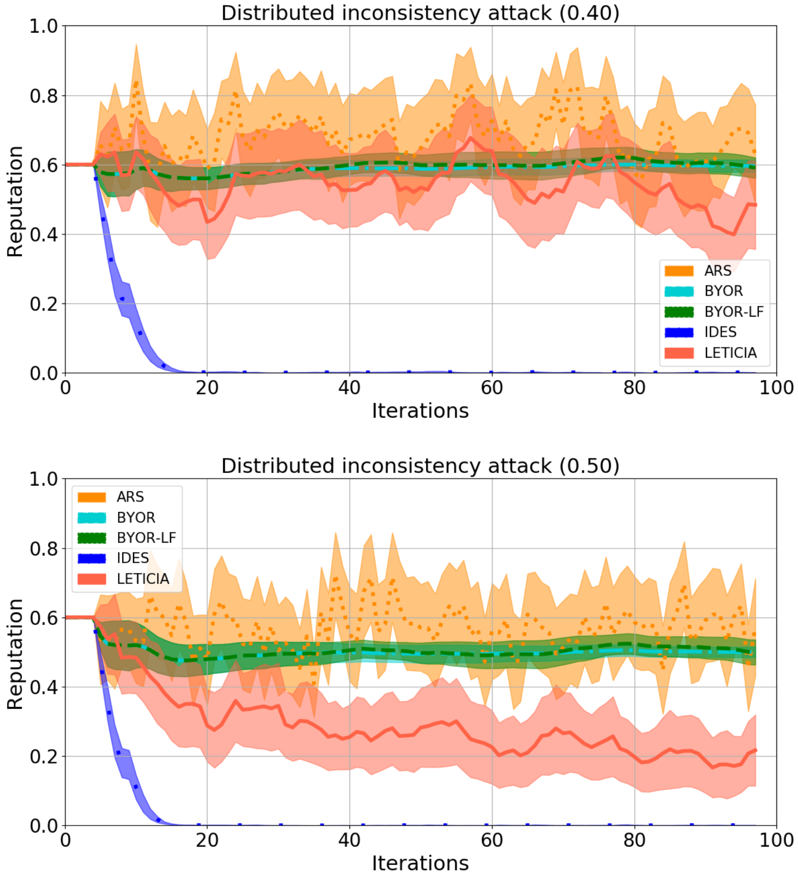


Source: author's own (2020)

For the distribution of 30% false messages, LETICIA kept the vehicle's reputation below the ARS algorithm. However, BYOR and BYOR-LF still performed better, further lowering the reputation. IDES reduced the vehicle's reputation close to zero after the 20th iteration of the simulation.

Figure 7.5 shows the inconsistency attacks, with rates of 40% and 50% for sending false messages. The LETICIA algorithm obtained the best results, as shown in the 40% and 50% graphs. On the 40% attack chart, LETICIA maintained the vehicle's reputation between 0.6 and 0.4. IDES maintained the previous attack percentage's behavior, leaving the reputation very close to zero just before the 20th iteration. However, LETICIA remains reputable below 0.6 with the possibility of resuming a high reputation in the event of a behavior change.

Figure 7.5 – Inconsistency distributed attack with vehicle sending false messages 40% and 50% of the time, respectively

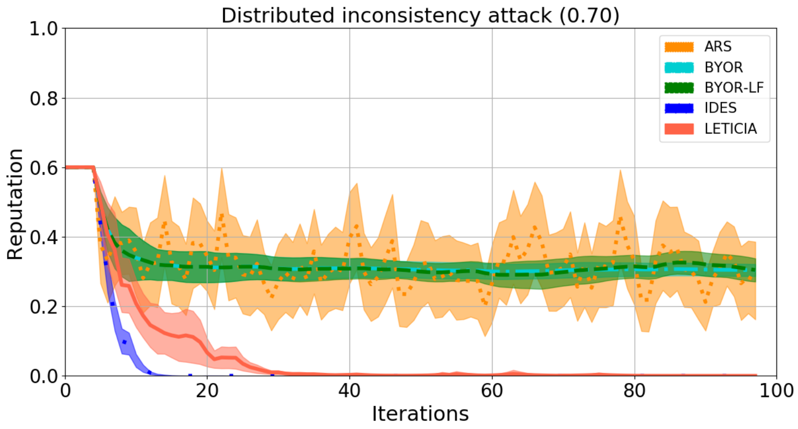


Source: author's own (2020)

On the 50% attack graph, in which the vehicle stays half the time sending fake messages, LETICIA drops the vehicle's reputation even more dramatically in the first iterations, keeping the reputation below 0.4. The LETICIA results were better than the ARS, BYOR, BYOR-LF algorithms. However, IDES maintains its behavior of reducing reputation close to zero in the first iterations.

This behavior tends to remain with higher percentages of attacks, as illustrated by Figure 7.6, which shows a 70% attack in the distributed inconsistency attack. For such an aggressive attack, LETICIA keeps the vehicle's reputation close to zero, drawing with IDES around the 30th iteration.

Figure 7.6 – Inconsistency distributed attack with vehicle sending false messages 70% of the time



Source: author's own (2020)

7.1.1 Bad-mouthing collusion attack

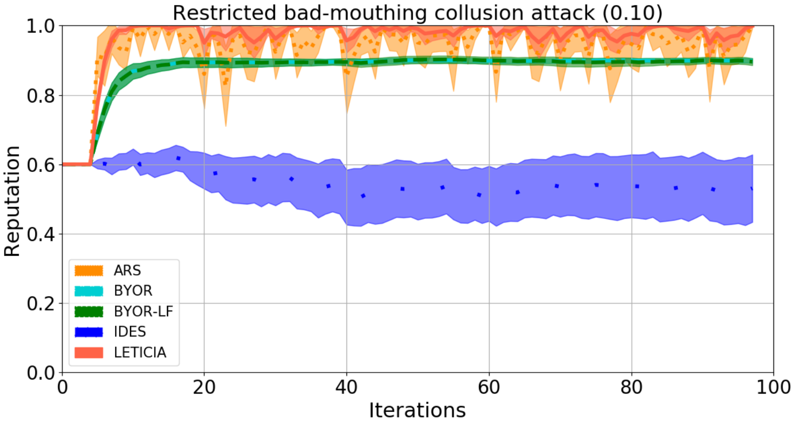
The reputation algorithms were also evaluated in the presence of collusion attacks by bad-mouthing. In this type of attack, malicious vehicles come together to manipulate a particular vehicle's reputation on the network. The consequence is that the other vehicles in the network wrongly evaluate the attacked vehicle, underestimating its reputation. The subgroups of collusion attacks by restricted and distributed bad-mouthing were evaluated.

Restricted bad-mouthing collusion

In this attack, vehicles with a reputation below a specified value send negative feedback about an honest vehicle. For vehicles with reputations below 0.1, the LETICIA algorithm kept the rated vehicle's reputation close to 1.0, unaffected by the bad-mouthing collusion attack, as can be seen in Figure 7.7. IDES was the most affected algorithm, keeping the reputation below 0.6 for almost the entire simulation period.

For vehicle attacks with reputations below 0.3 and 0.4, LETICIA still maintained a better result than the others, keeping the vehicle's reputation high, (FIGURE 7.8). For collusion with a reputation below 0.3, it maintained a reputation above 0.8. For collusion of vehicles with a

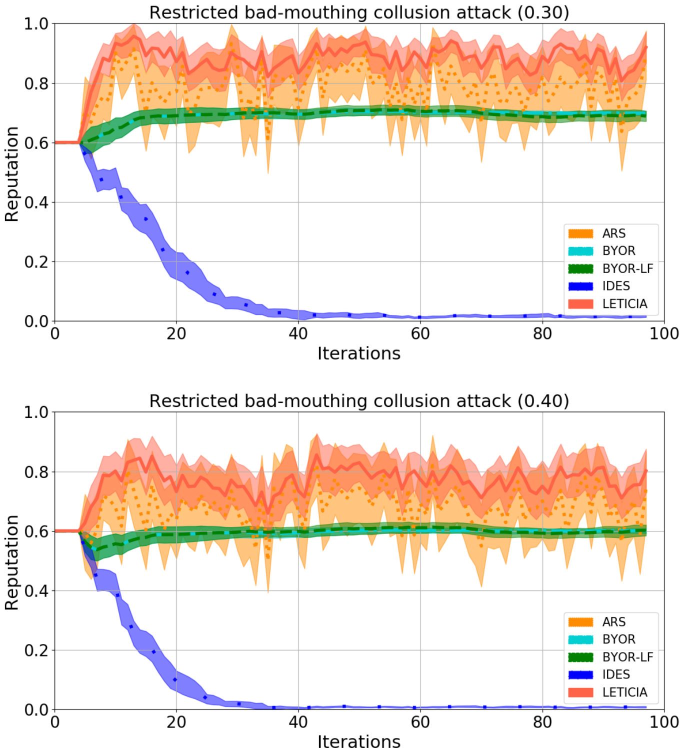
Figure 7.7 – Restricted bad-mouthing collusion attack with vehicles with 0.1 reputation is making attack



Source: author's own (2020)

reputation below 0.4, it maintained a reputation above 0.7. The ARS, BYOR, and BYOR-LF algorithms maintained the vehicle's reputation below LETICIA. IDES maintained the worst result, being affected by the attack.

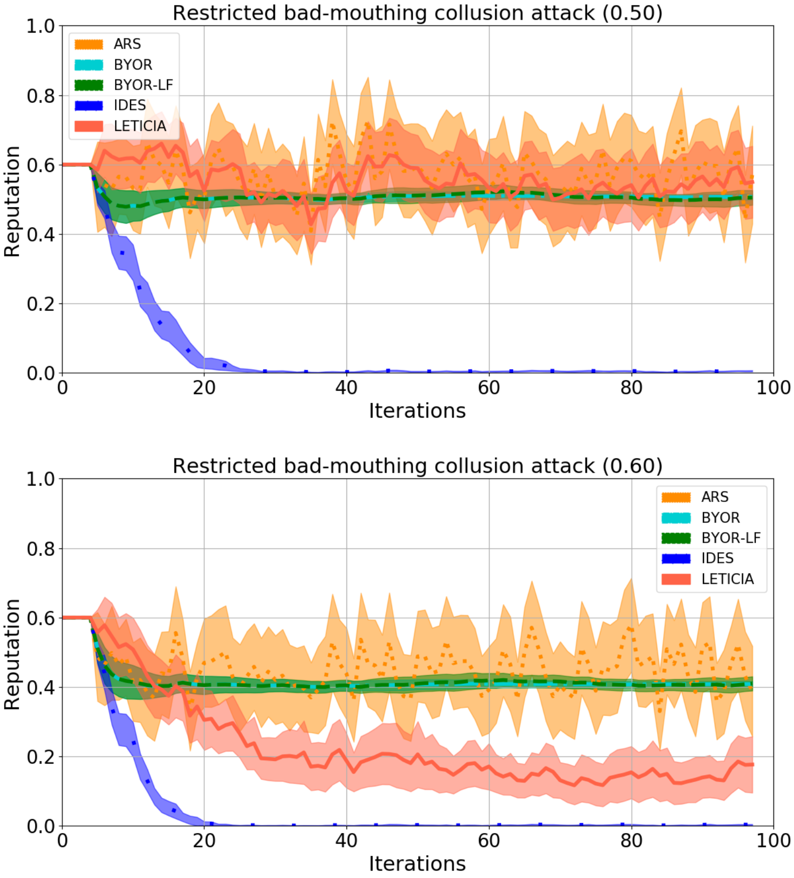
Figure 7.8 – Restricted bad-mouthing collusion attack with vehicles with 0.3 and 0.4 reputation is making attack, respectively



Source: author's own (2020)

The simulation in which the attacker vehicles have reputations below 0.5, LETICIA algorithm maintains the vehicle's reputation above the others. However, ARS was superior to the others, as can be seen in Figure 7.9. Nevertheless, in general, all algorithms maintain the vehicle's reputation below 0.6. From attacks in which vehicles have a reputation below 0.6, LETICIA cannot maintain an honest vehicle reputation, dropping dramatically.

Figure 7.9 – Restricted bad-mouthing collusion attack with vehicles with 0.5 and 0.6 reputation is making attack, respectively


Restricted bad-mouthing collusion attack (0.50)

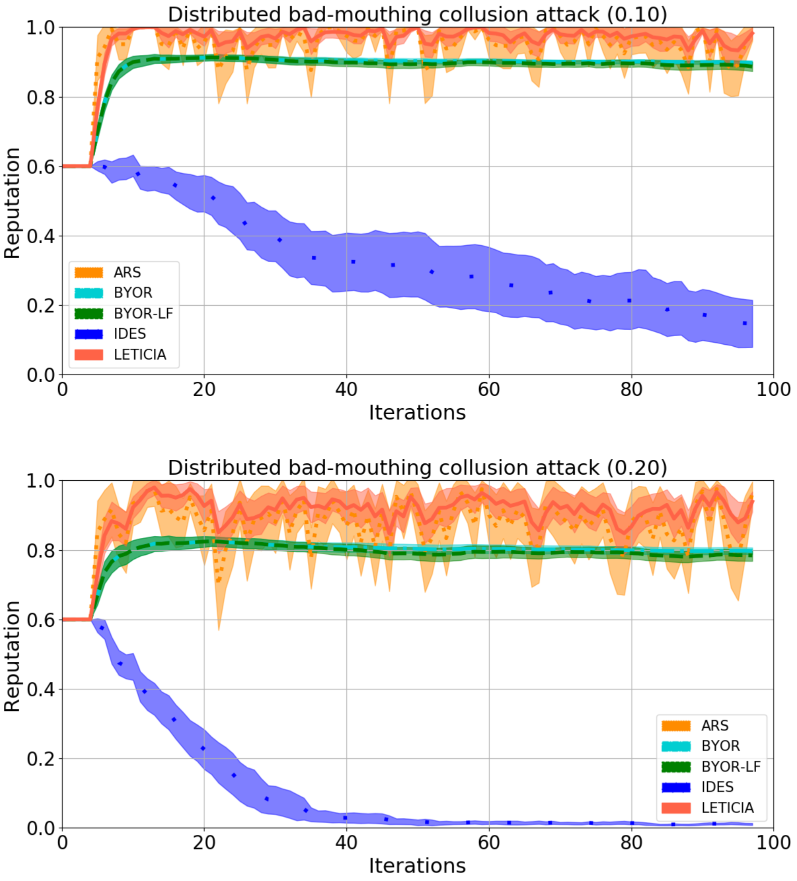
Restricted bad-mouthing collusion attack (0.60)

Source: author's own (2020)

Distributed bad-mouthing collusion

In this attack, vehicles are grouped, regardless of their reputations, and incorrectly send negative feedback following a determined probability distribution rate. Figure 7.10 shows the simulations with rates of 10% and 20 % of vehicles making attacks. For both tests, the LETICIA and ARS algorithms had similar results. LETICIA got a little better on both tests. ARS varied more, with slightly higher declines throughout the simulation. In the 10% attack, BYOR and BYOR-LF left the reputation at approximately 0.9 for almost every simulation. However, IDES got the worst result, quickly reducing the vehicle's reputation, reaching approximately 0.15. In the 20% attack, the LETICIA was around 90%, and the BYOR and BYOR-LF algorithms remained at 0.8. In contrast, IDES dropped its reputation to almost zero around iteration 35.

Figure 7.10 – Distributed bad-mouthing collusion attack with 10% and 20% of the vehicles making attack, respectively



Source: author's own (2020)

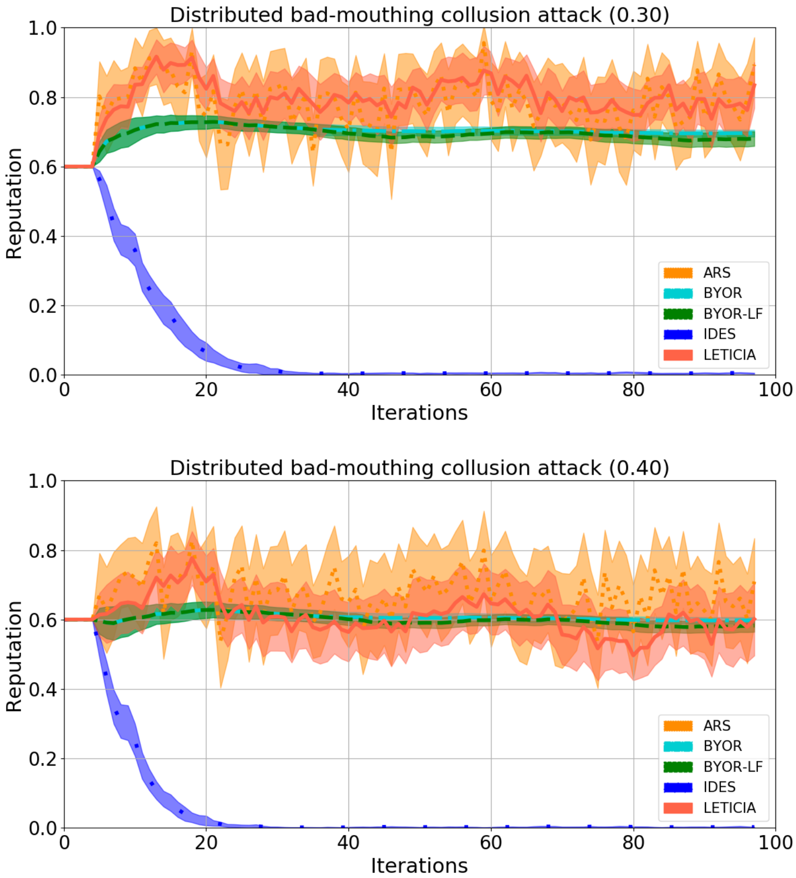
The Figure 7.11 shows the distributed bad-mouthing collusion attacks with 30% and 40% of the vehicles performing attacks, respectively. The first graph shows the same behavior of the 10% and 20% tests, in which LETICIA and ARS maintain a similar behavior, but in general, LETICIA remains slightly higher, keeping the reputation around 0.8. The BYOR and BYOR-LF algorithms continue with practically the same behavior, maintaining the vehicle's reputation around 0.7. In contrast, IDES continues to prove ineffective against this type of attack, practically zeroing in on reputation around the 20th iteration.

In the second graph, 40% of the vehicles are making attacks. The BYOR, BYOR-LF, and LETICIA algorithms were practically tied, maintaining the reputation and around 0.6. In this scenario, the ARS achieved a slightly better result, as it managed to maintain its reputation above 60%. IDES maintained its behavior, drastically reducing the reputation of the vehicle being attacked.

Finally, the graphics of Figure 7.12 show the behavior of the vehicle's reputation that sends messages suffering collusion attacks by distributed bad-mouthing, with rates of 50% and 60%. For the 50% attacker rate, all algorithms reduced the honest vehicle's reputation to values below the initial reputation, which was 0.6. The ARS maintained the reputation around 0.55, and the BYOR and BYOR-LF algorithms maintained the reputation at approximately 0.5.

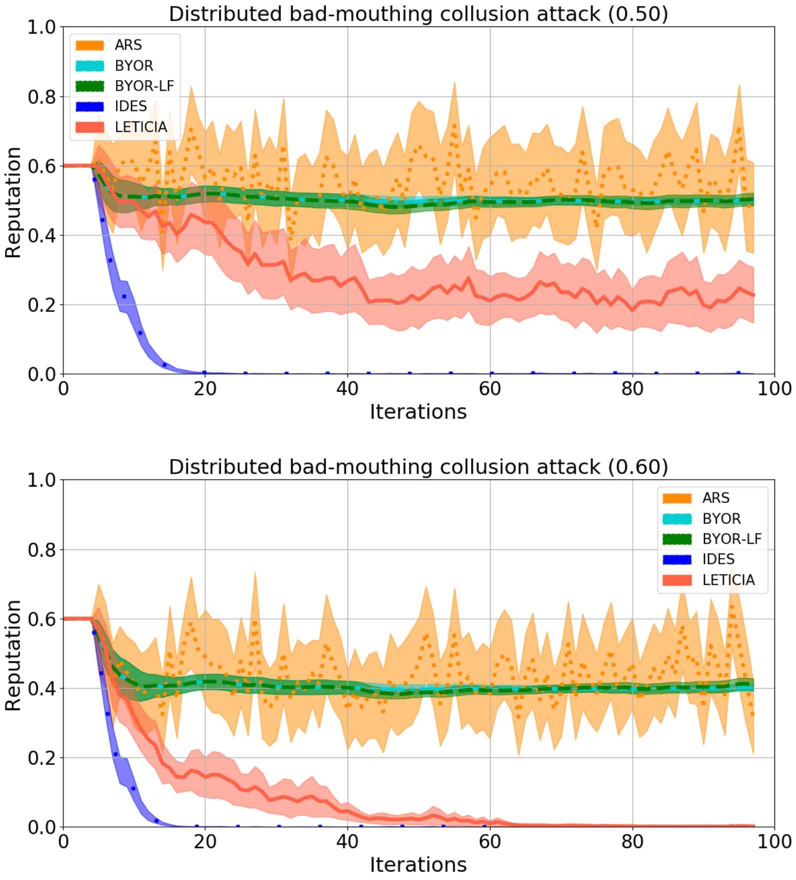
However, in this more aggressive scenario, where half of all vehicles in the network were attacking the vehicle that sent true messages, LETICIA reduced the honest vehicle's reputation to values below the other algorithms, below 0.3. It is justified since half the vehicles in the network are attackers. This behavior is in line with expectations since we assume in this work that most vehicles in the network are honest. This hypothesis has been used in VANET as shown (GHALEB et al., 2019). For even more aggressive attacks, as shown in the 60 % graph, the algorithm cannot sustain itself and brings the reputation close to zero in iteration 60, for the same reason described previously.

Figure 7.11 – Distributed bad-mouthing collusion attack with 30% and 40% of the vehicles making attack, respectively



Source: author's own (2020)

Figure 7.12 – Distributed bad-mouthing collusion attack with 50% and 60% of the vehicles making attack, respectively



Source: author's own (2020)

8 CONCLUSIONS

The advance of wireless communication between devices has increased the number of connected devices. It triggered several security problems. However, in VANET, this insecurity brings several inconveniences to drivers and can even be fatal. This work presented a vehicular network architecture that improves the security and reliability of the messages exchanged in the network through a digital signature system and a centralized vehicle reputation.

The proposed architecture, DVNAT, was used to compare and evaluate the reputation algorithms ARS, BYOR, BYOR-LF, IDES and LETICIA, against inconsistency and collusion attacks by bad-mouthing. The results show that for inconsistency and bad-mouthing collusion attacks, IDES keeps the vehicle's reputation always close to zero, not allowing a resumption in case of behavior change and showing to be utterly inefficient against collusion attacks by bad-mouthing. In the restricted inconsistency attack, the LETICIA algorithm was better because it followed the vehicle's behavior, reducing and increasing its reputation as expected, remaining less time with the reputation up. However, in distributed inconsistency attacks, LETICIA does better in more aggressive attacks, starting at 40% attacks.

For restricted bad-mouthing collusion attacks, low-reputation vehicles making attacks do not contribute much to reduce the honest vehicle's reputation when using LETICIA. Considering that most of the network's nodes are honest, in the case of bad-mouthing collusion attacks, LETICIA also did better. Even when 40% of the network makes attacks against the vehicle, it manages to maintain its reputation above the other algorithms. Therefore, this work showed the superiority of the proposed reputation algorithm, taking into account the inconsistency and bad-mouthing collusion attacks, in a robust architecture.

The LETICIA algorithm is effective against other types of reputation attacks, such as false information and betrayal, for example. DVNAT architecture can mitigate other types of attacks like impersonation, Sybil, and newcomer. If you need to mitigate attacks like eavesdropping (espionage) and MITM, just use encryption on critical messages. In this way, both (DVNAT and LETICIA) are very effective against various types and attacks documented in vehicular networks and can be applied in a variety of configurations, from critical traffic applications to information exchange applications as points of interest and entertainment for passengers.

8.1 Future works

As future work, we intend to expand the structure of DVNAT to allow the creation of modules containing other reputable algorithms so that they can be easily tested. It is also intended to investigate DVNAT and the tested algorithms:

- in a simulated scenario with a real trace;
- with several vehicles sending messages and being evaluated;
- with other types of reputation attacks, such as betrayal for example;
- using other metrics, such as response time and message loss, for example.

Finally, it is planned to validate the entire architecture (DVNAT and LETICIA) in a real environment, containing: vehicles, RSUs, infrastructure servers interacting through wireless communication, and all variables in the real world.

REFERENCES

AHMAD, F. et al. Man-in-the-middle attacks in vehicular ad-hoc networks: Evaluating the impact of attackers' strategies. **Sensors**, v. 18, n. 11, 2018. ISSN 1424-8220. Disponível em: <http://www.mdpi.com/1424-8220/18/11/4040>.

AHMED, S. A. M.; ARIFFIN, S. H. S.; FISAL, N. Overview of wireless access in vehicular environment (wave) protocols and standards. **Indian Journal of Science and Technology**, Vol 6 (7), p. 4994–5001, jul 2013. ISSN 0974-5645. 34355-35026-1-PB. Disponível em: <http://www.indjst.org/index.php/indjst/article/viewFile/34355/27974>.

ALNASSER, A.; SUN, H.; JIANG, J. Cyber security challenges and solutions for v2x communications: A survey. **Computer Networks**, v. 151, p. 52 – 67, 2019. ISSN 1389-1286. ALNASSER201952. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1389128618306157>.

AYUSHI. Article: A symmetric key cryptographic algorithm. **International Journal of Computer Applications**, v. 1, n. 14, p. 1–4, February 2010. Published By Foundation of Computer Science.

BANKOVIĆ, Z. et al. Detecting bad-mouthing attacks on reputation systems using self-organizing maps. In: HERRERO, Á.; CORCHADO, E. (Ed.). **Computational Intelligence in Security for Information Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. p. 9–16. ISBN 978-3-642-21323-6. 10.1007/978-3-642-21323-6_2.

BAO, S. et al. A lightweight authentication and privacy-preserving scheme for vanets using tesla and bloom filters. **ICT Express**, 2017. ISSN 2405-9595. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2405959517302333>.

BEHRISCH, M. et al. Sumo - simulation of urban mobility: An overview. In: **in SIMUL 2011, The Third International Conference on Advances in System Simulation**. [S.l.: s.n.], 2011. p. 63–68. Behrisch11sumo-.

BERNSTEIN, D. J. et al. High-speed high-security signatures. **Journal of Cryptographic Engineering**, v. 2, n. 2, p. 77–89, Sep 2012. ISSN 2190-8516. Bernstein2012. Disponível em: <https://doi.org/10.1007/s13389-012-0027-1>.

BITTL, S. et al. Emerging attacks on vanet security based on gps time spoofing. In: **2015 IEEE Conference on Communications and Network Security (CNS)**. [S.l.: s.n.], 2015. p. 344–352. 7346845.

Brendha, R.; Prakash, V. S. J. A survey on routing protocols for vehicular ad hoc networks. In: **2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)**. [S.l.: s.n.], 2017. p. 1–7.

CAVIN, D.; SASSON, Y.; SCHIPER, A. On the accuracy of manet simulators. In: **Proceedings of the Second ACM International Workshop on Principles of Mobile Computing**. New York, NY, USA: ACM, 2002. (POMC '02), p. 38–43. ISBN 1-58113-511-4. Cavin:2002:AMS:584490.584499. Disponível em: <http://doi.acm.org/10.1145/584490.584499>.

CERVANTES, C. et al. Um sistema de detecção de ataques sinkhole sobre 6lowpan para internet das coisas. **XIV Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais**, p. 153–166, 2014.

DOTZER, F.; FISCHER, L.; MAGIERA, P. Vars: a vehicle ad-hoc network reputation system. In: **Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks**. [S.l.: s.n.], 2005. p. 454–456. 1443536.

EICHLER, S. Performance evaluation of the ieee 802.11p wave communication standard. In: **2007 IEEE 66th Vehicular Technology Conference**. [S.l.: s.n.], 2007. p. 2199–2203. ISSN 1090-3038.

Engoulou, R. G. et al. A decentralized reputation management system for securing the internet of vehicles. In: **2019 International Conference on Computing, Networking and Communications (ICNC)**. [S.l.: s.n.], 2019. p. 900–904.

GHALEB, F. A. et al. Context-aware data-centric misbehaviour detection scheme for vehicular ad hoc networks using sequential analysis of the temporal and spatial correlation of the consistency between the cooperative awareness messages. **Vehicular Communications**, Elsevier, v. 20, 2019.

GRäFLING, S.; MäHöNEN, P.; RIIHIJäRVI, J. Performance evaluation of ieee 1609 wave and ieee 802.11p for vehicular communications. In: **2010 Second International Conference on Ubiquitous and Future Networks (ICUFN)**. [S.l.: s.n.], 2010. p. 344–348. ISSN 2165-8528. 5547184.

HARTENSTEIN, H.; LABERTEAUX, L. P. A tutorial survey on vehicular ad hoc networks. **IEEE Communications Magazine**, v. 46, n. 6, p. 164–171, June 2008. ISSN 0163-6804.

HASROUNY, H. et al. Vanet security challenges and solutions: A survey. **Vehicular Communications**, v. 7, p. 7 – 20, 2017. ISSN 2214-2096. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2214209616301231>.

HUSSAIN, R. et al. A hybrid trust management framework for vehicular social networks. In: **Internat. Conference on Computational Social Networks**. [S.l.: s.n.], 2016. p. 214–225.

IANIX. **Things that use Ed25519**. 2020. <https://ianix.com/pub/ed25519-deployment.html>.

IEEE1609.0. Ieee draft guide for wireless access in vehicular environments (wave) - architecture. **IEEE P1609.0/D9, July 2017**, p. 1–104, Jan 2017. 7982731.

IEEE1609.2a. Ieee standard for wireless access in vehicular environments–security services for applications and management messages - amendment 1. **IEEE Std 1609.2a-2017 (Amendment to IEEE Std 1609.2-2016)**, p. 1–123, Oct 2017.

JAIMES, L. M. S.; ULLAH, K.; MOREIRA, E. dos S. Ars: Anonymous reputation system for vehicular ad hoc networks. In: **2016 8th IEEE Latin-American Conference on Communications (LATINCOM)**. [S.l.: s.n.], 2016. p. 1–6. 7811600.

JOHNSON, D.; MENEZES, A.; VANSTONE, S. The elliptic curve digital signature algorithm (ecdsa). **International Journal of Information Security**, v. 1, n. 1, p. 36–63, Aug 2001. ISSN 1615-5262. Johnson2001. Disponível em: <https://doi.org/10.1007/s102070100002>.

JøSANG, A.; ISMAIL, R.; BOYD, C. A survey of trust and reputation systems for online service provision. **Decision Support Systems**, v. 43, n. 2, p. 618 – 644, 2007. ISSN 0167-9236. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167923605000849>.

KALRA, S.; SOOD, S. K. Elliptic curve cryptography: Survey and its security applications. In: **Proceedings of the International Conference on Advances in Computing and Artificial Intelligence**. New York, NY, USA: ACM, 2011. (ACAI '11), p. 102–106. ISBN 978-1-4503-0635-5. Kalra:2011:ECC:2007052.2007073. Disponível em: <http://doi.acm.org/10.1145/2007052.2007073>.

KCHAOU, A.; ABASSI, R.; GUEMARA, S. Toward a distributed trust management scheme for vanet. In: **Proceedings of the 13th International Conference on Availability, Reliability and Security**. New York, NY, USA: ACM, 2018. (ARES 2018), p. 53:1–53:6. ISBN 978-1-4503-6448-5. Kchaou:2018:TDT:3230833.3232824. Disponível em: <http://doi.acm.org/10.1145/3230833.3232824>.

KUMAR, S. S. **Elliptic Curve Cryptography for Constrained Devices**. Tese (Doutorado) — Ruhr-University Bochum, 2006. Disponível em: <http://www-brs.ub.ruhr-uni-bochum.de/netahtml/HSS/Diss/KumarSandeepS/diss.pdf>.

LI, X. et al. Rgte: A reputation-based global trust establishment in vanets. In: **2013 5th International Conference on Intelligent Networking and Collaborative Systems**. [S.l.: s.n.], 2013. p. 210–214. 6630411.

LOPEZ, P. A. et al. Microscopic traffic simulation using sumo. In: **The 21st IEEE International Conference on Intelligent Transportation Systems**. IEEE, 2018. Disponível em: <https://elib.dlr.de/124092/>.

MANVI, S. S.; KAKKASAGERI, M. S.; ADIGA, D. G. Message authentication in vehicular ad hoc networks: Ecdsa based approach. In: **2009 International Conference on Future Computer and Communication**. [S.l.: s.n.], 2009. p. 16–20. 5189734.

MISHRA, R.; SINGH, A.; KUMAR, R. Vanet security: Issues, challenges and solutions. In: **2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)**. [S.l.: s.n.], 2016. p. 1050–1055.

MOKHTAR, B.; AZAB, M. Survey on security issues in vehicular ad hoc networks. **Alexandria Engineering Journal**, v. 54, n. 4, p. 1115 – 1126, 2015. ISSN 1110-0168. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1110016815001246>.

MüHLBAUER, R.; KLEINSCHMIDT, J. H. Bring your own reputation: A feasible trust system for vehicular ad hoc networks. **Journal of Sensor and Actuator Networks**, v. 7, n. 3, 2018. ISSN 2224-2708. Jsan7030037. Disponível em: <http://www.mdpi.com/2224-2708/7/3/37>.

NATIVIDADE, D.; CORREIA, L.; SANTOS, A. Um algoritmo de reputação centralizado para redes veiculares contra ataques de inconsistência e bad-mouthing. In: **SBSeg - Simpósio Brasileiro de Segurança da Informação e Sistemas Computacionais**. [S.l.: s.n.], 2020.

NATIVIDADE, D.; CORREIA, L. H. Avaliação de algoritmos de assinatura digital em redes veiculares utilizando ambiente emulado. In: **Anais do XXV Workshop de Gerência e Operação de Redes e Serviços**. Porto Alegre, RS, Brasil: SBC, 2020. p. 181–194. ISSN 2595-2722. Disponível em: <https://sol.sbc.org.br/index.php/wgrs/article/view/12460>.

PEDROSO, C. et al. Mitigação de Ataques IDFs no Serviço de Agrupamento de Disseminação de Dados em Redes IoT Densas. In: **Anais SBSeg 2019**. Porto Alegre, RS, Brasil: SBC, 2019. Disponível em: <https://sol.sbc.org.br/index.php/sbseg/article/view/4254>.

PERBAWA, M. R.; AFRYANSYAH, D. I.; SARI, R. F. Comparison of ecdsa and rsa signature scheme on nlsr performance. In: **2017 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob)**. [S.l.: s.n.], 2017. p. 7–11. 8284007.

PERRIG, A. et al. **Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction**. [S.l.], 2005. 1-22 p. Disponível em: <https://www.rfc-editor.org/rfc/rfc4082.txt>.

RAVI, K.; KULKARNI, S. A. A secure message authentication scheme for vanet using ecdsa. In: **2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)**. [S.l.: s.n.], 2013. p. 1–6. 6726769.

RIVEST, R. L.; SHAMIR, A.; ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. **Commun. ACM**, ACM, New York, NY, USA, v. 21, n. 2, p. 120–126, fev. 1978. ISSN 0001-0782. Rivest:1978:MOD:359340.359342. Disponível em: <http://doi.acm.org/10.1145/359340.359342>.

ROSELINMARY, S.; MAHESHWARI, M.; THAMARAISELVAN, M. Early detection of dos attacks in vanet using attacked packet detection algorithm (apda). In: **2013 International Conference on Information Communication and Embedded Systems (ICICES)**. [S.l.: s.n.], 2013. p. 237–240. 6508250.

RSA LABORATORIES. **Answers to Frequetly Asked Question About today's Crytography**. 100 Marine Parkway, Suite 500, Redwood City, CA 94065-1031 USA, 1993. 1-204 p. Disponível em: <http://www.rsa.com/rsalabs/>.

RSA LABORATORIES. **RSA Laboratories' Frequently Asked Questions About Today' s Cryptography, Version 4.1**. 20 Crosby Drive, Bedford, MA 01730 USA, 2000. 1-269 p. Disponível em: <http://www.nordugrid.org/documents/rsalabs_faq41.pdf>.

Ruohomaa, S.; Kutvonen, L.; Koutrouli, E. Reputation management survey. In: **The Second International Conference on Availability, Reliability and Security (ARES'07)**. [S.l.: s.n.], 2007. p. 103–111.

SAKHRELIYA, S. C.; PANDYA, N. H. Pki-sc: Public key infrastructure using symmetric key cryptography for authentication in vanets. In: **2014 IEEE International Conference on Computational Intelligence and Computing Research**. [S.l.: s.n.], 2014. p. 1–6. 7238326.

SHRIVASTAVA, A.; SHARMA, K.; CHAURASIA, B. K. Hmm for reputation computation in vanet. In: **2016 International Conference on Computing, Communication and Automation (ICCCA)**. [S.l.: s.n.], 2016. p. 667–670. 7813806.

SINGH, K. et al. Authentication and privacy preserving message transfer scheme for vehicular ad hoc networks (vanets). In: **Proceedings of the 12th ACM International Conference on Computing Frontiers**. New York, NY, USA: ACM, 2015. (CF '15), p. 58:1–58:7. ISBN 978-1-4503-3358-0. Disponível em: <http://doi.acm.org/10.1145/2742854.2745718>.

SOMMER, C. **VEINS: Further publications**. 2019. Disponível em: <https://veins.car2x.org/publications/>.

SOMMER, C.; GERMAN, R.; DRESSLER, F. Bidirectionally coupled network and road traffic simulation for improved ivc analysis. **IEEE Transactions on Mobile Computing**, v. 10, n. 1, p. 3–15, Jan 2011. ISSN 1536-1233. 5510240.

SPAHO, E. et al. Vanet simulators: A survey on mobility and routing protocols. In: **2011 International Conference on Broadband and Wireless Computing, Communication and Applications**. [S.l.: s.n.], 2011. p. 1–10. 6103008.

Su, S. et al. A Reputation Management Scheme for Efficient Malicious Vehicle Identification over 5G Networks. **IEEE Wireless Communications**, v. 27, n. 3, p. 46–52, 2020.

TRČEK, D. **Trust and Reputation Management Systems: An e-Business Perspective**. Gewerbestrasse 11, 6330 Cham, Switzerland: Springer International Publishing, 2017. (SpringerBriefs in Information Systems). ISBN 9783319623740.

TURAN, F.; VERBAUWHEDE, I. Compact and flexible fpga implementation of ed25519 and x25519. **ACM Trans. Embed. Comput. Syst.**, ACM, New York, NY, USA, v. 18, n. 3, p. 24:1–24:21, abr. 2019. ISSN 1539-9087. Turan:2019:CFF:3323876.3312742. Disponível em: <http://doi.acm.org/10.1145/3312742>.

VARGA, A. The omnet++ discrete event simulation system. In: **In ESM'01**. [s.n.], 2001. Varga01theomnet++. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.331.1728>.

VARGA, A. Omnet++. In: ____. **Modeling and Tools for Network Simulation**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 35–59. ISBN 978-3-642-12331-3. Varga2010. Disponível em: <https://doi.org/10.1007/978-3-642-12331-3_3>.

WANG, J. et al. A survey of vehicle to everything (v2x) testing. **Sensors**, v. 19, n. 2, 2019. ISSN 1424-8220. S19020334. Disponível em: <http://www.mdpi.com/1424-8220/19/2/334>.

WANG, J. et al. Rprep: A robust and privacy-preserving reputation management scheme for pseudonym-enabled vanets. **International Journal of Distributed Sensor Networks**, v. 12, n. 3, p. 6138251, 2016. Disponível em: <https://doi.org/10.1155/2016/6138251>.

WANG, S.; YAO, N. Liap: A local identity-based anonymous message authentication protocol in vanets. **Computer Communications**, v. 112, p. 154 – 164, 2017. ISSN 0140-3664. WANG2017154. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366417309520>.

WASEF, A.; SHEN, X. Emap: Expedite message authentication protocol for vehicular ad hoc networks. **IEEE Transactions on Mobile Computing**, v. 12, n. 1, p. 78–89, Jan 2013. ISSN 1536-1233. 6081877.

World Heath Organization. **Road traffic injuries**. 2020. WHO. Disponível em: <https://www.who.int/en/news-room/fact-sheets/detail/road-traffic-injuries>.

ZHANG, J. A survey on trust management for vanets. In: **2011 IEEE International Conference on Advanced Information Networking and Applications**. [S.l.: s.n.], 2011. p. 105–112. ISSN 2332-5658.