TÉCNICO LISBOA

A secure content dissemination scheme for the Internet-of-Vehicles

Rodrigo Manuel de Almeida Colaço

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors:

Prof. Paulo Rogério Barreiros D'Almeida Pereira

Prof. Naércio David Pedro Magaia

Examination Committee:

Chairperson: Profª Teresa Maria Sá Ferreira Vazão Vasques

Supervisor: Prof. Paulo Rogério Barreiros D'Almeida Pereira

Member of the Committee: Prof. Miguel Nuno Dias Alves Pupo Correia

November 2021

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

My eternal gratitude to my supervisors Prof. Paulo Pereira and Prof. Naércio Magaia, for their noble, generous, and invaluable scientific support in this research process. Without their supervision, this work would not have been finished. I am very grateful for their permanent guidance and encouragement throughout this long walk with all its “ups and downs.” I sincerely appreciate their careful revision of all my work. I also would like to thank their friendship. This work is part of the SEEDS project (H2020-MSCA-RISE under grant No 101006411).

I wish to express my love and gratitude to my family for their constant support and giving me the strength, stability, and optimism during this stage of my studies, which was very demanding in all its dimensions.

A special thanks to all my professors and professors worldwide for teaching and believing in students. I have been fortunate to encounter extraordinary professors who always pushed me to go further and have confidence in myself.

Finally, I show gratitude to the Scientific Community for its vital contributions and sharing in various research fields. I hope this work I proudly share may be a worthy input to motivate other fellow researchers to continue exploring and building the next-generation IoV.

Abstract

This work presents to the reader an incursion into the world of the Internet of Vehicles (IoV), its possibilities, requirements, and challenges ahead. The IoV is conceptually studied, and data security is introduced as a primary necessity.

Generally defined, an IoV is leveraged by cooperative multi-hop forwarding enabled schemes to propagate the messages between the nodes, continuously working toward safe, robust, fair, and efficient communications. Thus, the IoV is an integrated network that supports intelligent road traffic management, intelligent dynamic information service, and intelligent vehicle control, demonstrating a perfect example of the application of Internet-of-Things technology in Intelligent Transportation Systems.

This work's main objective is to simulate message dissemination techniques in the IoV with and without cryptographic overheads imposed by cryptosystems. In this way, we will be in conditions to conclude about the impact of cryptosystems' overhead when applied to the IoV.

Specifically, this dissertation selects and describes the NTRU Re-Encrypt cryptosystem as the best candidate to integrate the IoV dissemination schemes. These schemes are implemented in the NS3 network simulator, taking into account different vehicle densities, multi-hop capabilities, and cryptosystem overheads (i.e., processing time and data length).

In the end, we analyze the statistics provided by the simulations and conclude about the practical use, impact, and integration of the NTRU ReEncrypt cryptosystem in the IoV.

Furthermore, and in addition to the IoV simulations, this work offers insightful information about the proxy re-encryption cryptosystems' history and motivations.

Keywords

Internet of Vehicles, Ad-Hoc Networks, Proxy Re-Encryption, NTRU, Crypto-Security.

Resumo

Este trabalho pretende levar o leitor numa viagem ao mundo da Internet dos Veículos (IoV), das suas possibilidades, requisitos e desafios. O conceito de IoV é estudado e o requisito de segurança dos dados é introduzido como uma necessidade primaria.

Geralmente definida, a IoV é levada ao máximo das suas capacidades através de esquemas cooperativos e habilitados de multi-hop para reencaminhamentos de mensagens na rede, entre os nós que a compõem e cuidam de forma a conseguir comunicações mais seguras, robustas, justas e eficientes. A IoV é uma rede integrada que suporta a gestão inteligente do tráfego rodoviário, serviços inteligentes de informação dinâmica, e controlo inteligente dos veículos, demonstrando assim que é um exemplo perfeito da aplicação da tecnologia da Internet-das-Coisas nos Sistemas Inteligentes de Transportes.

O principal objetivo deste trabalho é o de simular várias técnicas de disseminação de mensagens na IoV, com e sem os custos suplementares impostos pelos cripto sistemas. Desta forma, estaremos em condições de concluir acerca do impacto dos cripto sistemas quando aplicados na IoV.

Especificamente, nesta dissertação selecionamos e descrevemos o cripto sistema NTRU Re-Encrypt como o melhor candidato a integrar nos esquemas de disseminação para a IoV. Estes esquemas são implementados no simulador de redes NS3, tomando em conta diferentes densidades de veículos, capacidades de multi-hop, e custos associados ao cripto sistema (i.e., tempo de processamento e tamanho dos dados).

Por fim, fazemos uma análise das estatísticas fornecidas pelas simulações e concluímos acerca do uso prático do cripto sistema NTRU Re-Encrypt, assim como do seu impacto na integração de redes de IoV.

Para além das simulações de IoV, este trabalho oferece informação esclarecedora acerca dos cripto sistemas de proxy re-encryption, assim como a sua história e motivações.

Palavras Chave

Internet dos Veículos, Redes Ad-Hoc, Proxy Re-Encryption, Simulador de Redes, Cripto-Segurança.

Index

1. Introduction ... 1

1.1 Context and motivation ... 1

1.2 Objectives ... 2

1.3 Structure of the Thesis ... 2

2. State of the Art ... 4

2.1 Internet-of-Vehicles Architecture ... 4

2.1.1 Vehicle-to-everything ... 6

2.1.2 IoV Wireless Technologies ... 8

2.2 Secure Data Dissemination Schemes ... 9

2.2.1 Data Security ... 9

2.2.2 Proxy Re-Encryption ... 11

2.2.3 1998-2004: Embryonic Stage ... 12

2.2.4 2005-2013: Definition Stage ... 13

2.2.5 2014-2019: Innovation Stage ... 18

2.2.6 2020-Present: Quantum-Safe Stage ... 21

2.2.7 PRE Scheme Comparisson ... 22

2.3 Software Simulation Tools for the IoV ... 25

2.3.1 Performance Evaluation ... 25

2.3.2 Performance Metrics ... 27

2.3.3 Simulator Selection ... 28

2.3.4 NS3 Network Simulator ... 29

3. NTRU PRE Scheme ... 30

3.1 NTRUReEncrypt ... 30

3.1.1 History and Motivation ... 30

3.1.2 Algorithm Specifications and Limitations ... 31

3.1.3 Standard Parameter Sets ... 33

3.1.4 Algorithm's Performance ... 35

3.1.5 Message and Ciphertext Specifications ... 36

3.1.6 Space Cost Analysis ... 37

3.1.7 Re-encryption Limitation ... 38

3.2 NTRUReEncrypt Software Implementation ... 39

4. Network Architecture and Implementation ... 41

4.1 NS3 Architecture Overview ... 41

4.2 WAVE Module ... 42

4.3 C-V2X mmWave Module 44
4.4 Collecting Statistics 44
4.5 Code Implementation Details 48
4.5.1 Packet Processing Time 48
4.5.2 Packet Size 49
4.5.3 Packet Hop Limit 50
5. Performance Evaluation 51
5.1 Simulation Models 51
5.1.1 The IoV Scenario 51
5.2 Results and Discussion 54
6. Conclusions 64
6.1 Achievements and Contributions 64
6.2 Future Work 64
References 65

List of Figures

Figure 1: IoV Architecture Overview 5
Figure 2: V2X communication modes. In order: vehicle-to-vehicle (V2V) , vehicle-to-infrastructure (V2I), vehicle-to-sensor (V2S), vehicle-to-pedestrian (V2P), vehicle-to-roadside unit (V2R). Image extracted from [16] 6
Figure 3: Taxonomy of applications for the IoV. Image extracted from [16]. 8
Figure 4: Classification of various PRE Schemes 12
Figure 5: Proxy Re-encryption workflow 15
Figure 6: Performance evaluation process of OppNets 26
Figure 7: NTRUReEnc Parameter Sets 34
Figure 8: Encryption operation binary to trinary (left side)/ Decryption operation binary to trinary (right side) 37
Figure 9: Re-encryption success rate. 39
Figure 10: NTRUReEnc.java and Package Hierarchy 40
Figure 11: NS3 Key Components 42
Figure 12: WAVE Module architecture 43
Figure 13: mmWave Module schematic, extracted from [120]. 44
Figure 14: Statistics tracking components. 45
Figure 15: Flow Monitor 5-tuple flow 46
Figure 16: Flow Monitor Statistics per-flow 46
Figure 17: WireShark Packet inspection 47
Figure 18: Node-1 AODV Routing Table 48
Figure 19: “udp-l4-protocol.cc” - Send Packet function 49
Figure 20:" onoff-application.cc" - Send Packet func 50
Figure 21: Scenario 1 RWP Map 54

List of Tables

Table 1: Security requirements for the IoV.

Table 2: Property comparison of PRE schemes.

Table 3: Application Examples for Motivating and Evaluating OppNets (based on Dede et al. [1])

Table 4: High-level comparison between simulators (extracted from [1]).

Table 5: Named parameter sets.

Table 6: Computation Performance of NTRUReEncrypt.

Table 7: Space costs of NTRUReEncrypt (extracted from [2]).

Table 8: Changes between 802.11p and 802.11a Physical Layer.

Table 9: Network Parameters for NS-3 simulation scenario 1.

List of Acronyms

5GAA: 5G Automotive Association
ABE: Attribute-based Encryption
ACM: Adaptative Corruption Model
ACMnRO: Adaptative Corruption Model without Random Oracles
AES: Advanced Encryption Standard
AODV: Ad-hoc On-demand Distance Vector
API: Application Programmable Interface
BS: Base Station
CA: Certificate Authority
CAS: Collision Avoidance Systems
CB: Cell Broadcast
CBC: Cipher-Block Chaining
CCA1: Chosen-Ciphertext Attack
CCA2: Adaptative Chosen-Ciphertext Attack
CIA: Confidentiality, Integrity, and Availability
CIB: Conditional Identity-based Broadcast
CLE: Certificateless-based Encryption
CPA: Chosen-Plaintext Attack
C-PRE: Conditional PRE
CR: Collusion-Resistant
C-V2X: Cellular-V2X
DDH: decisional Diffie-Hellman
DDoS: Distributed Denial-of-Service
Dir: Directionality
DRM: Digital Rights Management
DSRC: Dedicated Short-Range Communications
DTN: Delay-tolerant Network
ECC: Elliptic-Curve Cryptography
ETSI: European Telecommunications Standards Institute
FC-PRE: Fuzzy Conditional PRE
FFT: Fast Fourier Transform
FMI: Finnish Meteorological Institute
GDPR : General Data Protection Regulation
GPS: Global Positioning System
IBBE: ID-Based Broadcast Encryption
IBE: Identity-Based Encryption
IBPRE2: Identity-Based Proxy Re-Encryption version 2
IEEE: Institute of Electrical and Electronics Engineers
IND: Indistinguishability
IoT: Internet-of-Things
IoV: Internet-of-Vehicles
ITS: Intelligent Transportation Systems
ITU: International Telecommunications Union
KP-ABPRE: Key-Policy Attribute-Based PRE
KPI: Key Performance Indicator
KP-PRE: Key-Private PRE
LTE: Long-Term Evolution
LWE: Learning With Error
MANET: Mobile Ad-hoc Network
MEC : Mobile Edge Computing
METIS: Mobile and wireless communications Enablers for Twenty-twenty Information Society
MitM: Man-in-the-Middle
MK: Master-Key

MLBC: Modified Lattice-Based Cryptography
NAC: Non-Adaptative Corruption
NetDevices: Network Devices
NI: Non-interactive
NT: Non-Transitive
NTRU: Nth Degree Truncated Polynomial Ring Units
OCB: outside of the context of Basic Service Set
OppNet: Opportunistic Network
OSI: Open Systems Interconnection
OSN: Online Social Networks
P2B: Privacy-Preserving Identity-Based Broadcast PRE
PDR: Packet Delivery Ratio
PEKS: PKE with keyword Search
PIST: Pre-Image Sampling Technique
PKE: Public-Key Encryption
PRES: PRE with keyword Search
PSE: Public-key Searchable Encryption
PTB: PRE-Type-Based
PWS: Public Warning System
QoE: Quality-of-Experience
QoS: Quality-of-Service
RCCA: Replayable Chosen-Ciphertext Attack
RK: Re-encryption Key
RKG: Re-encryption Key Generator
RO: Random Oracles
RSA: Rivest-Shamir-Adleman
RSU: Roadside Unit
RWP: RandomWayPoint
SCH: service channel
SCS: Speed Control Systems
SKG: Secret Key Generator
SM: Standard Model
SMS: Short Message Service
SOTA: State-of-the-Art
TBE: Type-Based Encryption
TB-PRE: Time-Based PRE
TCE: Token-Controlled Encryption
TTL: Time to Live
V2I: Vehicle-to-infrastructure
V2P: Vehicle-to-pedestrian
V2R: Vehicle-to-roadside unit
V2S: Vehicle-to-sensor
V2V: Vehicle-to-vehicle
V2X : Vehicle-to-everything
VANET: Vehicular Ad-hoc Network
VEC : Vehicular Edge Computing
VFC : Vehicular Fog Computing
WAVE: Wireless Access in Vehicular Environments
WBSS: WAVE basic service set

1. Introduction

This chapter provides the scope and objectives of this thesis and reveals the motivation and the global context to solve the problem.

1.1 Context and motivation

The evolution of vehicle technology deals primarily with assisting the driver on the road, providing safety and comfort. In the last years, vehicle technology has evolved in an unprecedented manner; modern cars are equipped with hundreds of sensors, not only used in the traditional powertrain, chassis, and body areas, but also in more advanced and critical applications related to media content and x-by-wire systems [3]. Moreover, the recent evolution in communication network technologies led to the emergence of intelligent connected node systems, giving nodes the possibility to communicate. Vehicles are complex systems that can be seen as nodes. At this level, vehicles are singular entities that require communication with each other to securely and efficiently exchange data in the connected network [4]. This type of network fits the Mobile Ad-hoc Network (MANET) approach, consisting of mobile nodes connected wirelessly in a self-configured, self-healing network without a fixed infrastructure. Specifically, we are interested in the Vehicular Ad-hoc Network (VANET), one of the most challenging MANET flavours due to high and unpredictable dynamic topology, frequent disconnections, and life-threatening issues. With the increasing number of vehicles equipped with computing technologies and wireless communication devices, intervehicle communication is a hot topic of research, standardization, and development [5]. VANETs enable a wide range of applications, such as preventing collisions, safety, blind crossing, dynamic route scheduling, real-time traffic condition, monitoring, etc. Despite having huge potential to address safety and efficiency issues, VANETs have not been able to attract commercial interest due to the fact of having a pure Ad-Hoc architecture, unreliable Internet service, incompatibility with personal devices, unavailability of cloud computing, lower accuracy of the services, and cooperative operational dependency of the network. Several vehicular communication applications are mission-critical; consequently, those issues must be addressed.

Moreover, even with the continuous modernization of vehicles and road infrastructure considering safety as a prime goal, the growing traffic casualties worldwide are a severe cause of concern. Reliable vehicular communications are mandatory and play a significant role in reducing traffic casualties. Renowned organizations have predicted considerable growth in the number of on-road vehicles. The expansion opens a significantly challenging but profitable market for connected vehicles [6]. VANETs need to be extended and evolved towards the Internet-of-Vehicles (IoV) to overcome such challenges. An IoV is defined as the real-time exchanged information on roads between vehicles and sensors, vehicles and vehicles, vehicles and road infrastructures, and vehicles and personal devices using different wireless communication technologies. Likewise, the Internet-of-Things (IoT) concept, IoV allows data collecting and information sharing about vehicles, roads, and their surroundings.

Generally defined, an IoV is leveraged by cooperative multi-hop forwarding enabled schemes to propagate the messages between the nodes, continuously working toward safe, robust, fair, and efficient communications.

Furthermore, the IoV includes enhanced data treatment, data computing, and secure data sharing over network platforms. Over such data, network platforms can efficiently manage and supervise vehicles and offer plenty of services such as multimedia and Internet applications, opening numerous opportunities for investors to improve or create new business. Thus, the IoV is an integrated network that supports intelligent road traffic management, intelligent dynamic information service, and intelligent vehicle control, demonstrating a perfect example of the application of IoT technology in Intelligent Transportation Systems (ITS) [7].

1.2 Objectives

One of the most challenging issues that next-generation wireless networks, such as IoV, will face is to provide Quality-of-Service (QoS) and Quality-of-Experience (QoE) guarantees given the high number of multimedia applications that are emerging every day [8]. Additionally, security is a crucial challenge since many security attacks can harmfully impact the network's reliability and user safety. Some examples of common threats to ITS nodes include Distributed Denial-of-Service (DDoS), jamming, Man-in-the-Middle (MitM), Sybil, eavesdropping, broadcast, Black-Hole node, and message tampering [9].

The volume of data required by connected vehicles in IoV will continue to rise. Some appealing content vehicle dissemination applications are of utmost importance for vehicles' correct functioning [10]. In this context, this thesis's main objective is to devise a secure content dissemination scheme leveraging novel cryptographic methods to handle and protect the IoV network from any cyber-attack that would otherwise jeopardize its reliability.

For this purpose, our proposed scheme will be implemented on a network simulator, and its performance will be evaluated.

Furthermore, an exhaustive comparison of State-of-the-Art (SOTA) approaches is made. In this way, we will be able to build more reliable simulations. Finally, improvements are highlighted through accurate analysis and key performance indicators (KPIs).

1.3 Structure of the Thesis

This manuscript is organized into six chapters as follow:

- **Chapter 1**: We present the purposes, structure, and motivation of this research work.
- **Chapter 2**: Introduces the State-of-the-Art. We describe concepts and relevant proposals used to evaluate opportunistic vehicular networks and cryptosystems with re-encryption capabilities that will reference this research work. Moreover, we offer a detailed description and comparison of the current technologies and tools employed in the simulation process.

- **Chapter 3**: Explains in detail our elected cryptosystem, an NTRU-based PRE scheme. A brief motivation overview is given, its performance is evaluated from where we acknowledge its limitations, and finally, we describe its software implementation.
- **Chapter 4**: We scrutinize in dept the network architecture and how it is implemented in the simulator. We investigate how the simulator works, what modules it has, how we can use them in our simulations, and finally, how we can obtain the desired metrics to evaluate the implemented system.
- **Chapter 5**: We explain the proposed simulation scenario and analyze the respective results according to different inputs that change the simulation environment. After, we provide feasible explanations that will help us conclude and understand the crucial characteristics of vehicular networks embedded with cryptosystems.
- **Chapter 6**: Lastly, this chapter offers the conclusions, contributions, and future work.

2. State of the Art

2.1 Internet-of-Vehicles Architecture

At its root, the overall SOTA of vehicular networking was composed of separated technological components with specific purposes and lacking network insight. Legacy vehicular services commenced by exploiting the mobile Short Message Service (SMS) system and Cell Broadcast (CB) technologies as the communication media, exposing passageways for real-time route forecast and warning services through a Public Warning System (PWS). The VARO-service, designed by the Finnish Meteorological Institute (FMI), is an excellent example of such a service, providing SMS weather warnings and route guidance to the end-user devices [11]. The wireless networking concept was already under a comprehensive research effort in telecommunications, particularly the Ad-hoc networking schemes with self-configurable and self-healing features. For this reason, numerous and distinct routing techniques were introduced and scrutinized.

Despite the increasing effort to achieve better communications efficiency and reliability, this is a continuous and evolving research field in security, routing, and wireless technologies. In the past decade, most IoV architectures heavily relied on the promising Wi-Fi-based technologies developed under the Institute of Electrical and Electronics Engineers (IEEE) 802.11 standards. This was a cumbersome approach due to its inherent limitations related to connection efficiency and reliability. Various vehicular network architectures, deploying a wide range of wireless technologies in different schemes, were successfully investigated, elaborated, and assessed. As an example, please consider the Carlink project and its follow-up project, the WiSafeCar (Wireless traffic Safety network between Cars) [12], [13]. The overall intention is to overcome the limitations imposed by communication systems supported by upgraded and integrated wireless technologies.

For this reason, we must study the Dedicated Short-Range Communications (DSRC) standards suite, based on multiple cooperating standards mainly developed by the IEEE. In particular, we focus on the core design aspects of DSRC, which is called Wireless Access in Vehicular Environments (WAVE). WAVE is highlighted in IEEE 1609.1/.2/.3/.4. The DSRC and WAVE standards have been the center of major attention in both research and industrial communities [14].

The distinctive vehicular WAVE system based on the IEEE 802.11p standard is the best example of upgraded Wi-Fi-based technology, widely in use. In addition, several other Wi-Fi-based protocols and standards surfaced to improve and harden the IoV architectures [15].

However, Wi-Fi and other DSRC still pose a significant challenge to ensure bullet-proof IoV architectures. These must be combined with mobile networks such as 4G, LTE (Long-Term Evolution), and 5G [16], [17]. Mobile networks, managed by Radio WAN operators, provide cellular communications with intrinsic reliability, availability, and security. Cellular communications claim to be promising candidates to overcome substantial difficulties verified in Wi-Fi-based architectures. Recently, i.e., in 2021, Oughton et al. [18] evaluated the two existing leading next-generation wireless broadband technologies, i.e., 5G and Wi-Fi 6. They have concluded that cellular and Wi-Fi technologies are more complementary than competitor technologies.

Figure 1 shows the IoV communications on which we will focus our research, considering that the IoV could be a multi-hop wireless network meant to supply several road applications. In the following subsection, we will introduce the communication in detail. Observing Figure 1, we have a clear idea that the main communication scopes are the inter-vehicle (vehicle to vehicle), vehicle-to-roadside (vehicle to any infrastructure), and inter-roadside (infrastructure to infrastructure) [19].

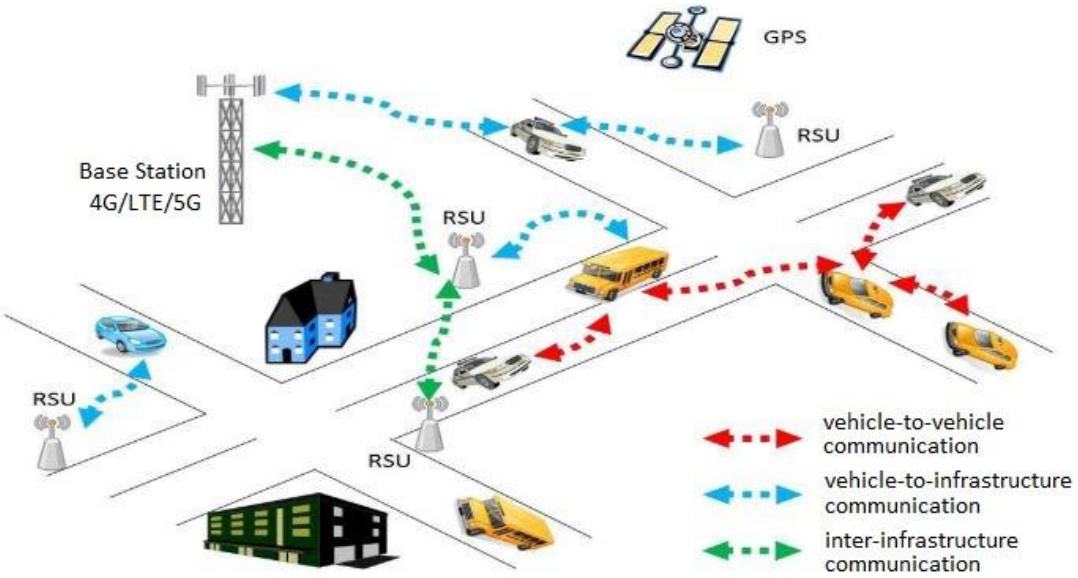

Figure 1: IoV Communication.

Furthermore, the IoV architecture is designed to support and merge three different networks [20]: the inter-vehicular network (from the vehicle to other vehicles), the intravehicular network (within vehicle-cyber-physical elements), and the mobile Internet and cellular network (from the vehicle to other nodes on the network). It is a massively distributed wireless communication system requiring data exchange on a vehicle-to-everything (V2X) communication mode. In this thesis, we refer to modes as the connection types, intuitively specifying the bidirectional connection flow of the exchanged data. Regardless of the technology in use, it must implement these IoV connectivity modes with defined protocols and data interaction standards, like IEEE 802.11p [21]. V2X communication modes are comprehensively detailed, analyzed, and classified in the following subsection. Such assessment is crucial to identify the best match between the different communication modes and the wireless technology to be deployed.

Supported by Magaia et al. [10] research, introducing fundamental concepts of computing paradigms and their use in emerging vehicular environments, we believe SOTA IOV architectures must aim to encompass novel computing paradigms to build a more robust and efficient network. In the study [10], we can find a comprehensive analysis of such paradigms. According to the authors in [10], a paradigm is a specific way of understanding a particular thing. Specifically, these computing paradigms aim to solve data availability, storage, and analysis in the vehicular network context.

Some central ideas about such paradigms and their correlation are, for example, that Mobile Edge Computing (MEC) paradigm encompasses Vehicular Fog Computing (VFC) and Vehicular Edge Computing (VEC). In this perspective, both roadside units (RSUs), and vehicles, with their Onboard Units (OBUs), act as edge nodes, increasing data processing efficiency [10].

2.1.1 Vehicle-to-everything

The V2X mode is composed of the five communication modes presented in Figure 2, vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-sensor (V2S), vehicle-to-pedestrian (V2P), and vehicle-to-roadside unit (V2R), respectively. Thus, in V2X is where communications are attained both with infrastructure and with vehicles.

We will only consider V2V and V2I modes for this thesis's core because these are the ones used by VANETs. When V2X is integrated into vehicles, drivers can receive critical information about weather patterns, nearby accidents, road conditions, road works warning, emergency vehicle approaching, and other drivers' activities using the same or a nearby route [22].

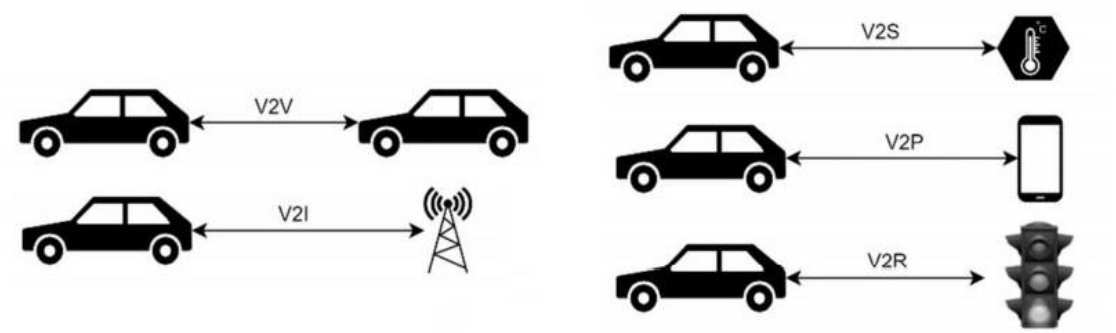

Figure 2: V2X communication modes. In order: vehicle-to-vehicle (V2V) , vehicle-to-infrastructure (V2I), vehicle-to-sensor (V2S), vehicle-to-pedestrian (V2P), vehicle-to-roadside unit (V2R). Image extracted from [16].

V2I communication is used between moving vehicles and static infrastructure, namely the RSU and Base Stations (BSs). The communication architecture is centralized, i.e., the RSU act as a central point for one or more vehicles. Moreover, a node may wish to disseminate information in several ways, primarily considering unicast and broadcast. For example, RSU can broadcast general data and unicast vehicle's requests. Typically, the RSU is attached to a power supply and a backbone network connection; therefore, it is unnecessary to consider these resources' consumption in its operation. Also, the RSU can be equipped with multiple directional antennas to overcome environmental challenges. In practice, the RSU's downlink channel could be made much more robust when compared to the uplink because it is expected to suffer much more congestion than the latter.

V2I communication is usually employed to deliver information from road operators or authorities to the vehicles. Roadwork warning is a typical example of a V2I service; vehicular access network transceivers are deployed into the roadwork area, informing vehicles about exceptional road maintenance operations. The RSU cannot provide continuous network connectivity for moving vehicles. Instead, RSU can merely act as a service hotspot, delivering a pre-configured high-bandwidth service data exchange between the vehicle and fixed network whenever there are in its vicinity area. One example of such a data dissemination network is introduced in [23].

The V2V communication approach is most suited for short-range vehicular communications. The general idea is that the moving vehicles themselves create a wireless communication network in an Ad-Hoc networking manner and on a highly opportunistic basis. The communication architecture is distributed, as individual vehicles communicate equally in an Ad-Hoc way. The data exchange between

passing vehicles is typically broadcast. Still, it can also be unicast, multicast (for example, a platoon of vehicles exchanging traffic information) and geocast (in the case of accident warnings, affecting a specific region). A pure V2V network does not need roadside infrastructure, making it fast and relatively reliable for sudden incidents requiring information disseminated on the road.

Therefore, it has been the primary candidate for real-time communications targeting vehicles' safety applications. V2V communications are very challenging. Vehicles' connectivity may not be possible in this mode when no vehicle is within the communication range. It also may be hard to establish since the vehicles are moving in different directions and at different speeds, due to which there might be abrupt network topology changes.

In the past, V2V communications mainly focused on particular communication cases instead of a general "all-purpose" network. The most typical use cases are broadcasting emergency messages or other critical data. These are relatively simple connections, not designed for massive data flows with different security and QoS levels.

Nowadays, vehicles rely on a panoply of different services with different requirements. Some services might require stringent real-time communications with negligible latency and an end-to-end between source and destination. Other services might only need to communicate asynchronously, with low time constraints. Asynchronous or time shifting communications always involve data storage between the source and destination and operate opportunistically. As an example, we have the Delay-tolerant networks (DTNs) that are asynchronous and opportunistic networks that enable communications in disruptive scenarios [24], focused on media applications such as GPS (Global Positioning System) map updates or any other media content services for automotive infotainment systems. According to Yang et al. [25] and We et al. [26], IoV applications can be divided into three categories: safety, efficiency, and infotainment. Figure 3 presents a conceivable taxonomy of IoV applications. Specifying a suitable taxonomy eases the process of selecting the most suited communication mode taking into account the service applicability inside the IoV. The three main pillars are:

1) **Infotainment applications**: offering quality and value-added services, attracting all sorts of stakeholders' investment by building and conducting businesses to meet users' general purpose and specific needs. Services are specified by considering applications' local or global connection requirements, distinguishing between cooperative local and global internet services.

2) **Transportation efficiency applications**: designed to improve vehicle and driver proficiency, for example, by providing solutions that reduce fuel consumption or travel time. Based on the application's core work environment and purposes, the category is further divided into cooperative driving, parking navigation, route navigation, and intersection control.

3) **Safety applications**: intended to provide services that ensure safe driving, typically achieved through message notifications that can alert the driver or even take immediate action employing car control techniques. These are the most researched applications and can be

divided into various systems, such as collision avoidance systems (CAS) and speed control systems (SCS).

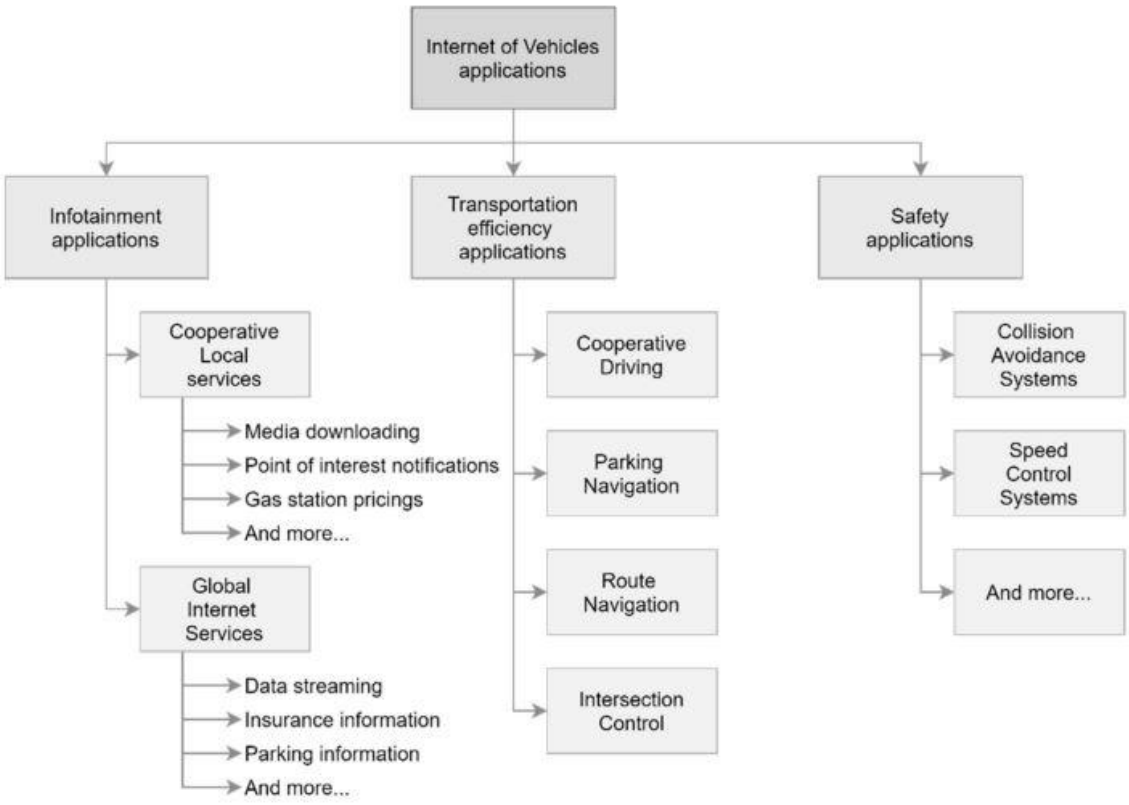

Figure 3: Taxonomy of applications for the IoV. Image extracted from [21].

Consequently, communication technologies must evolve to achieve solid IoV architecture, providing more bandwidth to support massive data flows with various data types and QoS. This wide variety of data constitutes the IoV Big-Data, which must be sourced, processed, and analysed. In addition, all communications must simultaneously meet stringent security requirements and enable efficient data dissemination schemes. Such schemes mainly depend on the available wireless technologies and the security methods applied to communications. Current wireless technologies are analysed in the following subsections, and an efficient, secure data dissemination scheme is proposed.

2.1.2 IoV Wireless Technologies

In an IoV, vehicles' connectivity is achieved through the OBU component equipped with necessary wireless technologies, such as DSRC, Wi-Fi-based, cellular 4G/LTE, 5G, etc. [27]. As stated above, and in addition to these technologies, RSUs are tailored with a backbone link that provides a fast connection to any required service, supporting V2X communication modes and enhancing their reachability.

DSRC is based on the IEEE 802.11p automobile-specific Wi-Fi standard, known as WAVE. In Europe, DSRC is named ITS-G5, G5 is derived from the frequency band (5.9GHz) upon which it was designed to operate. So far, DSRC has been considered the prominent communication technology for connected vehicles, but the development has stagnated. As such, the ever-evolving cellular technology

is becoming the leading technology, relying on 5G promises in delivering enhanced speed, stability, and security. We believe that both technologies should be deployed to achieve more reliable, efficient, and robust vehicular networks.

Cellular-V2X (C-V2X) is based on the LTE mobile communications standard and evolving to meet 5G standards, including benefits such as the millimeter-Wave (mmWave), which offers high throughput and bandwidth.

According to the 5G Automotive Association (5GAA), tests have shown that 5G networks can outperform 802.11p ones. Moreover, according to both International Telecommunications Union (ITU) and the Mobile and wireless communications Enablers for Twenty-twenty Information Society (METIS), 5G networks promise to be global once deployed [10]. Therefore, we can conclude that 5G and the following cellular technologies will dominate the market and eradicate WAVE that follows the 802.11p standards and other Wi-Fi-based technologies.

This work considers that vehicles, RSUs, and all IoV nodes are equipped with the best-suited communication technologies. Because communication technologies are constantly evolving, we are motivated to build efficient, secure data dissemination schemes in a modular approach, separating the communications flow from the technology that will enable it. In this way, we aim to guarantee full scheme functionality, regardless of the communication technology in use.

2.2 Secure Data Dissemination Schemes

This thesis is focused on studying, implementing, and proposing efficient, secure data dissemination schemes leveraged on a proxy re-encryption (PRE) system. Here we aim at answering the following questions: what is PRE? Why do we need it? And how are dissemination schemes built? We provide a theoretical explanation integrated with practical use-cases and implementations that emerged along with PRE history and communications security. We start by emphasizing data security importance and conclude with a comparison between presented PRE schemes, supported on tables showing essential requirements, advantages, and disadvantages. In this way, we will have solid foundations to build our PRE scheme crafted to the IoV.

2.2.1 Data Security

Data security is both the practice and the implemented technology devoted to protecting valuable and sensitive data, such as personal or financial information of the vehicles or any private information of any node inside the IoV network and integrated into the ITS. It must be addressed at all stages of the communications between the wide variety of nodes. As shown in Figure 1, the vehicles equipped with OBUs exchange data between themselves or RSUs using V2V and V2I modes, respectively. Moreover, the cyber domain to protect is also composed of cellular communication systems and other networks carrying susceptible data traffic and integrated into the IoV [28], such as those created by V2P or V2S communication modes.

The cybersecurity mission is to secure data within a cyber domain and protect digital assets against ever-growing cyber-attacks. Cybersecurity’s primary purpose is to ensure Confidentiality,

Integrity, and Availability (CIA) of data and services. The CIA triad is essential in cybersecurity as it provides vital security features, helps avoid compliance issues, ensures service continuity, and prevents reputational damage to the nodes [29].

Regarding *confidentiality*, cryptography is the best solution to deploy when having sensitive information in transit over the network because it provides proactive measures to prevent sensitive data from unauthorized disclosure while making it available only to the intended parties.

Through the process of encryption, cryptographic mechanisms ensure the confidentiality of sensitive data, converting plaintext of data into ciphertext, which is unreadable to any unintended entity. The idea of encryption/decryption algorithms is used to scram the plain text messages into ciphertext and vice-versa for decryption. A secret code or key supports this encoding/decoding. Without the secret key, the node cannot decrypt messages. Hence cryptographic mechanisms play a vital role in securing information. Cryptography is divided into several security mechanisms, including symmetric and asymmetric encryption.

In symmetric encryption, known as symmetric-key cryptography, a cryptographic scheme is considered in which the sender and receiver share a common key for message encryption/decryption. The key is distributed before transmission/communication between entities.

In asymmetric encryption, known as asymmetric-key cryptography or Public-Key cryptography, two distinct yet related keys are used, i.e., a key-pair: a public key (PK) is used for encryption and a secret key (SK) for decryption. From this point on, we will adopt the known term Public-Key Encryption (PKE), referring to asymmetric encryption. Intuitively, the PK is shared and known to everyone, while the SK must always be kept private to the respective node [30] [31].

In addition to complex cryptographic systems, solid passwords or passphrases and two-way authentication are other methods to ensure confidentiality and must always be considered for improved security and harden the CIA triad.

Integrity refers to preventing data from being tampered with, modified, or altered in an unauthorized manner. In IoV communications, transmitted data must be received intact and unaltered by any unauthorized party. Integrity is essential for data, whether in transit or stored. Various attacks compromise data integrity, such as a MitM or introducing malicious code in databases. The use of hashing algorithms is the most common method to check for data integrity. Other techniques include certificates and digital signatures.

In the IoV, exchanged messages are used by various applications, such as safety or infotainment. However, those messages could also be used maliciously (e.g., tracking vehicles). Therefore, vehicles use pseudonym identities (or certificates) provided by a Public Key Infrastructure (PKI). The PKI is a set of entities, roles, policies, hardware, software, and procedures to create, manage, distribute, use, store and revoke digital certificates and manage PKE [32]. Recently, Haidar et al. [33] have evaluated the performance of the PKI protocol regarding the reloading of certificates. Additionally, they have implemented a cooperative-ITS PKI architecture compliant with the European Telecommunications Standards Institute (ETSI).

Availability is also a security pillar, ensuring the constant availability of resources and services to only authorized parties and on time. All nodes must maintain reliable hardware and software in order to

provide consistent services. Other essential security controls for availability include data backup, patching, and redundant systems. Redundancy ensures fault tolerance by safeguarding that a secondary system will be available to continue delivering functions and services when a primary system fails to perform.

By researching social interactions between vehicles in the IoV, Magaia et al. [21] emphasize the security requirements described in Table 1, considering the application's vulnerabilities due to inherent communication issues or no control in data usage.

Table 1: Security requirements for the IoV.

Security Requirement	Description
Data authentication	Vehicle's identities should be verified when transferring data
Data integrity	Sent and received data should be verified to ensure correct data transferal
Data confidentiality	Data transmission between vehicles should be secret
Access control	Vehicles should be allowed to access services they are entitled to
Data non-repudiation	Vehicles cannot convincingly deny having generated signed data
Availability and Fault-Tolerance	Communication between vehicles should be ensured even under bad conditions in the event of an attack
Anti-jamming	Malicious vehicles cannot interrupt communications between other vehicles
Impersonation	A vehicle cannot impersonate another entity in the network
ID traceability	A vehicle's identity can be retrieved from sent messages
Vehicle privacy/anonymity	Sent messages can only be accessed by authorized vehicles and remote nodes. The vehicle's identity should be hidden

The CIA triad in cybersecurity, specifically for the IoV, is of utmost importance and must be continuously maintained. Security breaches and data thefts are increasing, damaging network nodes and conveying massive penalties due to the lack of compliance with data regulations, such as the General Data Protection Regulation (GDPR) [34].

2.2.2 Proxy Re-Encryption

A PRE system provides end-to-end communications security for messages traveling within the network. It comprises three entity types, the delegator, the proxy, and the delegatee. PRE is designed to efficiently convert a ciphertext originally intended for Alice (delegator) into the ciphertext of Bob (delegatee) via a semi-trusted proxy. Notice that once decrypted, both ciphertexts contain the same underlying plaintext. The prerequisite is that the proxy gains neither involved plaintext messages nor secret keys of Alice or Bob [35]. Featured with the specific translation property, PRE has attracted much attention from the industry and research community since its introduction in 1998 by Blaise, Bleumer, and Strauss [36]. In addition to secure data sharing in the cloud computation, PRE has also found many applications ranging from digital rights management (DRM) [37], VANETs [38], encrypted email forwarding [36], group key management [39] to distributed systems [40].

Furthermore, new definitions, security models, concrete constructions, and extensions of PRE have also been suggested so far. Different PRE schemes arose under different concepts to enhance security and efficiency while demonstrating features like transitivity and collusion-resistance to ensure minimal trust on the proxy and maximum key privacy [41]. The following paragraphs present the different concepts and characteristics that define the various PRE-based schemes. In addition, in Figure 4, we show some of the most general PRE-based ideas using: Public-Key Encryption, Identity-Based Encryption (IBE), Type-Based Encryption (TBE), Attribute-based Encryption (ABE), and Certificateless-based Encryption (CLE) [42], among others. Then, security models and properties will be introduced as required. Later, a conclusion is given by comparing and matching the PRE schemes *versus* the identified and most advantageous PRE characteristics to deploy in the IoV stringent environment.

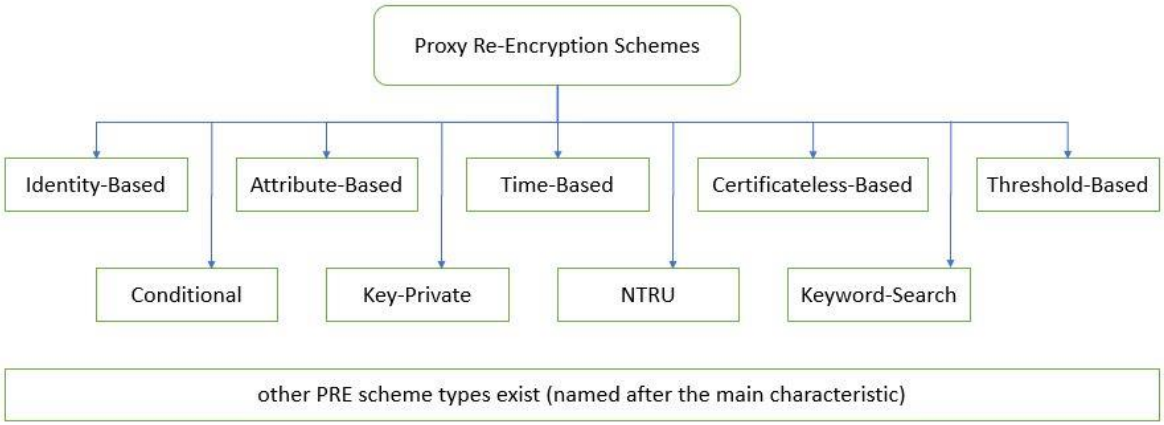

Figure 4: Classification of various PRE Schemes

2.2.3 1998-2004: Embryonic Stage

As stated before, PRE was introduced in 1998 by Blaze *et al.* [36]. The PRE scheme, known as atomic proxy encryption, was supported by the ElGamal encryption system and built to achieve bidirectional and multi-use capabilities. ElGamal is an asymmetric key encryption algorithm for PKE, based on the decisional Diffie-Hellman (DDH) key exchange method [43]. Nevertheless, the scheme could only achieve chosen-plaintext attack (CPA) security and was vulnerable to collusion attacks, such that proxy could expose the delegator's SK by colluding with the delegatee. A CPA is an attack model from cryptanalysis, which presumes that the attacker can obtain the ciphertexts for arbitrary plain texts. This attack aims to obtain information that reduces the security of the encryption scheme. For this reason, modern ciphers must provide semantic security, also known as ciphertext indistinguishability (IND).

Ciphertext IND is a property of many encryption schemes. If a cryptosystem possesses this property, an adversary (unauthorized party) will not distinguish pairs of ciphertexts based on their encrypted message. The property of indistinguishability under CPA is considered an essential requirement for any cryptosystem. For greater security, indistinguishability can be provided under chosen-ciphertext attack (CCA1) and adaptative chosen-ciphertext attack (CCA2), to be introduced later on in the subsequent PRE schemes [42].

Blaze *et al.* [36] left an interesting open problem constructing unidirectional and single-use PRE schemes. Consequently, in 1999, Jakobsson presented a unidirectional PRE scheme [44]. Specifically, this unidirectional PRE scheme requires the ciphertext re-encryption to be processed by multiple proxies, where each proxy owns a share of the re-encryption key (RK) secret. Therefore, if the number of honest proxies is larger than predefined, the SK of the delegator will not be disclosed even when the delegator is under collusion attacks.

In 2003, Ivan and Dodis [45] proposed a general construction of unidirectional single-use PRE. The system required splitting the delegator's SK into two parts and issuing them to the proxy and delegate separately. This approach suffers from numerous drawbacks that make it susceptible to attacks and unfeasible to deploy despite unidirectionality. In particular, the delegator's PK associated with the original ciphertext cannot be purely re-encrypted into the ciphertext intended for the delegate; this is, the delegatee cannot open the re-encrypted ciphertext with exclusively its own SK. Additional secrets from the delegator must be stored and managed by the delegatee to decrypt the re-encrypted ciphertext. Moreover, the proposed PRE scheme [45] is vulnerable to collusion attacks, specifically the ciphertext to be re-encrypted can be cracked in the event of a proxy and any delegate in the system conspiring together.

2.2.4 2005-2013: Definition Stage

In 2005, Ateniese *et al.* [40] described the desired characteristics and formalized the definition of the PRE scheme by considering that only a superficial and introductory notion of PRE was given by Blaze *et al.* [36]. In [40], novel unidirectional PRE schemes are presented, some of which rely on bilinear pairings for the first time. Additionally, the first time-based PRE schemes are presented, such that the delegator can periodically update delegation relationships without changing its PK, and the RK is only valid during a given amount of time. However, the re-encryption algorithm is single-use limited, where only the original ciphertext can be re-encrypted. Thus, constructing the multi-use unidirectional PRE scheme remains an open problem.

The IoV dramatically benefits from having single-use and multi-use ciphertext re-encryption capabilities. Different communication types may enjoy the extra security provided by single-use schemes or want the extra reachability supported by multi-use strategies. We can think of these capabilities as having what we call the *n*-multi-use capability, where *n* specifies the ciphertext re-encryption hop-limit. Also, we can acknowledge that the IoV benefits from having unidirectional communications schemes. Unidirectional schemes can achieve bidirectionality by stacking two distinct unidirectional channels with opposite communication directions, which provides more control and security than pure bidirectional schemes. Pure bidirectional schemes must share the same channel for both communication directions: ongoing and incoming ciphertexts, making them harder to control and more susceptible to interference and intrusion.

Security is made possible in cryptography through mathematical operations, such as addition, subtraction, OR, or XOR. Mathematics plays a vital role in developing security mechanisms supported by mathematical pairing/mapping functions and algorithms. We will observe various PRE schemes with

different pairing functions, such as bi-linear, multilinear, and even free pairing techniques. Cryptographic algorithms are intended around computational hardness, making such algorithms hard to break in practice by any unauthorized party. We are interested in obtaining the most efficient and secure schemes deployed in the dynamic IoV environment while requiring minimal computational processing to accomplish encryption, re-encryption, and decryption functions [46].

Before advancing into IBE-based PRE schemes, we provide a deeper analysis and a practical review of PKE-based PRE for the IoV. A PRE system supported on PKE comprehends a total system with the same premises and functionalities of the respective PRE and PKE systems.

The PKE cryptosystem offers security and requires two keys to work. Firstly, a PK must be made public and used to generate the ciphertext. Secondly, the SK is used to decrypt the ciphertext. The PK and the SK are not the same, yet they are related.

Compared to other encryption techniques, such as symmetric encryption that uses a unique key shared among all interested parties to encrypt/decrypt ciphertexts, a significant advantage of asymmetric PKE is that it does not force users to share their private SK. Therefore, it eases the cumbersome key management process [42]. Still, both techniques require key distribution to the sender and the receiver, enabling secure communications. Furthermore, PKE supports digital signing, which authenticates the recipient's identity and makes sure that the message was not tampered with in transit. The cons of asymmetric encryption are that it is time-intensive and requires considerably more computing effort, as we will later emphasize. Thus, possible architectures for upgraded efficiency while maintaining high-security standards contemplate hybrid cryptosystems. These systems combine symmetric and asymmetric algorithms. Hence, hybrid cryptosystems fulfil both symmetric and asymmetric cryptosystems' utility because they are more secure than symmetric cryptosystems and faster than asymmetric cryptosystems. The ISO/LEC standardization committee [47] suggests that a hybrid cryptosystem can be defined as the branch of asymmetric cryptography that uses convenient symmetric techniques to remove some of the problems inherent in the normal asymmetric cryptosystem.

Regardless of the implemented cryptographic approach over the IoV, it is considered that every node must operate cooperatively and have enough computational resources to perform the whole necessary cryptographic functions.

Nodes can be vehicles, RSU infrastructures, or any other entity that joins the IoV. For example, let us consider that Alice's vehicle, referred to as Alice, wants to forward encrypted data to Bob's vehicle, referred to as Bob, which is out of range. Such communication can be attained through one or multiple intermediary nodes, the proxy nodes. In this case, IoV leverages PRE by allowing a proxy to re-encrypt data, that is, to transform a ciphertext computed under Alice's PK, termed c_A, into another that only Bob's SK can open, termed c_B. Through re-encryption, a PRE scheme removes Alice's cumbersome task of having to decrypt data only to encrypt it again for Bob; instead, it immediately forwards c_A to the proxy node, along with an RK, to be defined in the following paragraph. The proxy's responsibility is only to re-encrypt the ciphertext with the respective RK, generating c_B, and to forward it to Bob. We can notice that PRE schemes offer end-to-end security in communications through always-on data cryptography; it also increases efficiency by reducing data encryption/decryption phases and gives the possibility for any proxy node to participate in the communications.

For clarity purposes, in Figure 5, we illustrate this example to define a unidirectional single-use PRE scheme supported by PKE methods. Firstly, Alice generates a PK-SK key pair (pk_A, sk_A) under a standard key generation algorithm for the underlying cryptosystem, crafted with all required crypto algorithms. With this key pair, Alice can decrypt the ciphertext meant to it, c_A, with its own SK (sk_A). To generate c_A, Alice must share its own PK (pk_A) so that another node can use it to encrypt data specifically for Alice. Encryption is done by applying a standard encryption algorithm requiring only two inputs: the PK of the recipient and the message, to be encrypted (m). As a delegator, Alice must decide who will be the delegate's final recipient of the encrypted message. The decision implies that Alice already knows the delegatee's PK. In our example, Bob is the delegatee with a known PK, pk_B. Subsequently, Alice creates an RK for a proxy, $rk_{A\to B}$, using a standard re-encryption key generation algorithm, which requires three inputs: pk_A, sk_A and pk_B, respectively. Afterwards, Alice communicates with a proxy vehicle in a V2V mode, transmitting both the encrypted message, that is, the ciphertext c_A, and the corresponding RK, $rk_{A\to B}$. The proxy vehicle applies a standard re-encryption algorithm over the received inputs $rk_{A\to B}$ and cyphertext c_A, and outputs ciphertext c_B. Notice that at this stage, the proxy vehicle does not know how to decrypt the message m contained in both c_A and c_B. Ciphertext c_B is the re-encrypted message destined to Bob. As a delegatee, Bob possesses the only key able to decrypt c_B, its own SK, sk_B [40].

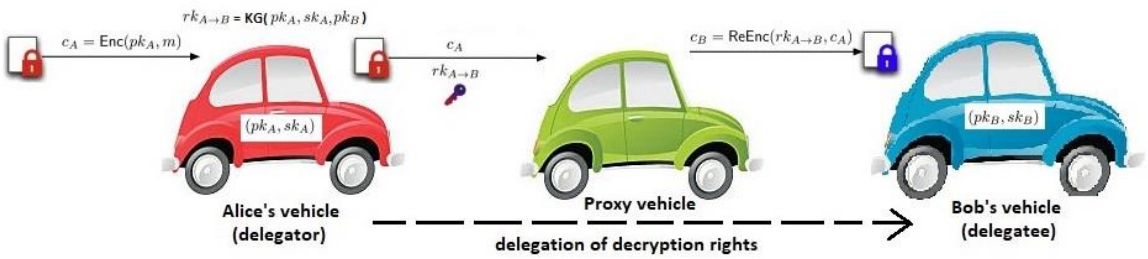

Figure 5: Proxy Re-encryption workflow

A well-designed and realized PRE system can accommodate efficient data dissemination schemes for the IoV, maintaining communications end-to-end security. In addition to data confidentiality and dissemination efficiency, availability is improved by enabling all nodes to share any class of responsibilities, more or less sensitive. Availability is crucial to building a robust fault-tolerant system and empowers nodes to trust each other since the intended destination node is the only one who can decrypt the ciphertext. Therefore, multiple redundant data traffic routes or links can be considered for simultaneous use or to serve as backup routes.

However, the re-encryption algorithm in [40] is limited to a single-use/single-hop mode. In this mode, only the original ciphertext can be re-encrypted. Thus, constructing a multi-hop unidirectional PRE scheme is still a challenge to address. For the IoV, a multi-hop mode is essential to support secure, efficient data dissemination management with QoS requirements.

In 2007, Canetti and Hohenberger [48] pushed the bar even higher by developing the first CCA2-secure PRE scheme, featuring bidirectional and multi-use properties [49]. Building a CCA2 security model implies that it is also CCA1 and CPA secure. The security model definitions can be found in [48]. Green and Ateniese introduced a PRE scheme in [50], supported on an ID-based PK cryptosystem

environment, known as IBE-PRE. Additionally, in [50], the IBE-PRE is proposed and inspired by Boneh-Franklin's ID-based encryption scheme [51], where ciphertexts under the delegator's identity can be re-encrypted under the delegatee's identity.

IBE is a PKE scheme in which the PK can be an arbitrary string. Thus, PKE is extended by IBE schemes, which have four algorithms: **(1) setup**: generates global system parameters and a master-key (MK), **(2) extract**: uses the MK to generate the SK corresponding to an arbitrary identity string, the PK-ID, **(3) encrypt**: encrypts messages using the PK-ID, and **(4) decrypt**: decrypts messages using the corresponding SK.

Firstly introduced by Shamir [52] in 1984 and after realized by Boneh-Franklin in 2001 [51], IBE has proven helpful in solving several key distribution issues and PKI deployment. IBE has permitted the development of a variety of novel cryptographic protocols, such as secret handshakes [53], Public-key Searchable Encryption (PSE) [54] [55], CCA2-secure PKE [56], and digital signatures [57].

IBE arose from Shamir's drive to simplify certificate management in email systems. For example, when Alice sends an encrypted email to Bob at bob@company.com, she encrypts her message using the PK string "bob@company.com," removing the need for Alice to obtain Bob's PK certificate by contacting the certificate authority (CA). When Bob receives the encrypted email, he contacts a third party, the Secret Key Generator (SKG). Bob authenticates himself to the SKG in the same way he would authenticate himself to the CA and obtains his SK from the SKG. Bob can then decrypt the message and read his email. We can notice that, unlike the existing secure email infrastructure, Alice can send encrypted emails to Bob even if Bob has not yet set up his PK certificate. Moreover, because the SKG knows Bob's SK, Boneh and Franklin [51] discuss key revocation and several new applications for IBE schemes.

Definitely, in [50], the IBE-PRE scheme allows a proxy to translate a ciphertext under Alice's identity into one computed under Bob's identity. The proxy uses RKs to perform the translation without learning the plaintext. Moreover, no information on the SKs of Alice and Bob can be deduced from the RKs. According to M. Green and G. Ateniese, in [50], the scheme constructions are compatible with existing Boneh-Franklin IBE deployments and can be implemented using existing secrets and parameters.

In 2008, Deng et al. [58] projected a pairing-free bidirectional and single-use PRE scheme to avoid expensive bilinear pairing operations. Independently, Libert and Vergnaud [59] presented a unidirectional PRE scheme with replayable chosen-ciphertext attack (RCCA) security in a non-adaptative corruption (NAC) model. Both of these constructions rely on bilinear pairings. It is vital to notice that despite the recent advances in implementation techniques, the pairing computation is still considered a costly operation. Hence, it would be desirable for cryptosystems to be shaped without relying on pairings, particularly in computational resource-limited scenarios.

In 2009, Shao et al. [60] intended to solve one of the open problems left in [48] by building the first unidirectional PRE scheme while achieving CCA security and collusion-resistance. Moreover, the expensive bilinear pairing operation was removed. Liang et al. [61] presented the concept of attribute-based PRE, ABE-PRE, and gave concrete construction for it, providing fine-grained access control. In

ABE-PRE, the delegator associated with some specified attributes could freely allow a proxy to re-encrypt a ciphertext concerning a particular access policy to another encryption under a different access policy. The notion of conditional PRE (C-PRE) was first introduced by Weng et al. [62] [63], where the delegator owns a fine-grained control over the delegation. The ciphertext can only be re-encrypted by the proxy and then decrypted by the delegate if this ciphertext satisfies a specific condition designed by the delegator. In the meantime, Ateniese et al. [64] presented the first key-private PRE (KP-PRE), which is CPA-secure, featured with key-private (or anonymous) re-encryption keys as an additional applicable property of PRE schemes. In [64], Ateniese et al. define a secure and key-private PRE scheme; it is also shown that the key-private property is not captured or achieved in former definitions and schemes, respectively.

In 2010, Weng et al. [65] proposed a new unidirectional PRE scheme and proved its chosen-ciphertext security in the adaptative corruption model without random oracles (ACMnRO). The scheme was compared with the best-known unidirectional PRE schemes proposed by Libert and Vergnaud in [59] and proved to be more efficient and secure by achieving CCA-security instead of the weaker RCCA-security. Sherman Chow et al. [66] showed that security proof in [58] is vulnerable to CCA by presenting a concrete attack. Subsequently, an improved pairing-free unidirectional PRE scheme has been projected to achieve high efficiency and CCA security. Chow et al. claim to gain increased efficiency and CCA-security using the token-controlled encryption (TCE) technique under the computational DDH assumption. Matsuda et al. [67] came up with a bidirectional CCA-secure PRE scheme without bilinear maps in the standard model; they have also studied and defined new variants of lossy trapdoor functions (LTDFs) based on the DDH assumption. Sur et al. [68] arose with the concept of CLE applied to PRE schemes. Based on the idea of certificateless-based public-key encryption, such as in [69], they present a unidirectional concrete scheme based on bilinear pairings, which enjoys the advantages of certificateless public-key cryptography while providing the functionalities of PRE.

Furthermore, in [68], the presented scheme is CCA-secure in the random oracle model and compatible with other CLE-based deployments. In [70], S. Lee et al. proposed a digital content sharing model for DRM, based on Ateniese et al.'s PRE scheme [40]. Finally, inspired by PKE with keyword search, Shao et al. [71] introduced a new cryptographic primitive, called PRE with keyword search (PRES), being the combination of PRE and PKE with keyword search (PEKS).

PRES is motivated by the following scenario in email systems: Charlie sends an encrypted email containing keywords like "urgent" to Alice under Alice's PK. Then, Alice delegates her decryption rights to Bob *via* her mail server. The desired situations are: (1) Bob can decrypt mails delegated from Alice by using only his PK, (2) Bob's mail gateway, with a trapdoor for Bob, can test whether the email delegated from Alice contains some keywords, such as "urgent," (3) Alice and Bob do not wish to give the mail server or mail gateway the access to the content of emails. PRES can be applied to the IoV environment, enhancing efficiency and security in encrypted data flows. Giving the delegatee the ability to search particular keywords in the ciphertext makes it possible to take preventive and fast decisions on what to do if the keywords are found or not.

In 2011, to achieve CCA security in the standard model without sacrificing collusion-resistance, Libert and Vergnaud [72] presented the first CCA-secure and collusion resistant unidirectional PRE scheme. Shao et al. [73] introduced the first CCA-secure unidirectional and multi-use PRE scheme, answering an open problem proposed in [48]. Chen et al. [39] came up with a solution to manage group keys securely, building the first PRE scheme based on Rivest-Shamir-Adleman (RSA) PK cryptosystem, with unidirectionality and multi-hop properties.

In 2012, Hanaoka et al. [74] announced the first generic construction of a CCA-secure unidirectional PRE scheme. In particular, the scheme in [74] employs complete CCA security (i.e., not relaxed CCA security such as RCCA security) proven against powerful adversaries, given a more favourable attack environment than in all previous works; also, random oracles are not required. They have established a novel methodology for designing PRE on a specific class of Threshold-based Encryption (TBE). Nevertheless, the PRE scheme cannot be regarded as the pure PRE scheme since the delegate needs a share of the delegator's secret to perform the decryption over the re-encrypted ciphertext. Shao et al. [75] solved an open problem left by Ateniese et al. [64], introducing a PRE scheme that achieves CCA security and key privacy in the standard model, supported by 5-Extended Decision Bilinear Diffie-Hellman assumption and DDH assumption.

In 2013, Isshiki et al. [76] highlighted that Hanaoka et al.'s proposal [74] does not fully capture the CCA security notion, and so, [76] presents a full CCA security definition that is extended from [74]. Additionally, Isshiki et al. also propose the first PRE scheme with CCA security in the standard model (i.e., without the random oracle idealization). They claim that their PRE scheme is efficient and relies on mild complexity assumptions in bilinear groups. Aiming to achieve more fine-grained control over the delegation, Fang et al. [77] introduced a new cryptographic primitive called fuzzy conditional PRE (FC-PRE), where the conditions in C-PRE are characterized as a set of descriptive keywords.

2.2.5 2014-2019: Innovation Stage

In 2014, Zhang et al. [78] exposed a vulnerability in Libert and Vergnaud's scheme [72], proven secure in the knowledge of secret key model but not secure in the chosen key model. Hence, Zhang et al. [78] presented an efficient CCA-secure PRE scheme in the more robust chosen key model. To protect against quantum attacks, Kirshanova [79] proposed a new unidirectional PRE scheme based on the hardness of the Learning With Error (LWE) problem. Kirshanova's scheme achieves collusion resilience and non-interactivity, i.e., does not require any trusted authority for the re-encryption key generation. Furthermore, the security proof of her scheme, provably CCA1-secure, is given in the selective model under the LWE assumption.

It is vital to acknowledge that a quantum-safe PRE scheme is supported on quantum-safe cryptography, referring to the efforts to identify algorithms resistant to attacks by classical and quantum computers. In this way, we can keep information assets secure even after a large-scale quantum computer has been built. Although still theoretical, quantum attacks should be considered when planning for the next-generation IoV.

Still, in 2014, Guo et al. [80] analyzed that almost all the PRE schemes failed to capture the forgeable re-encryption attack in the sense that RKs may be forged through the collusion attack launched by the proxy and a delegatee. Consequently, they first defined the concept of unforgeability of RKs to capture the above attack. Liu et al. [81] came up with the idea to incorporate the concept of time into the combination of ABE and PRE, proposing a time-based PRE (TB-PRE) scheme to allow a user's access right to expire automatically after a predetermined period.

In 2015, Tang et al. [82] designed and presented a unidirectional and multi-hop PRE scheme using multilinear maps under solid assumptions in the setting of multilinear groups [83]. It answers an interesting open problem left by Canetti and Hohenberger [48] about creating a PRE scheme with simultaneous unidirectional and multi-hop features. Such features are very desired in the IoV environment. For example, it enables efficient email forwarding by allowing a delegatee to forward its email received by the delegator to another delegatee. Nunez et al. [84] examined the existing security models for PRE schemes. They proposed a new nomenclature for these models regarding the accessibility of the decryption and re-encryption oracles considering the security game. Also, because most previous PRE schemes are constructed under the number-theoretic assumption, Nunez et al. [2] presented new bidirectional and multi-use PRE schemes based on NTRU, the well-known lattice-based asymmetric cryptosystem. NTRU stands for Nth Degree Truncated Polynomial Ring Units and corresponds to the first public-key cryptosystem not based on factorization or discrete logarithmic problems. NTRU is a lattice-based alternative to the RSA or the Elliptic-curve cryptography (ECC) and is based on the shortest vector problem in a lattice.

The advantages of NTRU are demonstrated in [2]: providing more efficient encryption and decryption, much faster key generation allowing the use of disposable keys (because keys are computationally easy to create), and requiring low memory use compared to other existing symmetric and asymmetric cryptosystems. These advantages are of utmost importance and suit well the IoV requirements. Moreover, lattice-based cryptography is not only desired for its efficiency but also because of its role in post-quantum cryptography.

By considering the asymmetric capacity among mobile devices and the server, Deng et al. [85] presented asymmetric cross-cryptosystem re-encryption by allowing an authorized proxy to convert a complicated ID-based broadcast encryption ciphertext deployed in the server into a simple ID-based encryption ciphertext affordable to mobile devices. In their work, Deng et al. devoted to building a hybrid ID-based re-encryption mechanism that can bridge ID-based broadcast encryption (IBBE) system and an IBE system. The IBBE allows multiple authorized visitors to access the same outsourced data, and the re-encryption achieves the access rights delegation from an IBBE user to an IBE user. The IoV can be significantly enhanced by cross-cryptosystem re-encryption concerning communications efficiency and flexibility, which is supported on different cryptosystems integration.

In 2016, Deng et al. [86] extended the previous work in [85] by proposing a new PRE pattern, referred to as an identity-based proxy re-encryption version 2 (IBPRE2). With IBPRE2, it is possible to take advantage of IBBE to securely share data with a set of recipients and then incorporate an additional one into the authorized group through a re-encryption mechanism without decrypting the IBBE ciphertext

nor leaking any sensitive information. In [86], the authors formalize the security requirements for IBPRE2 and propose a provably CCA-secure scheme. Lu and Li [87] have proposed a certificate-based PRE scheme without bilinear pairings, reducing the computation cost. In addition, the proposed scheme in [87] is proven secure under the computational Diffie-Hellman assumption. By incorporating C-PRE, IBE-PRE, and broadcast-PRE, Xu et al. [88] proposed a versatile primitive called Conditional Identity-based Broadcast (CIB) PRE and formalized its semantic security. In [88], the CIB-PRE scheme is shown to be efficient and secure. Additionally, it is important to notice that the initial ciphertext, the re-encrypted ciphertext, and the RK have all constant sizes, and the parameters to generate the RK are independent of the original receivers of any initial ciphertext.

The CIB-PRE allows a sender to encrypt a message to multiple receivers by specifying these receivers' identities. The sender can delegate an RK to a proxy to convert the initial ciphertext into a new one to a new set of intended receivers. Moreover, the RK can be associated with a condition such that only the matching ciphertexts can be re-encrypted, which allows the original sender to enforce access control over its remote ciphertexts in a fine-grained manner.

In 2017, Nunez et al. [89] analyze the secure access delegation problem. They review the main PRE schemes so far and provide a detailed analysis of their characteristics; they also study, i.e., theoretically and empirically, the efficiency of particular schemes based on their implementation. Jiang and Guo [86] proposed an encrypted data sharing scheme for secure cloud storage based on conditional PRE broadcast technology. Their scheme achieves broadcast data sharing by taking advantage of broadcast encryption and achieves dynamic sharing that enables adding a user to and removing a user from sharing groups dynamically without changing the encryption PKs. In [90], the correctness and security are proved, the performance is analyzed, and the experimental results are shown to verify the proposed scheme's feasibility and efficiency.

In 2018, Yasumura et al. [91] proposed an ABE-PRE for revocation in cloud storage. They showed that the proposed method reduces the communication cost by one-quarter compared to existing ones. Current revocation methods of ABE such as [92], [93], and [94], are offered based on using ABE to encrypt the data entirely.

However, actual implementations are designed in a hybrid fashion, i.e., cryptosystems that take advantage of ABE and symmetric encryption, such as the Advanced Encryption Standard (AES), are used for upgraded performance. Data is encrypted with AES in such hybrid encryption, and the AES key is encrypted with ABE. Consequently, because existing revocation methods affect only ABE ciphertext, this fact introduces a problem in which users can keep the AES key before the revocation and decrypt data even after the users are revoked. Therefore, in [91], it is highlighted that although existing revocation methods can be applied to revoke users from ABE, re-encrypting data with a new AES key is necessary so that the old AES key can no longer be used.

Also, in 2018, and considering the increasing interest in fog computing techniques, Wang [95] proposed a leakage resilient ID-based PRE scheme implemented over two platforms. The obtained results showed that the scheme was feasible in practice. Independently, online social networks (OSNs) were becoming popular around the world due to their openness. However, OSNs faced several issues

such as fine-grained access control, efficiency, and usability. For that reason, Huang et al. [96] proposed PRECISE, a secure and efficient identity-based private data sharing scheme in OSNs with Big-Data, in which the data owner could conveniently and securely broadcast private data to a group of users at one time. Huang et al. [96] adopted attribute-based C-PRE to guarantee that only the data disseminators whose attributes satisfy access policy can disseminate the data to their own social space. In detail, the RK is associated with a set of attributes such that only the matched ciphertexts can be re-encrypted, allowing the data owner to enforce access control over the disseminated ciphertexts. Nahri et al. [97] presented necessary premises for deploying the IoV while integrating Big-Data analytics of road network traffic measurements.

In 2019, to achieve enhanced PRE efficiency and security Reddy et al. [98] proposed an algorithm named Modified Lattice-Based Cryptography (MLBC). MLBC is based on PKE, and it provides more efficiency than other existing cryptography techniques. Simulations evaluated the MLBC based on several parameters: encryption time, proxy key generation time, re-encryption time, and total computation time. Concerning the performance analysis of the symmetric PRE scheme, and because most PRE schemes only use asymmetric key cryptography, Meiliasari et al. [99] devoted themselves to measuring the performance of the re-encryption function, using the AES Cipher-Block Chaining (CBC) re-encryption time as a baseline. Wu et al. [100] built a multi-hop and unidirectional identity-based PRE from lattices. They have split the functions of decryption and re-encryption separately and designed the double keys mechanism, where two keys are generated for each user, one key is used to decrypt the ciphertext by Pre-Image Sampling Technique (PIST), and the other is used to create the RK by Bonsai Trees technique. In [100], the generation of the RK is non-interactive, collusion resistant, and secure in the model of selective-identity-IND-CPA over the decisional LWE assumption. Compared with some previous identity-based PRE schemes from bilinear pairings, the format of transformed ciphertext in [100] remains unchanged; furthermore, it can also resist quantum analysis. Manzoor et al. [101] believe that the centralized architecture model in IoT systems will struggle to scale up to meet the demands of future IoT systems. Thus, they have projected a blockchain-based PRE scheme for IoT data sharing to tackle scalability and trust issues and automatize payments, boosting availability and other QoS requirements.

Consequently, compared to the business scenario where data is stored in a cloud-based infrastructure, there is no need for manual verification of the payments and the predefined requirements. Also, disputes on these aspects are entirely avoided. Lin et al. [102] were motivated to overcome most metro city issues about finding parking spaces, so they have proposed a real-time parking service with PRE in vehicular cloud computing. The scheme provides real-time parking lot information for drivers and integrates PRE to achieve many security requirements such as mutual authentication, user anonymity, data confidentiality, traceability, and revocability.

2.2.6 2020-Present: Quantum-Safe Stage

In 2020, Maiti and Misra [103] proposed a Privacy-Preserving Identity-Based Broadcast PRE (P2B) scheme. P2B uses the Lagrange interpolation polynomial theorem to provide privacy to identities

of the receiver group of broadcasted re-encrypted ciphertext. P2B is proven to be CPA-secure using the random oracle model and successfully hiding the receivers' identities. Moreover, P2B was implemented and compared with other existing systems, reducing the decryption time by 68% than the recent existing Broadcast-PRE schemes and 98% than the existing privacy-preserving schemes. Regarding data dissemination efficiency and security, Sharma et al. [104] studied schemes that attempt a combined solution integrating PRES schemes searching on encrypted data supported by C-PRE schemes. They have identified authentication hazards on recent techniques and proposed extensions to improve security along with data owner authentication.

In 2021, Dutta et al. [105] studied IBE-PRE schemes, acknowledging their potential but noticing that so far, there was no collusion-resistant unidirectional IBE-PRE scheme secure in the standard model, which could withstand quantum attacks. For that reason, they have presented the first concrete constructions of collusion-resistant unidirectional IBE-PRE from lattices for both selective and adaptive identity, which are secure in the standard model based on the hardness of the LWE problem. However, all the proposed constructions are limited to single-hop capabilities. Luo et al. [106] proposed a new key-policy attribute-based PRE (KP-ABPRE) scheme from standard lattices. Unlike the previous KP-ABPRE schemes based on classical-theoretic assumptions, the authors present the first KP-ABPRE unidirectional scheme based on the LWE problem, which is widely believed to be quantum-resistant. Furthermore, their KP-ABPRE design achieves multi-hop capabilities with CPA security in the selective security model based on the LWE assumption.

2.2.7 PRE Scheme Comparison

Table 2 compares the most exciting and referenced PRE schemes to summarize this section. The comparison is supported by the most crucial PRE properties described as follow:

1) Directionality (Dir): The PRE scheme's ability to allow the proxy to translate the delegator's ciphertext into the delegatee's ciphertext. It can be unidirectional, bidirectional, or broadcast.
2) Hop: In a multi-hop PRE scheme, ciphertexts generated by either the Encrypt or the ReEncrypt algorithms can be taken as input to ReEncrypt to be re-encrypted. In contrast, only the original ciphertexts generated by Encrypt can be re-encrypted with the ReEncrypt algorithm in a single-hop PRE scheme.
3) Security Model: In typical PKE environments, security definitions are generally developed from the hostile relationship between a security goal and an adversary with a specific attack. With the indistinguishability goal and various attacks, e.g., CPA and CCA, different models are used such as the Standard Model (SM), the Random Oracles (RO), or the adaptative corruption model (ACM).
4) Mapping/ Assumption: The type of cryptography key mapping and pairings with the corresponding base assumptions to prove the harness of the security model.
5) Collusion-Resistant (CR): In a collusion-resistant PRE scheme, even the proxy colluding with the delegatee cannot recover the delegator's SK. Otherwise, the SK of the user will be

disclosed in the case this user delegates its decryption rights to the malicious proxy and participants.

6) Non-Transitive (NT): The PRE scheme is called non-transitive if the decryption rights cannot be redelegated by the proxy alone. Thus, it is infeasible for the proxy to calculate $rk_{i\to k}$ from $rk_{i\to j}$ and $rk_{j\to k}$.
7) Non-interactive (NI): If the SK sk_j of the delegatee (user j) is not required in the RK generation algorithm ReKey, then the underlying PRE scheme is regarded to be NI. Thus, an RK $rk_{i\to j}$ can be generated with the delegator's secret/public key-pair (sk_i, pk_i) and the delegatee's PK pk_j. I.e., the SK sk_j is not required as the input of the algorithm ReKey.
8) PRE-type-based (PTB): Various PRE types exist to achieve different goals under different environments and foundations. Generally, these schemes rely on PKE techniques.

Table 2: Property comparison of PRE schemes.

Scheme	**Properties**							
	Dir	**Hop**	**Security Model**	**Mapping/ Assumption**	**CR**	**NT**	**NI**	**PTB**
Blaze *et al.* [36]	bi	multi	CPA (SM)	Group (DDH)	No	No	No	Atomic
Ateniese et al. [40]	Uni	Single	CPA (SM)	Pairing (eDBDH)	Yes	yes	yes	Temporary
Canetti et al. [48]	bi	multi	CCA2 (SM)	Pairing (DBDH)	No	No	No	IBE
Green et al. [50]	Uni	Multi	CPA (RO)	Pairing (DBDH)	No	Yes	Yes	IBE
Deng et al. [58]	Bi	Single	CCA (RO)	Pairing-free (mCDH)	No	No	No	-
Libert et al. [59]	Uni	Single	RCCA (SM-NAC)	Pairing (3-wDBDH)	Yes	No	Yes	Temporary
Shao et al. [60]	Uni	Single	CCA (RO)	Pairing-free (DDH)	Yes	Yes	Yes	-
Liang et al. [61]	Uni	single	CCA (RO)	Pairing (qDBDH)	Yes	-	No	ABE
Ateniese et al. [64]	Uni	single	CPA (SM)	Pairing (eDBDH)	No	-	-	Key-Private
Weng et al. [65]	Uni	Single	CCA (ACMnRO)	Pairing (DBDH)	Yes	-	Yes	-
Matsuda et al. [67]	Bi	Multi	CCA (SM)	Pairing-free (DDH)		No		PKE-rLTDF
Sur et al. [68]	Uni		CCA (RO)	Pairings (pBDHI)	Yes	-	Yes	CLE
Shao et al. [71]	Bi	Multi	CCA (RO)	Pairings (DBDH)	Yes	Yes	Yes	PRES
Libert et al. [72]	Uni		CCA (SM)	Pairings (3-QDBDH)	Yes	Yes	Yes	-
Shao et al. [75]	Uni	single	CCA (SM)	Pairings (5-eDBDH)	Yes	yes	Yes	Key-Private
Isshiki et al. [76]	Uni	Single	CCA (SM)	Pairings (AmDBDH)	Yes	Yes	Yes	-
Fang et al. [77]		Single	CCA (RO)	Pairings (DBDH)	Yes	-	Yes	FC
Kirshanova [79]	Uni	Single	CCA (SelM)	Lattices (LWE)	Yes	Yes	Yes	-
Nunez et al. [2]	Bi	Multi	CPA (Ring-LWE)	Lattices (LWE)	Yes	Yes	Yes	NTRU PKE
Xu et al. [88]	Br	Single	sID-CPA (RO)	Pairings (DBDH)	Yes	Yes	Yes	CIB
Wu et al. [100]	Uni	Multi	sID-CPA (SM)	Lattices (LWE)	Yes	Yes	Yes	IBE

Maiti et al. [103]	Br	Single	sID-CPA (RO)	Bilinear (Lagr. Interp)	Yes	Yes	Yes	P2B
Dutta et al. [105]	Uni	Single	sID-aID-CPA (SM)	Lattices (LWE)	Yes	Yes	Yes	IBE
Luo et al. [106]	Uni	Multi	CPA (SelM)	Lattices (LWE)	Yes	Yes	Yes	KP-ABE

This chronological research on PRE evolution can be divided into four phases:

1) 1998-2004: PRE notion received minimal attention and minimal work effort throughout its embryonic stage. The PRE concept was introduced [36] but lacked solid guidelines and robust security.

2) 2005-2013: PRE desired characteristics are expressed and formalized [40], enabling practical PRE schemes deployment with intrinsic efficiency and security requirements. Subsequently, PRE received tremendous attention from researchers worldwide. Various PRE schemes arose to tackle different problems regarding data sharing architectures. Table 2 reveals the hasty PRE advances towards improved efficiency and security. Efficiency is achieved by specifying and introducing new PRE primitives crafted to accomplish the type of task. However, during this phase, most PRE schemes rely on security models that are constructed under the DDH assumption with bilinear pairings or by some similar modified premise. Although proven solid and attaining high-security standards, pairing-based cryptography requires heavy computational processing and is vulnerable to quantum attacks. Moreover, most PRE applications are fashioned to support relatively static data sharing topologies, such as email servers or media content databases.

3) 2014-2019: Innovative PRE implementations ascended, integrating and mixing several existing PRE schemes in an effort to respond to all demanding environments imposed by the inherent technological evolution accompanied by the upsurge of mobile user data consumption and transmission. In this phase, researchers started focusing on PRE schemes designed to enable wireless communications applied to mobile and dynamic network topologies, such as the VFC or the broadcast ID-based PRE techniques for IoV applications. Furthermore, new cryptographic structures were investigated, shifting the typical DDH-based number-theoretical assumptions hardness to the lattice-based cryptography, conjecturing harder problems on point lattices such as the LWE problems and their efficient ring-based variants to build the foundation for secure cryptographic systems. Indeed lattice-based cryptography includes apparent resistance to quantum attacks (in contrast with most number-theoretic cryptography), high asymptotic efficiency and parallelism, security under worst-case intractability assumptions, and solutions to long-standing open problems in cryptography [107].

4) 2020-Now: By observing Table 2, one can quickly notice that the research trend focuses on PRE schemes featured with quantum security supported by LWE assumptions. The current phase demands stringent security specifications while offering efficient and fast computational processing PRE schemes. In a nutshell, considering mobile nodes with low

computational power and yet with high-security standards and massive data flows, the next-generation PRE schemes must be more efficient, secure, and included with desired properties to meet the necessities of highly dynamic topologies.

2.3 Software Simulation Tools for the IoV

IoV is a complex vehicular network that englobes numerous nodes that aim to wirelessly communicate in various manners (i.e., unicast, multicast, broadcast).

Over the last decade, a new paradigm in end-to-end communications has emerged, mostly in academia and industrial research laboratories, based on the notion that the next-hop between sender and receiver is not known in advance. These networks, typically called DTNs and Opportunistic Networks (OppNets), are characterized as opportunistic because, like nodes in mobile *Ad-Hoc* networking infrastructure, the forwarding nodes are mobile and dynamic, i.e., they come and go in unpredictable ways [108]. More precisely, OppNets are a mechanism of mobile *Ad-Hoc* networks that exploit users' mobility to provide data sharing. Individual nodes store, carry and forward messages using direct communication between devices; no fixed communications infrastructure is required.

Due to its highly dynamic topology, leveraging the IoV architecture could include delay-tolerant capabilities taking advantage of all opportunities to exchange messages.

Many researchers lack the resources to implement a “physical” IoV for experimentation; therefore, simulation tools play an essential role in measuring the performance of the proposed research. Simulating the IoV with software tools can be more or less complex, depending on the desired outcomes and the used simulator. The selected simulator should be capable of implementing the different features of the IoV and supporting performance metrics that will allow us to evaluate the network performance. Hence, choosing “the best” simulator for this work is a decision that needs to be made with much thought. To decide which simulation tool to use, we must consider several aspects such as the environment where simulations are performed, its capabilities, its effectiveness in providing meaningful results, etc.

We give special attention to Dede et al.'s [1] paper and insights to understand the existing OppNet simulation tools. This up-to-date paper organizes a detailed description and assessment of the various tools, mobility models, and protocols focused on routing and data dissemination techniques.

2.3.1 Performance Evaluation

To successfully perform an evaluation, we must first define our goals and the system. Establishing these evaluation goals is critical to this evaluation process. It will help set the type of evaluation, define the system and the load, and select the metrics to obtain. For example, the goal can be to compare different data propagation protocols or technologies, so we must evaluate in different scenarios and under different loads to determine which contexts they perform better. On the other hand, if our goal is designing a solid new infrastructure, the evaluation must be performed using a load similar to the expected one in the real deployment scenario. Therefore, the performance evaluation process consists of three main elements, as shown in Figure 6:

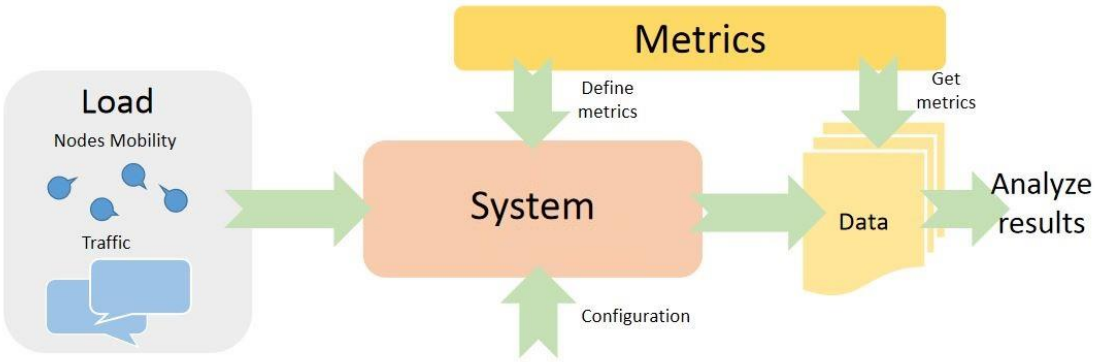

Figure 6: Performance evaluation process of the Simulator. Extracted from [1].

In detail, we have:

- **System**: The system, in our case, is the OppNet to be evaluated (integrated into the IoV) and comprises all its elements from hardware to software that can affect the performance of these networks, such as communication range, data transmission speed, buffer capacity, overhead, user behavior, etc.
- **Load**: The load (or workload) represents the type and quantity of requests in a system. The requests are generated by the network nodes that are embedded with devices that transmit messages. More precisely, we can distinguish between:
    - **Nodes mobility**: defining the movement of the nodes and their interactions are crucial for the evaluation of the network. Nodes' actions are usually restricted to the scenario to evaluate. The complexity of these scenarios can range from simple restricted square areas to realistic map-based scenarios.
    - **Traffic load**: that is, all messages and network requests generated by the nodes. This traffic load can be as simple as the diffusion of one message to all nodes in the network to more complex messaging patterns resembling the use of social messaging mobile applications.
- **Metrics**: Finally, we also need to define the metrics to evaluate the performance of the network. These metrics mainly evaluate the information diffusion in terms of the dissemination time, delivery rate, or the number of hops. Concerning the nodes and the network infrastructure, we can also obtain metrics about buffer occupation, network overhead, energy consumption, etc. When the performance evaluation is done, we usually have a large amount of data that must be processed to get the desired metrics and proceed with the analysis of the results. Regarding the statistical nature of the metrics, we can only obtain deterministic values (for example, the mean or average values), or we can determine its stochastical distribution. In the following subsection, we present in detail what metrics will be considered for this work.

The characteristics of the previous elements depend on the evaluation technique, such as real-world measuring, simulating, or creating analytical models. In our case, we will use the simulation method, which is usually a simplified model of a system and the load implemented in software. A

common approach is to combine a network simulation tool with real mobility traces to reproduce mobile nodes' real dynamics and interaction. Nevertheless, this simulation can be computationally intensive.

2.3.2 Performance Metrics

Performance metrics provide the means to evaluate the performance of a given system. The type of metrics to use in assessing a system differs from system to system. Due to the nature of OppNets, some used metrics have more or lesser importance than some others. In the following, a list of widely used metrics are presented together with a brief explanation:

- **Delivery Rate**: The delivery ratio provides a metric of how many packets were delivered to an intended recipient before the packet was removed from the network. The main reason for removing a packet from the network is when the packet reaches its expiration time or hop limit.

- **Latency or Delivery Delay**: The delivery delay provides a metric of how fast a message can be delivered to an intended recipient (or a group of recipients) considering a per-node or per-network scope. For example, in case of emergency messages, it is always critical to know the speed of data propagation. On the other hand, other scenarios require knowing how fast the data is propagated to each user depending on different node densities, mobility patterns, user preferences, etc.

- **Overhead**: This shows the overhead of the algorithms and protocols used as follows.

    - **Overhead of irrelevant data and duplicated data per node**: In OppNets, nodes also receive uninterested data and duplicated copies. Therefore, the percentage of the number of irrelevant data received/sent and the number of copies received with regard to the total number of data received/sent indicates the overhead per node.

    - **Overhead of cryptography**: Cryptographic system implementations always require additional overheads such as increased message size or more computational operations. Consequently, this overhead must be analyzed to understand how cryptographic mechanisms impact the network.

    - **Overhead of cache usage**: This shows the cache utilization concerning the total memory space available in the node. For example, after receiving the data, even though the node is not interested in it, it might have to store and carry it until another node is encountered because others might be interested in it (cooperation principle). Consequently, storing non-desired data will also cause an overhead in cache usage for irrelevant data vis-à-vis the preferences of a node.

    - **Overhead of energy consumption**: The percentage of cache usage for irrelevant data will end up causing overhead of energy consumption as well.

2.3.3 Simulator Selection

Notably, Dede et al. [1] reviewed and compared four reliable and open-source simulation tools widely accepted by the scientific community, explicitly: Adyton, OMNeT++, ONE, and NS3. Dede et al. defined OppNets as the set of applications and services running on end-user devices, such as OBUs, that use direct communication opportunities to exchange information. Subsequently, they have classified various applications according to their usage. Table 3 presents and classifies some examples for applications using unicast, multicast, and broadcast.

Table 3: Application Examples for Motivating and Evaluating OppNets (based on Dede et al. [1]).

Classification	**Unicast or Multicast (Destination-oriented)**	**Broadcast (Destination-less)**
User-driven	Announcements, messaging, chatting	Crowdsourcing, people as sensors, recommendations, public announcements
Automatic/ event-based	Security and safety, newsletters, health alerts (allergies, smoke, etc.)	Public safety alert (floods, intense solar radiation, etc.)
Automatic/ periodic	Vehicular monitoring, critical device status	Public reports (traffic, weather, health)

This simple classification is crucial for identifying and characterizing the best data propagation approaches and the metrics to consider depending on the application's class. For example, the simulator implemented protocols designed to serve destination-oriented applications will most likely not perform well in destination-less applications and vice-versa.

Table 4, shows a high-level comparison between the simulators. One may notice that C++ is the preferred programming language due to its fast performance in completing the simulations, compared with Java [1]. Besides time efficiency, we are also concerned with having good traffic models and the SOTA communication technologies to provide a more detailed and realistic analysis of various traffic scenarios using different technologies to exchange messages. Consequently, we consider OMNeT++ and NS3 to be the strongest candidates because, besides offering support for communication technologies, they can contemplate more complex scenarios where radio propagation and interference models are used as an approach to the real-cases and provide more accurate results.

However, we must acknowledge that approaches to real-case scenarios are computationally intense and time-consuming.

Compared to OMNeT++, the most significant disadvantage of NS3 is that the simulator structure is complex, making it hard to build and modify simulations customized to the IoV. NS3 also lacks a standard user interface with debugging and visualization options. On the other hand, OMNeT++ has a sophisticated user interface, with many possibilities for visualizing and inspecting various scenarios while maintaining outstanding performance. Furthermore, a clear separation between node behavior and node parameters makes OMNeT++ simulation scenarios relatively easy to construct.

Nevertheless, we will use NS3 to carry out our simulations. Compared to OMNet++, we consider NS3 to be the ideal candidate because in addition to support models based on the standards IEEE

802.11p and IEEE 1609.4 DSCR/WAVE; it is the only one that supports models based on the 5G mmWave [109], i.e., the link technologies enabled by the network devices (NetDevices).

Table 4: High-level comparison between simulators (extracted from [1]).

Tool	Adyton	ONE	OMNeT++	ns-3
Platforms	Linux	Java (JDK 6+)	Win, Linux, Mac	Linux, Mac, FreeBSD
Programming language	C++	Java	C++, NED	C++, Python
Parallelizable	-	-	+	o
BSD / Linux API compatibility	-	-	-	+ (DCE framework)
Documentation	+	+	++	+
Mailing lists and tutorials	o	++	++	+
User interface	-	+	++	-
Mobility models	o	+	++	o
Radio propagation models	-	o	+	+
Interference models	-	o	+	+
Link technologies	-	-	+	+
OppNets data propagation models	++	++	o	o
User behavior models	-	-	-	-
Traffic models	++	++	++	++
Energy consumption models	-	o	+	+
Battery models	-	o	+	++
Scalability	+	-	+	o

- no support, o partial support, + adequate support, ++ well supported

Moreover, NS3 can perform simulations based on real-traffic mobility datasets, thereby removing the need for complex mobility models. Dede et al. [1] concluded that the simulations using real mobility traces are very time-consuming. Their simulations took 20 times longer to complete than the same simulations using a random mobility model. However, they suggest using a hybrid mobility model supported on both real mobility traces and mobility models to obtain more reliable results towards real scenario implementations.

2.3.4 NS3 Network Simulator

The NS3 is a discrete-event network simulator [110]. It was developed within the NS3 project, started in 2006. The first release was in June 2008, and the project aims to maintain an open environment for researchers to contribute and share their software.

The core of NS3 and all its modules are written in C++. Additionally, it also provides a Python wrapper for the simulation core and all its APIs. Thus, the simulation scenario scrips can be written in C++ and Python. The core is highly modular and flexible, made of standard components across all protocol, hardware, and environmental models [1].

Finally, we can argue that NS3's communication modules have been designed to match the interfaces to the standard Linux Application Programmable Interfaces (APIs). For this reason, it is possible to move a simulation setup to the real world and vice versa, which makes NS3 very attractive and worth investing effort to learn, build and deploy the various meticulous IoV scenarios.

3. NTRU PRE Scheme

In this section, we architect our PRE scheme to best suit the IoV environment, offering end-to-end security and data dissemination efficiency. For this purpose, we require a solid cryptographic system enjoying the following PRE capabilities: multi-hop, collusion-resistant, non-transitive, non-interactive, quantum-safe, and computationally inexpensive.

We envision and plan for the futuristic IoV where quantum attacks are a reality. The data dissemination can be attained through multiple IoV nodes (i.e., vehicles and RSUs) working cooperatively towards common goals. Multi-hop allows nodes to count on each other (on proxies) to relay ciphertexts to reach some otherwise unreachable destination. Consequently, we need to ensure that unauthorized parties will never gain access to the ciphertexts by conspiring between themselves, i.e., collusion-resistance. In addition, we must also ensure that the proxy node cannot redelegate ciphertexts in an unauthorized manner through RK inspection and guessing, i.e., non-transitivity. Non-interactivity is essential because the SKs must always be kept private. Thus, the RK generation algorithm must not require the target node, delegatee, SK.

To embrace all the above-stated PRE properties, we have selected Nunez et al. [2] bidirectional and multi-hop PRE scheme based on NTRU, a widely known cryptosystem. This PRE scheme is defined as NTRUReEncrypt, and it is a version based on the variant of NTRU proposed by Stehlé and Steinfeld [111], relishing IND-CPA security under the proven hardness of the ring-LWE problem. It also uses moderate key sizes, excellent asymptotic performance, and conjectured resistance to quantum computer attacks. Furthermore, we expect to obtain the best computational speed, efficiency, and low resource consumption. As shown in [2], the NTRUReEncrypt system outperforms all others by one order of magnitude.

3.1 NTRUReEncrypt

In this subsection, the NTRU cryptosystem is analyzed and explained in detail. In this way, we will understand, implement, modify and assess the NTRUReEncrypt scheme.

3.1.1 History and Motivation

In 1994, Peter Shor developed a quantum algorithm for integer factorization and the discrete logarithm problem, making RSA, ECC, and the Diffie-Hellman key exchange breakable by a quantum computer. Therefore, new problems difficult to solve by quantum computers are a must. For example, RSA is considered secure in the classical setting because factoring is computationally hard. As we will observe, the NTRU is deemed secure in the classical setting because its underlying lattice problem is computationally hard. Moreover, it is considered secure in the quantum setting because there are no efficient quantum attacks against it.

The NTRU cryptosystem was first proposed in 1996 and published in 1998 by Hoffstein et al. [112]. Since then, in the literature, we can find several descriptions of the NTRU cryptosystem. For example, some researchers describe the inherent NTRU's lattice cryptography [113], others define

NTRU more generally [111], or compare it to other widely used cryptosystems while highlighting NTRU's general advantages [114]. Also, we can find efficient implementations for the VANET environment [115]. From another perspective, the NTRU is also the target of various researches aiming to find and establish well-known parameter sets to ensure that the cryptosystem will work and provide the desired performance under certain security levels [116], [117]. Finally, we can also find NTRU speed records for its implementation on real hardware such as an FPGA or a GPU using the CUDA platform [118], demonstrating the tremendous advantage of using NTRU's lattice operations in a parallelized fashion to gain speedups of around three to five orders of magnitude when compared to ECC and RSA, with similar security levels.

3.1.2 Algorithm Specifications and Limitations

The original NTRU is a PKE cryptosystem based on polynomials in the polynomial convolution ring [119], also known as the quotient ring, $\boldsymbol{R_{NTRU}} = \boldsymbol{Z}[\boldsymbol{x}]/(\boldsymbol{x^n} - 1)$, where $\boldsymbol{n}$ is a prime parameter (non-secret). Therefore, elements of the ring $\boldsymbol{R_{NTRU}}$ are integer polynomials of degree less than $\boldsymbol{n}$. The operators + and * denote addition and multiplication in the ring, respectively. The addition of polynomials in $\boldsymbol{R}$ is simple polynomial addition. Multiplication is polynomial multiplication followed by reduction $mod\ \boldsymbol{x^n} - 1$, (also known as taking the convolution product of the coefficient vectors). The two additional main parameters of NTRU are the integer $\boldsymbol{q}$, an integer that is a power of 2 and known as the large modulus to which each coefficient is reduced (non-secret), and the small polynomial $\boldsymbol{p} \in \boldsymbol{R_{NTRU}}$, which usually takes values $\boldsymbol{p} = 3$ or $\boldsymbol{p} = \boldsymbol{x} + 2$, and is known as the small modulus to which each coefficient is reduced (non-secret). Overall, the polynomial operations are performed in the ring $\boldsymbol{R_{NTRU}}$ modulus $\mathbf{q}$ or $\mathbf{p}$, denoted correspondingly $\boldsymbol{R_{NTRU}/q}$ and $\boldsymbol{R_{NTRU}/p}$.

The SK consists of a polynomial $\boldsymbol{f} \in \boldsymbol{R_{NTRU}}$ chosen at random, with a determined number of coefficients equal to 0, -1, and 1. The polynomial $\boldsymbol{f}$ must have an inverse in both $\boldsymbol{R_{NTRU}/q}$ and $\boldsymbol{R_{NTRU}/p}$, respectively, $\boldsymbol{f_q^{-1}}$ and $\boldsymbol{f_p^{-1}}$. For efficiency, $\boldsymbol{f}$ is chosen to be consistent to 1 modulus $\boldsymbol{p}$. The PK consists of a polynomial $\boldsymbol{h} = \boldsymbol{p} * \boldsymbol{g} * \boldsymbol{f_q^{-1}}\ mod\ \boldsymbol{q}$, where $\boldsymbol{g} \in \boldsymbol{R_{NTRU}}$ is a polynomial chosen at random (secret but discarded after initial use).

The encryption process is quite simple: a plaintext $\boldsymbol{M}$ from message space $\boldsymbol{R_{NTRU}/p}$ is encrypted to ciphertext $\boldsymbol{C} = \boldsymbol{h} * \boldsymbol{s} + \boldsymbol{M}\ mod\ \boldsymbol{q}$, where $\boldsymbol{s}$ is a small random polynomial in $\boldsymbol{R_{NTRU}}$, known as the "blinding" polynomial (secret but discarded after initial use). The decryption process of the ciphertext $\boldsymbol{C}$ starts with computing $\boldsymbol{C'} = \boldsymbol{f} * \boldsymbol{C}\ mod\ \boldsymbol{q}$. If the resulting polynomial $\boldsymbol{C'}$ is sufficiently small, then the result of the reduction is $\boldsymbol{p} * \boldsymbol{g} * \boldsymbol{s} + \boldsymbol{f} * \boldsymbol{M} \in \boldsymbol{R_{NTRU}/q}$. Subsequently, $\boldsymbol{C'}$ is reduced modulus $\boldsymbol{p}$, obtaining $\boldsymbol{f} * \boldsymbol{M}$. Lastly, this result is multiplied by $\boldsymbol{f_p^{-1}}$ to extract the original message $\boldsymbol{M}$; note that the last step is redundant if $\boldsymbol{f}$ is consistent to 1 modulus $\boldsymbol{p}$.

The NTRUReEncrypt is based on the NTRU cryptosystem and extends it by adding the definitions of the re-encryption and re-encryption key generator (RKG) algorithms. Also, the secret polynomial $\boldsymbol{f}$ has to be congruent to 1 modulus $\boldsymbol{p}$, because it is necessary to correctly decrypt re-encrypted ciphertexts. In [2]**,** the NTRUReEncrypt scheme is presented by the authors, where a proxy is given a re-encryption key $\boldsymbol{rk_{A \to B}}$ that allows it to translate ciphertext $\boldsymbol{C_A}$, i.e., the message $\boldsymbol{M}$ encrypted under

Alice's PK, $\boldsymbol{pk_A}$, into a re-encrypted ciphertext $\boldsymbol{C_B}$, which is the same message $\boldsymbol{M}$ decryptable by Bob's SK, $\boldsymbol{sk_B}$.This scheme consists of the following five algorithms:

- **KeyGen(**1^k**)**: On input the security parameter $\boldsymbol{k}$, the output of key generation algorithm for Alice is (sk_A, pk_A), where $\boldsymbol{sk_A} = (\boldsymbol{f_A}, \boldsymbol{g_A})$ and $\boldsymbol{pk_A} = \boldsymbol{h_A}$. Let $\boldsymbol{f_A^{-1}}$ denote the inverse of $\boldsymbol{f_A}$ in the ring $\boldsymbol{R_{NTRU}/q}$.
- **ReKeyGen(**sk_A, sk_B**)**: On input the SK of Alice, $\boldsymbol{sk_A}$ and the SK of Bob, $\boldsymbol{sk_B}$, the algorithm ReKeyGen (RKG) computes $\boldsymbol{rk_{A\to B} = f_A * f_B^{-1}\ mod\ q}$, which is the RK between Alice and Bob. The RK can be computed by a simple protocol originally proposed in [48], as follows: Alice selects $\boldsymbol{r} \in \boldsymbol{R_{NTRU}/q}$ and sends $\boldsymbol{r * f_A\ mod\ q}$ to the proxy, then Bob sends $\boldsymbol{r * f_A * f_B^{-1}\ mod\ q}$ to the proxy, so that it can compute $\boldsymbol{rk_{A\to B} = f_A * f_B^{-1}\ mod\ q}$.
- **Enc(**pk_A, M**)**: On input the PK of Alice, $\boldsymbol{pk_A}$ and the message $\boldsymbol{M} \in \boldsymbol{R_{NTRU}/p}$, the encryption algorithm generates a small random polynomial $\boldsymbol{s} \in \boldsymbol{R_{NTRU}}$, and outputs the ciphertext $\boldsymbol{C_A = h_A * s + M\ mod\ q}$.
- **ReEnc(**$rk_{A\to B}$, C_A**)**: On input of $\boldsymbol{rk_{A\to B}}$ and the corresponding ciphertext $\boldsymbol{C_A}$, the re-encryption algorithm ReEnc generates a random polynomial $\boldsymbol{e} \in \boldsymbol{R_{NTRU}}$ and outputs ciphertext $\boldsymbol{C_B = C_A * rk_{A\to B} + p * e\ mod\ q}$.
- **Dec(**sk_B, C_B**)**: On input, Bob's secret key $\boldsymbol{sk_B = f_B}$ and a ciphertext $\boldsymbol{C_B}$, the decryption algorithm Dec computes the polynomial $\boldsymbol{C'_B = C_B * f_B\ mod\ q}$ and outputs the final polynomial $\boldsymbol{M = C'_B\ mod\ p}$, which is the original message $\boldsymbol{M} \in \boldsymbol{R_{NTRU}/p}$ sent by Alice.

According to [2]**,** we would like to emphasize the following details:

- The message $\boldsymbol{M}$ is a polynomial padded with random bits and masked according to a hamming weight restriction. Thus all messages are trinary polynomials, where the three coefficient options are $\boldsymbol{d_{+1}}$**,** $\boldsymbol{d_{-1}}$ and $\boldsymbol{d_0}$ which are represented by the following integer numbers: +1,-1, and 0, respectively. For security and integrity purposes, each coefficient option has minimum quantity requirements given by the parameter $\boldsymbol{d_m}$.
- The random term $\boldsymbol{e}$ included in the ReEnc algorithm is known as the error term and is necessary to prevent a simple ciphertext attack from the delegatee. Envisage that a re-encrypted ciphertext is of the form $\boldsymbol{C_B = C_A * rk_{A\to B} = C_A * f_A * f_B^{-1}}$; that is without the term $\boldsymbol{e}$. Consequently, the delegatee could easily extract the SK of the delegator based on the observation that $\boldsymbol{C_A * f_A = C_B * f_B}$ is obtained by simply rearranging the equation terms. Finally, assuming that $\boldsymbol{C_A}$ is invertible modulus $\boldsymbol{q}$, the attacker can extract the SK by computing $\boldsymbol{f_A = C_A^{-1} * C_B * f_B}$.
- The scheme is multihop because it supports multiple re-encryptions. Yet, this property is limited since adding the error term during the ReEnc algorithm produces an increasing error that grows on each hop and reflects on the resulting ciphertext $\boldsymbol{C_B}$. The Dec algorithm tolerates a certain amount of accumulated error. Nevertheless, in all cases, the decryption will eventually fail after a certain number of re-encryptions. Our experiments show that this limitation heavily depends on the chosen NTRUReEnc parameters. Later on, we will discuss and present the results in more detail.

3.1.3 Standard Parameter Sets

Different parameter sets can be used for the NTRUReEncrypt cryptosystem. The research in [116] and [117] explains the importance of choosing the proper parameter sets. To ensure functionality, we use the secure standard settings implemented by default, shown in Figure 7. On the next page, we put Figure 7 under the spotlight so that we can clearly understand how these parameter sets are fed into the coded NTRUReEncrypt cryptosystem. Also, we can notice that all parameters are commented, providing a reasonable explanation for their existence.

Table 5 shows the considered named parameter sets in the form (n, prod, k, p, q), where:

- **n/p/q**: The number of polynomial coefficients/ small modulus/ large modulus.
- **prod**: A Boolean value to enable/disable (i.e., True/False) the use of product form polynomials, respectively. We fix and hide the value set to True as it performs faster computation.
- **k**: The security level (in bits), 128 (uses SHA 256), and 256 (uses SHA 512).

Table 5: Named parameter sets.

Parameters (n, k, p, q)	Name
439, 128, 3, 2048	APR2011_439_FAST
1087, 256, 3, 2048	EES1087EP2_FAST
1171, 256, 3, 2048	EES1171EP1_FAST
1499, 256, 3, 2048	EES1499EP1_FAST

The heart of the NTRU-based cryptosystems is the multiplication of polynomials, which can be supported on the Fast Fourier Transform (FFT) to improve its computational performance. For that reason, we will only consider the “FAST” parameter sets since they enable the use of the FFT and run much faster, i.e., require less processing time to complete.

```
/**
 * Constructs a parameter set that uses product-form private keys (i.e. </code>polyType=PRODUCT</code>).
 * @param N number of polynomial coefficients
 * @param q modulus
 * @param df1          number of ones in the private polynomial <code>f1</code>
 * @param df2          number of ones in the private polynomial <code>f2</code>
 * @param df3          number of ones in the private polynomial <code>f3</code>
 * @param dm0          minimum acceptable number of -1's, 0's, and 1's in the polynomial <code>m'</code> in the last encryption step
 * @param maxM1        maximum absolute value of mTrin.sumCoeffs() or zero to disable this check. Values greater than zero cause the constant coefficient of the message to always be zero.
 * @param db           number of random bits to prepend to the message; should be a multiple of 8
 * @param c            a parameter for the Index Generation Function ({@link IndexGenerator})
 * @param minCallsR    minimum number of hash calls for the IGF to make
 * @param minCallsMask minimum number of calls to generate the masking polynomial
 * @param hashSeed     whether to hash the seed in the MGF first (true) or use the seed directly (false)
 * @param oid          three bytes that uniquely identify the parameter set
 * @param sparse       whether to treat ternary polynomials as sparsely populated ({@link SparseTernaryPolynomial} vs {@link DenseTernaryPolynomial})
 * @param fastFp       whether <code>f=1+p*F</code> for a ternary <code>F</code> (true) or <code>f</code> is ternary (false)
 * @param hashAlg      a valid identifier for a <code>java.security.MessageDigest</code> instance such as <code>SHA-256</code>
 */
EncryptionParameters(int N, int q, int df1, int df2, int df3, int dm0, int maxM1, int db, int c, int minCallsR, int minCallsMask, boolean hashSeed, byte[] oid, boolean sparse, boolean fastFp, String hashAlg)
/**
 * A set of parameters for NtruEncrypt and NtruReEnc. Several predefined parameter sets are available and new ones can be created as well.
 */
/** A conservative (in terms of security) parameter set that gives 256 bits of security and is optimized for key size. */
EES1087EP2 = EncryptionParameters(1087, 2048, 120, 120, 0, 256, 13, 25, 14, true, new byte[] {0, 6, 3}, true, false, "SHA-512");
/** A product-form version of <code>EES1087EP2</code> */
EES1087EP2_FAST = EncryptionParameters(1087, 2048, 8, 8, 11, 120, 0, 256, 13, 25, 14, true, new byte[] {0, 6, 3}, true, true, "SHA-512");
/** A conservative (in terms of security) parameter set that gives 256 bits of security and is a tradeoff between key size and encryption/decryption speed. */
EES1171EP1 = EncryptionParameters(1171, 2048, 106, 106, 0, 256, 13, 20, 15, true, new byte[] {0, 6, 4}, true, false, "SHA-512");
/** A product-form version of <code>EES1171EP1</code> */
EES1171EP1_FAST = EncryptionParameters(1171, 2048, 8, 7, 11, 106, 0, 256, 13, 20, 15, true, new byte[] {0, 6, 4}, true, true, "SHA-512");
/** A conservative (in terms of security) parameter set that gives 256 bits of security and is optimized for encryption/decryption speed. */
EES1499EP1 = EncryptionParameters(1499, 2048, 79, 79, 0, 256, 13, 17, 19, true, new byte[] {0, 6, 5}, true, false, "SHA-512");
/** A product-form version of <code>EES1499EP1</code> */
EES1499EP1_FAST = new EncryptionParameters(1499, 2048, 7, 6, 11, 79, 0, 256, 13, 17, 19, true, new byte[] {0, 6, 5}, true, true, "SHA-512");
/** A parameter set that gives 128 bits of security and uses simple ternary polynomials. */
APR2011_439 = new EncryptionParameters(439, 2048, 146, 130, 126, 128, 12, 32, 9, true, new byte[] {0, 7, 101}, true, false, "SHA-256");
/** Like <code>APR2011_439</code>, this parameter set gives 128 bits of security but uses product-form polynomials and <code>f=1+pF</code>. */
APR2011_439_FAST = new EncryptionParameters(439, 2048, 9, 8, 5, 130, 126, 128, 12, 32, 9, true, new byte[] {0, 7, 101}, true, true, "SHA-256");
/** A parameter set that gives 256 bits of security and uses simple ternary polynomials. */
APR2011_743 = new EncryptionParameters(743, 2048, 248, 220, 60, 256, 12, 27, 14, true, new byte[] {0, 7, 105}, false, false, "SHA-512");
/** Like <code>APR2011_743</code>, this parameter set gives 256 bits of security but uses product-form polynomials and <code>f=1+pF</code>. */
APR2011_743_FAST = new EncryptionParameters(743, 2048, 11, 11, 15, 220, 60, 256, 12, 27, 14, true, new byte[] {0, 7, 105}, false, true, "SHA-512");
```

Figure 7: NTRUReEnc Parameter Set

3.1.4 Algorithm's Performance

In view of testing the five algorithms' performance (i.e., EncKeyGen, ReEncKeyGen, Enc, Dec, and ReEnc), we have created suitable bench table scripts which run several iterations for each algorithm and display the results in microseconds accounting for the following metrics: minimum time, average time, operations per second. Table 6 presents the results of our tests performed on a computer with the following specifications:

- Hardware: 16 GB of RAM and CPU Intel(R) Core(TM) i5-8250U @ 1.60GHz.
- Software: comprising the application and the operating system:
    - Operating System: Windows 10 Home, version 20H2, 64-bit based.
    - Application: Eclipse IDE for Java Developers, version 2021-06 (4.20.0).

Each test is composed of five isolated runs of the five NTRUReEncrypt algorithms. Each run executes the algorithms with a different number of iterations, described in Table 5. We can notice that the key generator algorithms (i.e., EncKeyGen, ReEncKeyGen) have a lower number of iterations because we are not interested in using such results since, for this work, we consider that all nodes already have their keys generated.

Table 5: Number of iterations per algorithm.

Algorithm	EncKeyGen	ReEncKeyGen	Enc	Dec	ReEnc
# iterations	10	10	50	50	50

Moreover, the different trials have different message sizes, correspondingly, 80, 550, and 1100 bytes. For comparison purposes, we use the same parameter set, i.e., EES1171EP1_FAST, used by Nunez et al. [2] and only consider the metrics' total average values. Table 6 reveals the trial's results for the five crypto functions, measuring the processing time required for each message.

By comparing with the results obtained in [2], it is noticeable that our crypto functions ran much faster. This is straightforward when considering that the operations run in a more favourable environment, i.e., our configuration setup is more recent and has more computational resources.

Table 6 presents the total average computation time in µs of NTRUReEncrypt's three main algorithms, namely Encrypt, Decrypt, and ReEncrypt, which will be used as the reference values in our simulations to impose the cryptosystem processing overheads. As said above, the other two algorithms, EncKeyGen and ReEncKeyGen, do not have a crucial role in this work since they are used only for key generation. Please note that we assume that key generation and management processes are already implemented, assuring that all nodes wishing to communicate can do so. Consequently, all key sharing and processing overheads are disregarded in our simulations. Moreover, we focus on the parameter set EES1171EP1_FAST because it was already used in previous NTRU benchmarks [118], [2]**.**

Finally, it is interesting to note that the results are almost identical regardless of the message size. This is the expected behaviour considering that the message is always padded to have the same final length, depending only on the preferred parameter set. As we will see in the following subsections, each parameter poses specific limitations for which messages (plaintexts) and ciphertexts must account.

Table 6: Computation Performance of NTRUReEncrypt.

Parameter Set: EES1171EP1_FAST						
Msg Size	**Metric**	**EncKeyGen**	**ReEncKeyGen**	**Encrypt**	**Decrypt**	**ReEncrypt**
80 bytes	**Min Time (μs)**	8371	7831	73	457	469
	Av Time (μs)	8833	8146	107	558	559
	Operations/sec	113.20	122.75	9270.42	1789.17	1787.96
550 bytes	**Min Time (μs)**	8599	8216	73	492	457
	Av Time (μs)	8853	8374	89	853	565
	Operations/sec	112.95	119.42	11216.04	1713.70	1769.84
1100 bytes	**Min Time (μs)**	8801	8049	82	443	478
	Av Time (μs)	9002	8541	120	495	583
	Operations/sec	111.08	117.08	8295.24	2019.09	1715.14
Total Av Time (Ref. Value)		8896	8354	105	545	569

3.1.5 Message and Ciphertext Specifications

The message is built as an array of characters (i.e., chars). For example, let us consider the message $\boldsymbol{M}$: “Hello World!”, i.e., an array of 12 chars. Now, bearing in mind that in Java, each char is 2-byte or 16 bits, we can conclude that message $\boldsymbol{M}$ total size is $12 * 2 = 24\ bytes$.

Subsequently, NTRUReEncrypt converts the char array into a trinary polynomial, represented by an array of integers (i.e., the polynomial coefficients). This conversion and padding follow the rules in the IEEE P1363.1 standard for Public-Key cryptographic techniques based on hard problems over lattices [117].

The specified methods for binary to trinary conversion and vice-versa are visible in Figure 8, where the trinary combination {-1, -1} is not used. The binary to trinary function starts by converting each three-bit quantity to two ternary coefficients, then concatenates the resulting ternary quantities to obtain the final output, i.e., the integer array that forms the polynomial coefficients. The trinary to binary function does the reverse; it starts by converting each set of two ternary coefficients to three bits, then concatenates the resulting bit quantities to obtain the final output, i.e., the original char array (representing the original message).

Please note that we cannot simply use these conversion operations and apply raw NTRU encryption to encrypt a message securely. Instead, the message must be pre-processed before encryption and post-processed after encryption to protect against active attackers who might modify the message in transit. Also, trinary polynomials allow the same security as binary polynomials but use shorter keys.

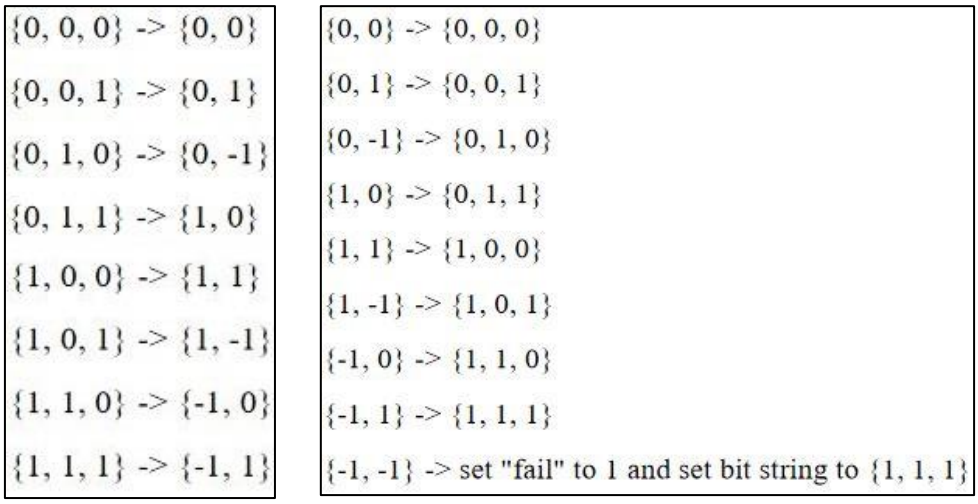

Figure 8: Encryption operation binary to trinary (left side)/ Decryption operation binary to trinary (right side).

3.1.6 Space Cost Analysis

Nunez et al. [2] theoretical analysis of the computational costs associated with NTRUReEncrypt shows that the main algorithms, namely encryption, decryption, and re-encryption, only need a single multiplication. More precisely, the re-encryption algorithm takes an additional multiplication for computing the term $\boldsymbol{p} * \boldsymbol{e}$, but this can be computed beforehand and, the polynomial $\boldsymbol{p}$ is small and known in advance. As stated before, the core operation of NTRU is the multiplication of polynomials, which can be done in $O(\boldsymbol{n} * log\ \boldsymbol{n})$ time using the FFT [118].

Regarding the space costs associated with NTRUReEncrypt, Table 7 presents a theoretical analysis of the size of the messages, keys, and ciphertexts and the measure of ciphertext expansion, i.e., the length increase of a plaintext when it is encrypted.

Table 7: Space costs of NTRUReEncrypt (extracted from [2]).

Element	Size
Keys	$O(\boldsymbol{n} * log_2\ \boldsymbol{q})$
Message	$O(\boldsymbol{n})$
Ciphertext	$O(\boldsymbol{n} * log_2\ \boldsymbol{q})$
Ciphertext expansion	$O(log_2\ \boldsymbol{q})$

We can perceive that the keys have the same size as the ciphertexts, i.e., correspond to polynomials functions with the same size and degree. Also, the message size depends on the parameter $\boldsymbol{n}$. We will stick to the EES1171EP1_FAST parameter set, which specifies $\boldsymbol{n} = 1171$, and supports up to 220 bytes of plaintexts. This limitation is flexible and varies according to the selected parameters. For example, the EES1499EP1_FAST specifies $\boldsymbol{n} = 1499$ supports up to 280 bytes plaintext.

Consequently, when the input plaintext overflows the maximum size limitation, the cryptosystem removes all the excess data and encrypts it. On the other hand, when the plaintext size is shorter than the maximum limit, it is padded to achieve it. This characteristic goes according to cryptosystem requirements and provides enhanced security.

The ciphertext size is always given by expanding the message size by a fixed value that only depends on the $\boldsymbol{q}$ parameter. Concerning all our parameter sets, with $\boldsymbol{q} = 2048$, the expansion factor is computed as:

$$\boldsymbol{Expansion\ Factor}_{\text{EES1171EP1\_FAST}} = log_2 2048 = 7.62$$

We can conclude that for the parameter set EES1171EP1_FAST:

- Maximum message/plaintext size = $\boldsymbol{220\ bytes}$.
- Constant Ciphertext size = $220 * 7.62 \cong \boldsymbol{1677\ bytes}$.

We will consider these results in our simulations, which will allow us to observe the impact of size overheads imposed in the communications.

It is essential to notice that all ciphertexts have the same size, independently from the number of re-encryptions. However, each re-encryption adds a certain amount of error (inherent to the cryptosystem when performing the calculations over the polynomial functions), which imposes limitations that we explain in the next subsection.

Furthermore, Nunez et al. [2] also compared other lattice-based cryptosystems, such as the PRE scheme from Aono et al. [120], where keys and messages are typically matrices whose dimensions grow linearly with the parameter $\boldsymbol{n}$. Overall, it is shown that NTRUReEncrypt cryptosystem has much lower space costs, even when using more security bits.

3.1.7 Re-encryption Limitation

Re-encryption is the process that enables converting a ciphertext destined to Alice into a ciphertext destined to Bob. Thus, re-encryption is fundamental to allow efficient multi-hop capabilities in the IoV environment. For this reason, we must study the re-encryption process that produces an increase in the error terms of the ciphertexts, which can potentially lead to decryption failures. With each re-encryption comes the associated error increment. The NTRUReEncrypt cryptosystem can handle a certain number of cumulative errors. In this subsection, we analyze the limit regarding the maximum number of possible re-encryptions.

Figure 9 shows the success rate of decrypting a ciphertext (vertical axis) after several re-encryptions (horizontal axis). We select the two sets that use a larger number of coefficients for this experiment, EES1171EP1_FAST, and EES1499EP1_FAST.

It can be seen that the decryption failure depends on the choice of parameters. Thus, overall, increasing the central parameter values will decrease the probability of decryption failure, which is not guaranteed due to the complex nature of how all the parameters interact [117].

In Figure 9, we can immediately observe that EES1171EP1_FAST has decryption failures much sooner than EES1499EP1_FAST. For example, EES1171EP1_FAST has 99% success until the 10th re-encryption, while EES1499EP1_FAST has 99% success until the 21st re-encryption.

As stated above, re-encryption is intrinsic to multi-hop capabilities, and it is essential to recognize these limitations when implementing the IoV architecture. We anticipate that the IoV environment will benefit from having at most ten hops, as studied by Chen et al. [116]. The results show that the maximum routing hop count parameter is a significant impact factor on the communication quality of the VANET.

Concerning our simulations, the hop number limitation will be imposed by setting the corresponding number in the Time to Live (TTL) within the IPv4 packet.

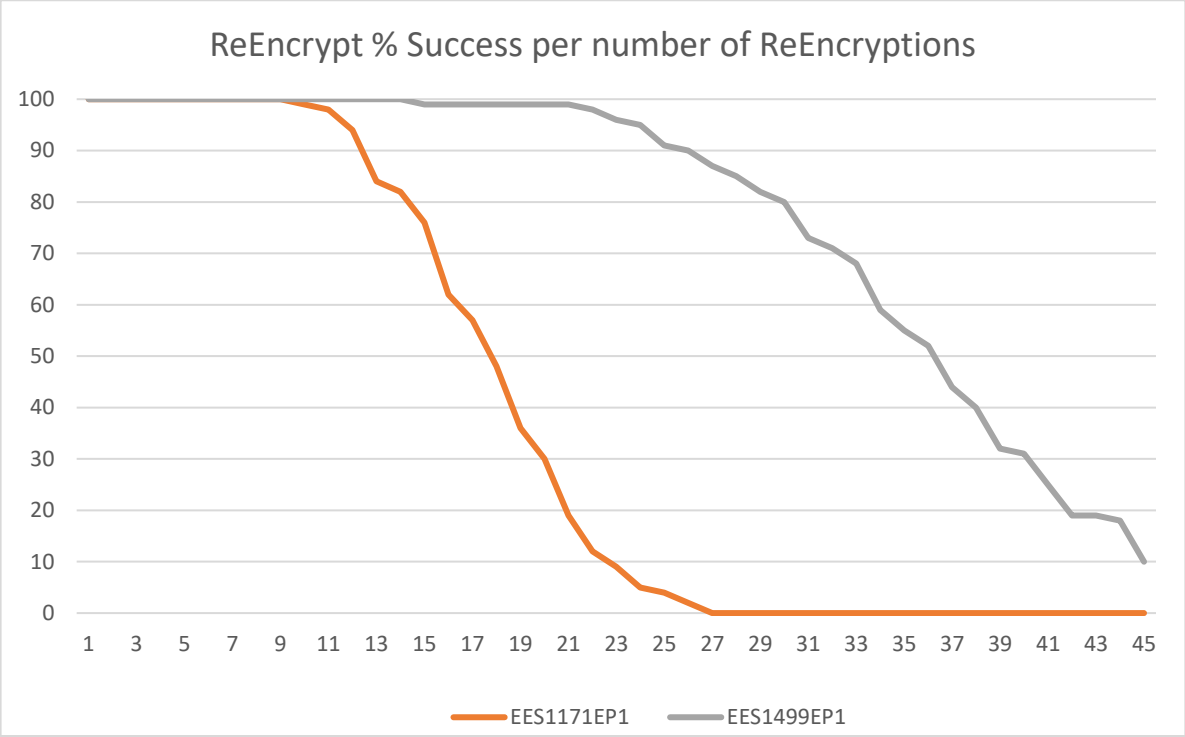

Figure 9: Re-encryption success rate.

3.2 NTRUReEncrypt Software Implementation

The NTRUReEncrypt cryptosystem is implemented in Software using the Eclipse IDE for Java Developers, version 2021-06 (4.20.0).

Specifically, we use an unbroken Java implementation of the NTRU PK cryptosystem, termed NTRUEncrypt. The BouncyCastle cryptography libraries support this implementation, which is based on the Java Cryptography Architecture (JCA) and the Java Cryptography Extension (JCE) frameworks. The NTRUEncrypt construction follows the IEEE P1363.1 standard for lattice-based public-key cryptography [117]. On top of NTRUEncrypt, we add and edit a prototype implementation in Java of the NTRUReEncrypt scheme proposed by Nunez et al. [2].

Various approaches can leverage the recommended hybrid cryptosystem concept, for example, by using a combination of the asymmetric NTRUReEncrypt and the symmetric AES cryptosystems to encrypt arbitrary-length messages.

In Figure 10, we can observe part of the NTRUReEncrypt Java implementation and the hierarchy previously described of the installed packages, correspondingly on the right and left sides. Please note that we are keen to share this customized implementation with fellow researchers to enhance it and continue studying it. For this reason, the figure may provide immediate support on how we organize our project in Eclipse IDE.

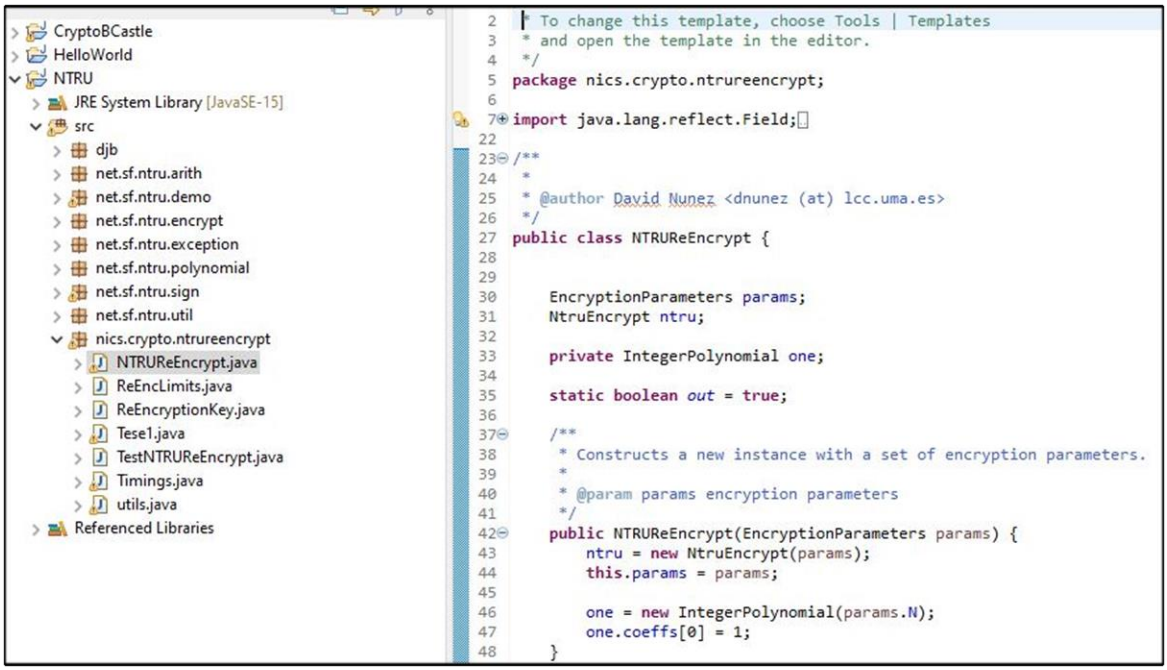

Figure 10: NTRUReEnc.java and Package Hierarchy

4. Network Architecture and Implementation

4.1 NS3 Architecture Overview

This chapter intends to let the reader understand the NS3 simulator and all its components. For this reason, we start by providing a more general introduction of the NS3 simulator and, in the following subsections, specify the modules and code we will use in our simulations.

As discussed above in this work, NS3 is a network simulator mainly implemented in C++ and provides bindings for Python [121].

Therefore, simulations for NS3 have to be written in one of these code languages. A simulation is realized by processing a discrete list of all events about to occur. This list of events is sorted ascendingly by the event occurrence time, managed by the NS3 event scheduler functions that we will later explain.

The NS3 provides several model types allowing the user to specify all the different components of a network. To run the simulation, the user has to create all network parts needed for the simulation with respect to the following categories [121]:

- **Node**: Represents a network element to which Applications, Protocol stack, and NetDevices can be added. Each NetDevice is attached to a Channel, over which it sends and receives Packets. A Node may be connected to more than one Channel through multiple NetDevices. In addition, a Node may use various protocol components, which handle Packets received by the NetDevice. Each Node can also run a list of Applications. The binding interfaces between the components of the node are designed to be similar to those in real systems. The interface between Application and Protocols is based on a class called Socket. The application uses this class to send and receive Packets to the Protocol stacks attached to a Node.
- **Device**: The device is also termed NetDevice, and represents the actual part of a network node that is responsible for network communications, like the ethernet card.
- **Packet**: Packets are designed to have the same format as the packets sent in real networks. Each Packet contains the following:
    - One Byte buffer to store serialized content of the headers and trailers added to the packet. It is expected to match that of real network packets bit for bit.
    - Metadata to describe the type of headers and trailers which were serialized in the byte buffer.
    - A set of packet tags containing simulation-specific information that cannot be stored in the packet byte buffer or could otherwise interfere with the protocols in use.
- **Channel**: A Channel is a logical path over which information (i.e., Packets) flows. It can represent a wired network cable or a wireless transmission medium. Each Channel object may be connected to a list of NetDevices.

In Figure 11, we can see a simple NS3 node architecture where two nodes are connected through a channel. This work will consider two types of nodes, i.e., the RSU and vehicle nodes. Furthermore,

each node contains two kinds of NetDevices, the Wave NetDevice (for the simulation of the WAVE system based on the 802.11p standard) and the mmWave NetDevice (for the simulation of 5G cellular networks operating at mmWaves - C-V2X).

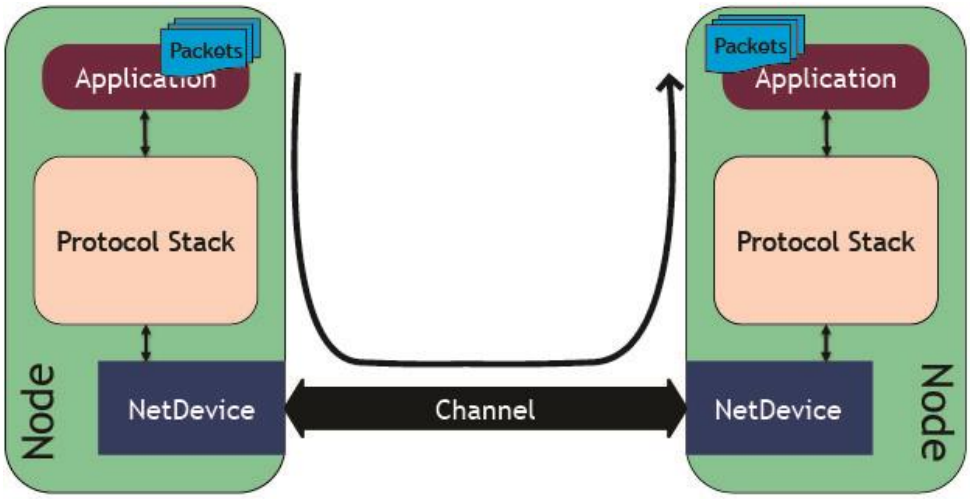

Figure 11: NS3 Key Components

4.2 WAVE Module

Regarding the implementation of the WAVE standards [122], NS-3 currently offers the following two options for the VANET environment:

- *802.11p MAC/PHY Layer* – Simple implementation with only Single-Channel operation, adaptation of the native 802.11a implementation, which can simulate the standard WAVE channel with 10MHz of bandwidth in OFDM at the specific 802.11p data rates.
- *Wave compliant MAC Layer* – Supports Multi-Channel operation and many other features from IEEE 1609.x protocols, such as BSM and WSA, extending the previous option, the 802.11p MAC/PHY.

As shown in Figure 12, this module adds new features on top of the existing Wi-Fi Module. Besides, we can observe the interactions between the different layers, starting from the base with the WifiChannel up to the WaveNetDevice. All these layers require complex configurations, and we will use all the typical ones based on the provided examples of the WAVE module, which are defined according to the standards. All the necessary documentation and source files can be retrieved from the developer's website [123].

It is vital to understand how this module works since we will have to modify it to satisfy our scenario requirements. One of the novelties in the WAVE operation when compared to the standard 802.11a is that it can establish a connection without using 802.11 authentication services operating outside of the context of Basic Service Set (OCB). This allows communication to happen quickly in rapidly carrying environments. This is why multiple channels are used: a control channel (CCH) typically manages connections and one or more service channels (SCHs) to send actual data to devices. WAVE devices can organize WAVE basic service sets (WBSSs), defining a specific SCH for their communication.

Spectrum allocation for these channels and other adopted parameters are shown in Table 8 [124]. We can perceive that the time parameters are within the same order of magnitude, in microseconds, as the processing time of the crypto functions used by the NTRUReEncrypt cryptosystem.

Table 8: Changes between 802.11p and 802.11a Physical Layer.

Parameter	802.11a Value	802.11p Value	Changes
Frequency Range	5.15-5.85 GHz	5.85-5.95 GHz	Different
Bandwidth	20 MHz	10 MHz	Halved
Bit rate (Mbps)	6, 9, 12, 18, 24, 36, 48, 54	3, 4.5, 6, 9, 12, 18, 24, 27	Halved
Guard Interval	0.8 µs	1.6 µs	Doubled
Preamble duration	16 µs	32 µs	Doubled
Symbol duration	4 µs	8 µs	Doubled
FFT period	3.2 µs	6.3 µs	Doubled

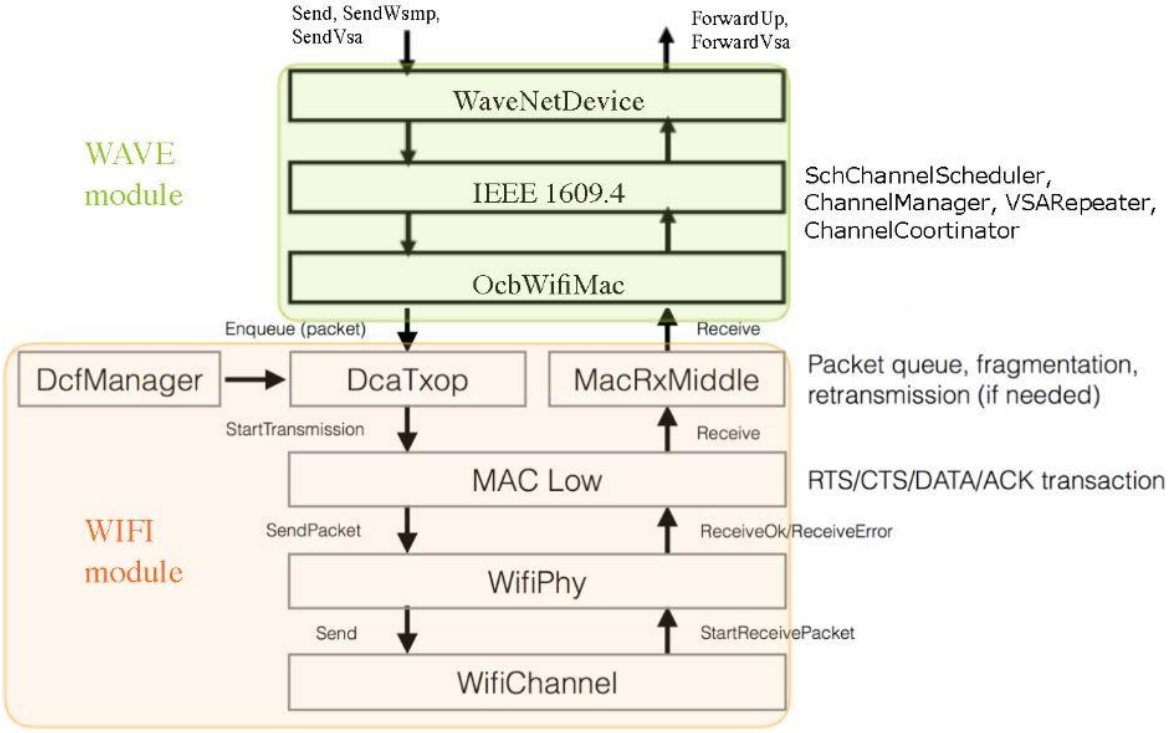

Figure 12: WAVE Module architecture

Overall, we need to recognize the complexity of the possible configurations and how the module is layered. In this way, we can precisely change the source code to best suit our simulations, for example, by acknowledging that the OCB concept is introduced with the *OcbWifiMac layer*, which allows stations to connect to each other when in direct range.

On top of that, IEEE 1609.4 stack is implemented, including *ChannelScheduler*, *ChannelManager,* and *ChannelCoordinator*, which handles the operations of alternating channels (between CCH and SCH). We are interested in adding the time overheads introduced by the crypto

functions. For this reason, it is essential to know where the schedulers are implemented. In our case, these schedulers take advantage of the NS-3 simulator event scheduler, which can be modified to add extra processing time.

4.3 C-V2X mmWave Module

The mmWave module provides a basic implementation of mmWave user and infrastructure devices, including the propagation models, PHY, and MAC. Inspired by the existing LTE module, this module has been designed to provide a completely customizable simulation tool for mmWave devices. Figure 13 shows a schematic diagram of the mmWave module.

We can adopt the default parameter values provided by the example source codes provided by the module developers [125]. Furthermore, this module is referenced in several papers describing in detail the implemented features, such as [126] and [127].

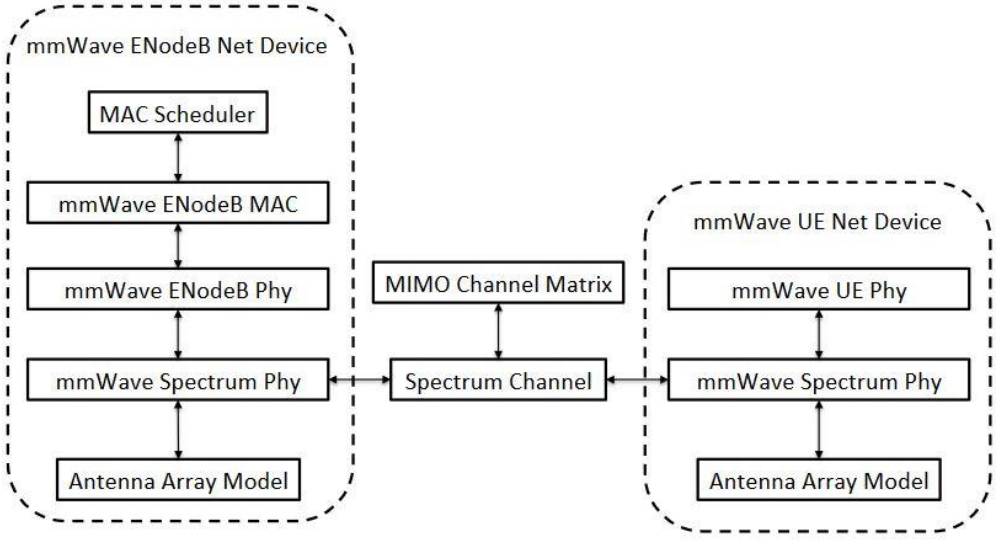

Figure 13: mmWave Module schematic, extracted from [128].

4.4 Collecting Statistics

Many approaches can be taken to obtain the NS3 statistics, giving results for different parts of the network. However, the main focus here is to collect results with the same granularity and at the same network level. For that purpose, three different ways of obtaining outcomes are explored in this work, as shown in Figure 14.

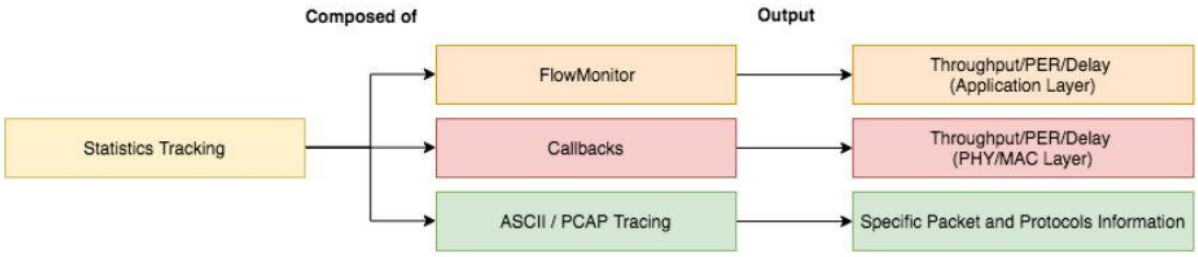

Figure 14: Statistics tracking components.

One valuable and customizable way of obtaining results is by using the *Callbacks* system and the *Simulator::Shedule* to call a specific method, for example, when a packet reception or a packet loss event happens. When using the schedule method, it is possible to make a particular piece of code run multiple times in parallel with the simulation steps, collecting, for instance, throughput per second. These statistics are collected at the PHY/MAC layer of the network stack and can be collected per link or node. Events being traced during simulation run-time are: packets dropped at PHY/MAC layer, packets received successfully or with error, and how much time the physical device spends on each state (receiving, sending, idle) to measure medium congestion. Furthermore, Schedule calls allow us to schedule events at precise times. For example, we can prepare a packet to send event at one time and schedule it to happen sometime in the future. In the following subsection, we will understand how to use this feature to impose time overheads in the network.

There is also a framework used to collect statistics on NS3 that was considered, named Flow Monitor [129]. This framework consists of multiple probes placed at different levels of simulation to follow the packet's path. The concept of flow here is considering the packets that have the same “five-tuple,” which is given by (source-IP, dest-IP, protocol, source-port, dest-port). Statistics for each detected flow are shown at the end of the simulation. Although this framework collects statistics, they work upon IP layer (IPv4 and IPv6) supporting L4 protocols (TCP or UDP). This tool is suitable for retrieving traffic information from UDP or TCP flows generated between two vehicles.

The data collected for each flow are [130]:

- timeFirstTxPacket: when the first packet in the flow was transmitted;
- timeLastTxPacket: when the last packet in the flow was transmitted;
- timeFirstRxPacket: when the first packet in the flow was received by an end node;
- timeLastRxPacket: when the last packet in the flow was received;
- delaySum: the sum of all end-to-end delays for all received packets of the flow;
- jitterSum: the sum of all end-to-end delay jitter (delay variation) values for all received packets of the flow, as defined in RFC 3393;
- txBytes, txPackets: total number of transmitted bytes / packets for the flow;
- rxBytes, rxPackets: total number of received bytes / packets for the flow;
- lostPackets: total number of packets assumed to be lost (not reported over 10 seconds);
- timesForwarded: the number of times a packet has been reportedly forwarded;

- delayHistogram, jitterHistogram, packetSizeHistogram: histogram versions for the delay, jitter, and packet sizes, respectively;
- packetsDropped, bytesDropped: the number of lost packets and bytes, divided according to the loss reason code (defined in the probe).

It is worth stressing out that the probes measure the packet bytes, including IP headers. The L2 headers are not included in the measure.

To help us visualize the statistics, we have printed them in XML format. In the following figures, we show some snippets of the XML file produced by the Flow Monitor, where we can find the necessary information for each flow in the network. As stated above, a flow is characterized as being a unique 5-tuple (source address, destination address, protocol, source port, destination port), identified by an integer value, the flow-id. For example, in Figure 15, we show the 5-tuple of $\text{flow} - \text{id}: 1$, where we can observe that the source node has $\text{IP} = 10.1.0.21$ and the destination node has $\text{IP} = 10.1.0.1$. More importantly, we can notice the destination $\text{port} = 1234$, which is the port that we predefine to connect the node application that generates and receives interesting data to analyze. Flow Monitor captures all network connections, and these make use of various ports. Therefore, by clearly identifying our application port, we can easily find the flows of interest. The source port is a random port number that is momentarily used to send UDP packets (i.e., data) as typical behaviour of the selected UDP protocol identified by the respective protocol number 17.

```
30543     <Flow flowId="1" sourceAddress="10.1.0.21" destinationAddress="10.1.0.1"
      protocol="17" sourcePort="49153" destinationPort="1234">
30544       <Dscp value="0x0" packets="173" />
30545     </Flow>
```

Figure 15: Flow Monitor 5-tuple flow

Without choosing a predefined and unique port for our customized application, it would be very cumbersome if not impossible to identify the flow-ids of interest among all the others in the network. To have an idea, when we use 120 nodes, there are up to 7534 flow-ids in our simulations, and we are only interested in 20. Moreover, besides our user-configured applications, others maintain the network such as the routing protocols, which create a high number of flows when transmitting signaling packets between the nodes.

In Figure 16, we present a snippet where we can find all the above-mentioned Flow Monitor statistics per flow.

```
1 <?xml version="1.0" ?>
2 <FlowMonitor>
3   <FlowStats>
4     <Flow flowId="1" timeFirstTxPacket="+3.86217e+09ns" timeFirstRxPacket="+2.07269e+11ns"
  timeLastTxPacket="+2.99675e+11ns" timeLastRxPacket="+2.39597e+11ns" delaySum="+8.58822e+09ns"
  jitterSum="+9.44058e+09ns" lastDelay="+482371ns" txBytes="42904" rxBytes="3224"
  txPackets="173" rxPackets="13" lostPackets="159" timesForwarded="2">
```

Figure 16: Flow Monitor Statistics per-flow

The most general way of collecting results is to enable Ascii or PCAP traces on NS3. It is possible to do so on the *YansWifiPhy* object, which allows tracing each packet sent from one node to others. This option is also enabled on our simulation code. However, it was only used for debugging purposes while configuring and inspecting the network behaviour. For example, in Figure 17, we show a node's PCAP trace file opened in WireShark packet inspector software.

We can see that two types of packets appear and both use a UDP connection, but one is specified as being AODV and the other simply UDP. AODV packets are automatically set by the AODV routing protocol, while the other UDP packet is sent by the BSM application (i.e., safety application) that we will use in our simulations. Many more other signalling and data packets are sent in the network, such as ARP requests. These traces provide the full scope inspection, from low-level to high-level.

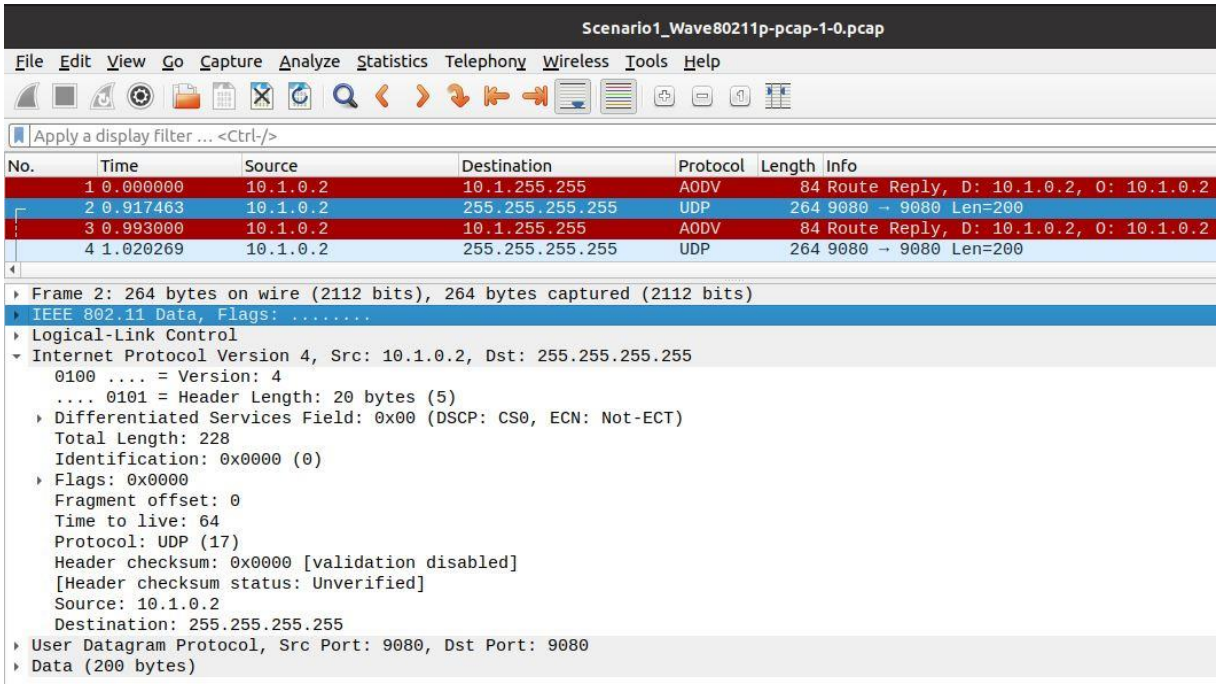

Figure 17: WireShark Packet inspection

Furthermore, it is crucial to note that NS3 adds the packet headers overhead. For example, in Figure 17, we can see the UDP BSM packet, which is broadcasted by the application and has a data size equal to 200 bytes (defined by us, according to the standards of BSM applications). Yet, the total length of the packer is 264 bytes, where the additional bytes are the additional headers such as UDP (8-byte), IPv4 (20-byte), etc.

Finally, we also leverage NS3 by inspecting the node's mobility log file and corresponding routing tables, which allow us to analyze network behaviour in dept. For example, in the next figure, we present a snippet of node-1's routing table trace, where it can be seen that it has three installed routes for nodes 2, 3, and 6. Notice that the IP: 127.0.0.1 is localhost (that is why it is always 1-hop distance) and serves only for internal routing purposes.

```
Node: 1, Time: +5.00s, Local time: +5.00s, Ipv4ListRouting table
  Priority: 100 Protocol: ns3::aodv::RoutingProtocol
Node: 1; Time: +5.00s, Local time: +5.00s, AODV Routing table

AODV Routing table
Destination     Gateway         Interface       Flag    Expire                  Hops
10.1.255.255    10.1.255.255    10.1.0.2        UP      +9223372036.85s         1
127.0.0.1       127.0.0.1       127.0.0.1       UP      +9223372036.85s         1

10.1.255.255    10.1.255.255    10.1.0.3        UP      +15223372036.85s        2
127.0.0.1       127.0.0.1       127.0.0.1       UP      +9223372036.85s         1

10.1.255.255    10.1.255.255    10.1.0.6        UP      +38223372036.85s        5
127.0.0.1       127.0.0.1       127.0.0.1       UP      +9223372036.85s         1
```

Figure 18: Node-1 AODV Routing Table

4.5 Code Implementation Details

In the previous subsection, we acknowledged the three main approaches to collect and manipulate static values, with which we can analyze our simulation scenarios. In this subsection, we take a closer look at NS3's tools and source code. It is essential to recognize what, where, and how we can make changes in NS3's source code and the additional modules in order to build our crafted scenarios. We want to simulate different IoV scenarios while changing crucial variables to assess their impact. These variables are the imposed conditions or scenario configurations that dictate the outcome of the results that we require. For this reason, in the following subsections, we explain how we modified the code to provide the desired outcomes, depending on the packet processing time, packet hop limit, and packet size in the network.

4.5.1 Packet Processing Time

We need to manipulate the packet processing time to mimic the time overhead added by the cryptosystem. In this way, we will simulate the conditions of having and not having the cryptographic processing time required for the decryption, encryption, and re-encryption functions. Consequently, we will be able to analyze the corresponding impact on the overall network.

At first, we were tempted to simply add some delay functions inside the node, which makes sense, but we must be careful not to put the simulator to sleep. Delay functions are mainly achieved by blocking/stopping the execution of the given process. Yet, we do not want to block the simulator, which is event-driven, and so the packet processing is not the only work the simulator has to do. We want to model our packet processing time, i.e., taking time to process some internal work. The way to do this is via the *Callbacks* system and the *Simulator::Shedule* core functions. We focus on UDP packets, given that we will make exclusive use of them, operating at L4 of the Open Systems Interconnection (OSI) model. When we call the *Schedule* method over a send UDP packet Event, we are inserting the packet on an internal queue that will send it at the specified time. Then, to introduce processing time, we merely have to identify the appropriate *Schedule* function responsible for scheduling the packets sending

process within the Simulator and modify it accordingly. We can find this function in a C++ file at the heart of NS3, in the “src/internet/model” directory, which is named *“udp-l4-protocol.cc”.*

The function that sends packets is termed *Send* (see Figure 19). Again, we can observe that the only required modification is to schedule the “*m_downTarget*” inner function to perform its work later; this function is responsible for passing L4 UDP packets to L3. Adding the required virtual processing time, named *“cryptoDelay,”* which was previously calculated as follows: 105µs, 545µs, and 569µs for encryption, decryption, and re-encryption functions, respectively. Furthermore, it is crucial to recognize this can be done by modifying other source files inside the same *Internet* module, such as *“ipv4-l3-protocol.cc”*, at layer 3 of the OSI model and containing *“IpForward,” “Send”* and *“Receive,”* functions to forward, send and receive packets, correspondingly.

```
void
UdpL4Protocol::Send (Ptr<Packet> packet,
                     Ipv4Address saddr, Ipv4Address daddr,
                     uint16_t sport, uint16_t dport, Ptr<Ipv4Route> route)
{
  NS_LOG_FUNCTION (this << packet << saddr << daddr << sport << dport << route);

  UdpHeader udpHeader;
  if(Node::ChecksumEnabled ())
    {
      udpHeader.EnableChecksums ();
      udpHeader.InitializeChecksum (saddr,
                                    daddr,
                                    PROT_NUMBER);
    }
  udpHeader.SetDestinationPort (dport);
  udpHeader.SetSourcePort (sport);

  packet->AddHeader (udpHeader);

  //m_downTarget (packet, saddr, daddr, PROT_NUMBER, route);

  /* Add processing delay in UDP-L4 core function responsible for sending them */
  Time cryptoDelay = NanoSeconds (569);
  Simulator::Schedule (cryptoDelay, &UdpL4Protocol::m_downTarget, this, packet, saddr, daddr, PROT_NUMBER, route);
}
```

Figure 19: “udp-l4-protocol.cc” - Send Packet function

NS3 uses sockets to connect the nodes and instantaneously transfer packets once they are put in the event queue via *Schedule* calls. In Figure 19, we can observe the detailed implementation of the *Send* function, where it makes packet checksum, adds appropriate headers, and passes it down to L3 after the configured processing time overhead. It is only at L3 that the actual socket is created for the node's connection. At this stage, the processing time has already added the required delay to impact the simulation.

4.5.2 Packet Size

We are interested in using two different packet sizes to simulate and study the impact of plaintext size expansion when converting into ciphertext. We consider the size of a 220-byte plain text being encrypted into a 1677-byte ciphertext. Finally, we highlight that NS3 adds the necessary packet headers. For example, the UDP packets have an extra of 28 bytes (20-byte for IP and 8-byte for UDP headers).

4.5.3 Packet Hop Limit

This variable is used to limit the maximum allowed packet hops in the network. To achieve it, we only have to configure the application that generates the packets. In our case, it will be the OnOff application, already implemented in the NS3 source code in the “src/applications/model” directory, in the file named: “*onoff-application.cc*”.

The configuration is done by modifying the UDP IPv4 TTL field, which limits the maximum hop count to the desired number. We are interested in simulating our scenarios with the three different quantities of one, ten, and unlimited, matching the conditions of single hop, multi-hop with the recommended number of hops, and unlimited multi-hop, respectively. Simulating these conditions allows us to analyze the impact of multi-hop features in IoV. Besides, we assume that single hop enables a node to re-encrypt a ciphertext only once mimicking the first PRE we presented in Figure 5. Consequently, we can match the TTL value to the maximum number of permitted re-encryptions.

In Figure 20, we can observe the code section related to the application packet send function. This function generates the entire packet, inserts the headers, tags, and the TTL field with the inner function *“tag.SetTtl”*, which receives one integer input, the TTL.

```
void OnOffApplication::SendPacket ()
{
  NS_LOG_FUNCTION (this);

  NS_ASSERT (m_sendEvent.IsExpired ());

  Ptr<Packet> packet;
  if (m_unsentPacket)
    {
      packet = m_unsentPacket;
    }
  else if (1) //m_enableSeqTsSizeHeader)        // always add full packet header
    {
      Address from, to;
      m_socket->GetSockName (from);
      m_socket->GetPeerName (to);
      SeqTsSizeHeader header;
      header.SetSeq (m_seq++);
      header.SetSize (m_pktSize);
      NS_ABORT_IF (m_pktSize < header.GetSerializedSize ());
      packet = Create<Packet> (m_pktSize - header.GetSerializedSize ());
      // Trace before adding header, for consistency with PacketSink
      m_txTraceWithSeqTsSize (packet, from, to, header);
      packet->AddHeader (header);

      SocketIpTtlTag tag;
      tag.SetTtl (10);           // here we define the desired TTL, i.e., the Hop limit
      packet->AddPacketTag (tag);
    }
```

Figure 20:" onoff-application.cc" - Send Packet func

5. Performance Evaluation

This chapter focuses on evaluating the secure dissemination of messages and studies the combined use of fixed and mobile nodes to establish an IoV to distribute general and specific interest data. Later, we analyze the outcomes and provide plausible explanations.

5.1 Simulation Models

We consider the network topology in Figure 1, where vehicles are mobile nodes that travel around the city, and the deployed RSUs are fixed nodes strategically placed to support communications. The RSUs are inter-connected through a wired network that provides relatively high bandwidth and fast access to all necessary data contents to disseminate in our simulations. For this reason, it is assumed that RSUs have all the data contents that vehicles may need. Moreover, the RSUs have elevated antennas with more transmission power to sustain robust connections with broader coverage areas than vehicles.

The dissemination is supported by the well-known AODV routing protocol, which builds the necessary routing tables to disseminate data. We can inspect these routing tables per node, containing the IP addresses of next-hop node interfaces and the number of hops required to reach the destination.

Nodes generate messages via software applications, known as apps. Apps are configured to sync and pair receiver/transmitter NetDevices, creating a wireless channel used by the nodes to send/receive messages. According to the app specifications, these messages have different requirements such as size (in bytes or bits) and the data bitrate or bandwidth (in Mbps or Hz) necessary to accomplish the IoV network. All these messages are transmitted wirelessly and can be easily intercepted by any node. Therefore, we must employ cryptographic systems that come with overheads (i.e., processing delays, size expansion, etc.) and limitations imposed by the cryptosystems (i.e., number of re-encryptions, message size, etc.).

To analyze the computational overhead cost of our proposed NTRUReEncrypt scheme, we consider the previously measured total average times for encryption, re-encryption, and decryption operations. Finally, we should use these measurements and test with several IoV scenarios involving different numbers of devices, applications, communication technologies (i.e., 802.11p or C-V2X), and protocols to evaluate our scheme.

5.1.1 The IoV Scenario

In scenario 1, we run an IoV scenario considering only V2V communication and the parameters specified in Table 9. The simulation has a duration of 300 seconds per run each with a different seed responsible for generating the random numbers used by diverse models and functions of the simulator. It most involves 120 nodes (i.e., mobile vehicles) moving according to the Random Way Point (RWP) mobility model with speeds randomly varying between 0-25 m/s and no pause time. Random values also generate the initial node positions in vector format in a tri-dimensional space (x,y,z). The total area matches a 1000x1000 m, thus coordinates x and y have values in the range [1-1000] while the z

coordinate has values in the range [1.0-2.0] which defines the altitude of the netDevice, i.e., the antenna, required to simulate the two-ray-ground loss propagation model. All nodes are equipped with the WAVE netDevice, using the 802.11p standard with continuous access to a 10 MHz Control Channel for all traffic. In addition, each node may have installed one or two applications, namely App1 and App2. The wireless channel conditions and all configurations related to the physical layer are selected based on the WAVE module examples (see Figure 21).

The applications generate different traffic patterns to mimic a simple vehicular network. The traffic consists of sending UDP packets between nodes specified as follows:

- **App1**: Installed in all nodes, broadcasts 200 bytes BSM safety messages at $0.1\ Hz$ (or ten times per second) at a constant speed of 6 Mbps. This application introduces typical traffic inside IoVs using the WAVE architecture, considering the standard 145 m distance.

- **App2**: Installed in 20 predefined source nodes, we consider this to be group A. The source nodes attempt to route 220 bytes of plaintext messages at an application rate of 2 Mbps to one specific sink node. The sink nodes are contained in group B and match the source nodes one-to-one. Consequently, there are 20 different sink nodes. All nodes cooperate in the routing/forwarding process and use AODV to build the routing paths enhance enabling multi-hop. This application switches between two subsequent states, the ON and OFF states. The duration of each state is a random value in the range of 0 to 2 seconds. During the OFF state, the application is turned off, while during the ON state, the application sends unicast messages, i.e., the UDP packets. Simulates specific source-destination communications relying on proxies to build the AODV paths and forward messages accordingly.

This scenario considers the importance of encrypting all transmitted data, independently of its utilization or purposes. For this reason, we will compare several cases differentiated by the imposition of cryptography overheads in terms of additional delays (crypto functions processing time) and extra data size as specified above in the NTRU cryptosystem section, according to the expansion factor.

Additionally, we are interested in analyzing the IoV performance according to different numbers of maximum allowed re-encryptions. Specifically, we will consider three cases. In this way, we will understand how important it is to enable multi-hop features and build cryptosystems capable of re-encrypting ciphertexts several times. The three cases will limit the number of re-encryptions of 1, 10, and 200 occurrences.

In Figure 22, we present an example of scenario 1. For simplicity, the figure only contains three source-sink pairs and nine proxy nodes (a total of 15 vehicles). We will exclusively consider App2 traffic in the static analysis because it reflects the unicast connections we want to evaluate in terms of performance with and without the cryptosystem overheads. In addition, Figure 22 also provides us with the visual logic about how the source-sink pairs are built. In the example, we consider a total of 100 nodes created inside an indexed array from 1 to 100. Afterward, we select the first 20 nodes as source nodes (Group A) and the subsequent 20 as sink nodes (Group B). Finally, all the remaining nodes are proxies, crucial to understanding the multi-hop cooperation impact and overhead costs.

Table 9: Network Parameters for NS3 simulation scenario 1.

Parameter		Value
Map Area		1000x1000 m
Number of Runs with different Seeds		3
Mobility Model		Random Way Point
Number of Vehicles		40, 60, 80, 100, 120
Propagation Loss Model		Two ray ground
Transmit Power		7.5 dBm
Node Speed		0 - 25 m/s
Routing Protocol		AODV
Transport Protocol		IPv4/ UDP
Number of Re-encryptions Limit		1, 10, 200
App1	Data Size	200 B
	Transmission range	145 m
	Transmission Rate	6 Mbps
App2	Data Size	220 B, 1677 B
	Transmission range	500 m
	Transmission Rate	2 Mbps
Simulation Time		300 s

```
// App1 data rate
static ns3::GlobalValue g_rate ("S1WAVErate",
                                "Data rate",
                                ns3::StringValue ("2048bps"),
                                ns3::MakeStringChecker ());

// Physical mode used by the WAVE PHY module
static ns3::GlobalValue g_phyModeB ("S1WAVEphyModeB",
                                    "PHY mode (802.11a)",
                                    ns3::StringValue ("DsssRate11Mbps"),
                                    ns3::MakeStringChecker ());
// WAVE physical layer mode
static ns3::GlobalValue g_80211mode ("S1WAVE80211mode",
                                     "802.11 mode (0=802.11a;1=802.11p)",
                                      ns3::UintegerValue (1),
                                      ns3::MakeUintegerChecker<uint32_t> ());
// WAVE physical layer data rate
static ns3::GlobalValue g_phyMode ("S1WAVEphyMode",
                                   "PHY mode (802.11p)",
                                   ns3::StringValue ("OfdmRate6MbpsBW10MHz"),
                                   ns3::MakeStringChecker ());
// Loss model
static ns3::GlobalValue g_lossModel ("S1WAVElossModel",
                                     "Propagation Loss Model",
                                     ns3::UintegerValue (3), //2-Ray-Ground
                                     ns3::MakeUintegerChecker<uint32_t> ());
// Fading model
static ns3::GlobalValue g_fading ("S1WAVEfading",
                                  "Fast Fading Model",
                                  ns3::UintegerValue (0),
                                  ns3::MakeUintegerChecker<uint32_t> ());
// Transmit Power
static ns3::GlobalValue g_txp ("S1WAVEtxp",
                               "Transmission power dBm",
                               ns3::DoubleValue (7.5),
                               ns3::MakeDoubleChecker<double> ());
// GPS accuracy
static ns3::GlobalValue g_gpsAccuracyNs ("S1WAVEgpsAccuracyNs",
                                         "GPS sync accuracy (ns)",
                                         ns3::DoubleValue (40),
                                         ns3::MakeDoubleChecker<double> ());
```

Figure 21: WAVE Default Configurations

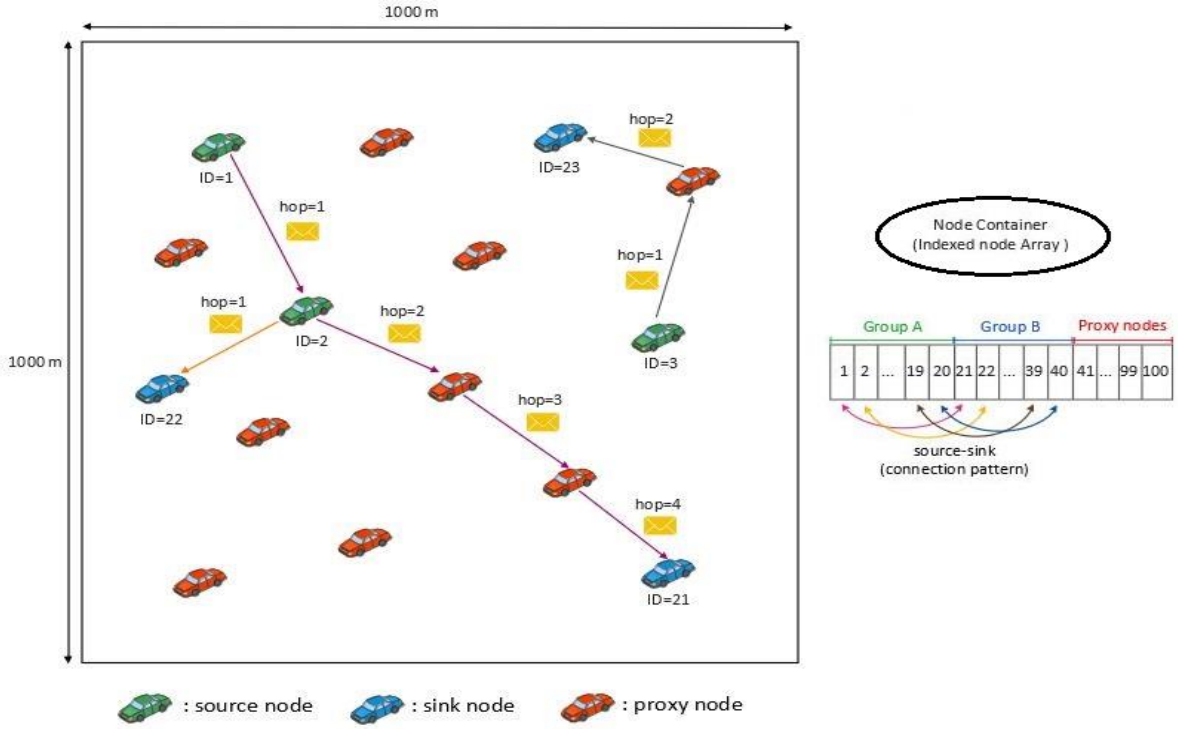

Figure 22: Scenario 1 RWP Map

5.2 Results and Discussion

In this subsection, we present simulation results and discussion. We present these statistics by bar graphs calculated as the average value of three runs, each with a different seed value. We highlight the importance of using different seeds for the same simulations. These are used to generate all random values in the simulation, and the scenario heavily depends on NS3's functions to produce random values.

The represented outputs are the overall IoV average values, considering only the 20 source-sink flows described in the previous subsection. In this way, we believe that the statistical results are more consistent and easier to understand. However, if we take a particular node's example, we may be misled because the introduced inputs to change the environment may never be felt by that particular node.

Our simulations provide the results printed in an XML file for practical data parsing. Besides, each simulation outputs the general, or total, results on the console, as shown in figure 9. The simulation inputs are selected to deliver outputs that allow us to make insightful conclusions about the impact of the NTRUReEncrypt cryptosystem and the use of multi-hop features in IoV with different node densities.

The three considered cases have a different number of maximum re-encryptions, as follows:

- **200 Re-Encryptions (case-1):** To simulate the IoV with unlimited re-encryptions, since packets will never require more than 200 hops to reach their destination in practice.

- **10 Re-Encryptions (case-2):** To simulate the IoV with the optimal number of maximum hops and re-encryptions that a ciphertext can afford. After passing this limit, the package is dropped.
- **1 Re-Encryption (case-3):** To simulate the IoV with only one possible re-encryption, i.e., transmitting a packet with a maximum of two hops before reaching the destination. Such as the example provided in Figure 5.

For each case, we simulate the IoV with a different number of vehicle nodes, increasing the number of proxies that cooperate in forwarding packets to their destinations. Also, we obtain results for three distinct situations regarding the NTRU ReEncrypt cryptosystem overheads, as follows:

- **PRE-0:** Does not consider any crypto function's overhead, i.e., no cryptosystem is implemented, and the packet data size is 220 bytes.
- **PRE-1:** Considers the processing time the crypto functions require to re-encrypt the ciphertexts as they transverse the IoV to reach their destination (per-hop). Does not consider any overhead caused by expanding the plaintext into ciphertext, i.e., the packet data size is 220 bytes. Our motivation is to analyze if the cryptographic time overhead alone will impact the overall network.
- **PRE-2:** Considers crypto function's processing time and packet data size expansion by the expansion factor previously studied to transform plaintext into ciphertext, i.e., the packet data size is 1677 bytes.

Our goal is to measure the network performance and compare the different cases to make conclusions about the impacts caused by extra overheads and the different number of nodes with varying limitations of multi-hop.

Below, we present the results and provide feasible explanations. In Figure 23, we show an example of the console printed results for the PRE-2 simulation with the following inputs:

- Run number: 7, seed number: 3, case: 2.

```
---------------------------------------
All Tx Packets : 3446
All Rx Packets : 1164
All Delay Per Packet : 0.106266
All Total Delay : 366.193
All Lost Packets : 2282
All Drop Packets : 135
Packets Delivery Ratio : 33%
Packets Lost Ratio : 66%
All number of hops : 4232
All Throughput: 33.3335
```

Figure 23: Overall IoV simulation results.

- **Packet Delivery Ratio (PDR)**

In the following figures, we can observe and compare the PDR for each case. The PDR measures the ratio of the number of packets delivered in total to the total number of packets sent from the source node to the destination node. It immediately stands out that for cases 1 and 2, the PDR increases with the number of nodes. Also, these two cases have very similar results, while case-3 demonstrates an opposite behaviour, decreasing the PDR while increasing the number of nodes. To better understand and describe such effects, we need to consider other metrics, such as the average number of hops or the end-to-end delays, which we will see later.

We can conclude that adding more proxy nodes in the network improves the PDR since more proxy nodes are willing to cooperate and forward the packets to the destination. The PDR for case-1 PRE-0 is in the range of [15; 37] %, while for PRE-1 is [14; 36] %. The PDR for case-2 PRE-0 is in the range of [15; 34] %, while PRE-1 remains in [14; 36] %. Moreover, when comparing PRE-0 with PRE-1, we perceive that the added processing time makes little difference with at most a 3% variance.

On the other hand, case-3 has slightly degraded PDR when we add more proxy nodes. Such can be explained by the fact that adding more nodes adds more congestion to the network, which degrades the packet deliveries. Moreover, in case-3, the nodes can only do at most one re-encryption, two hops, which covers concise areas, so the probability of reaching the destination is very low. Thus, at its best, we can observe that it only accomplishes a PDR ranging from 3% to 8%.

In this way, we can acknowledge the importance of having a cryptosystem capable of performing several re-encryptions. Otherwise, we cannot leverage the node's willingness to cooperate.

Finally, we see that PRE-2 has the same rising behaviour described for PRE-0 and PRE-1 but with improved proficiency. We expected that a bigger packet could negatively impact the PDR because it takes more time to be transmitted and, therefore, would be more prone to get errors due to more exposure to interference. Therefore, we will investigate the other statistics to explain this performance.

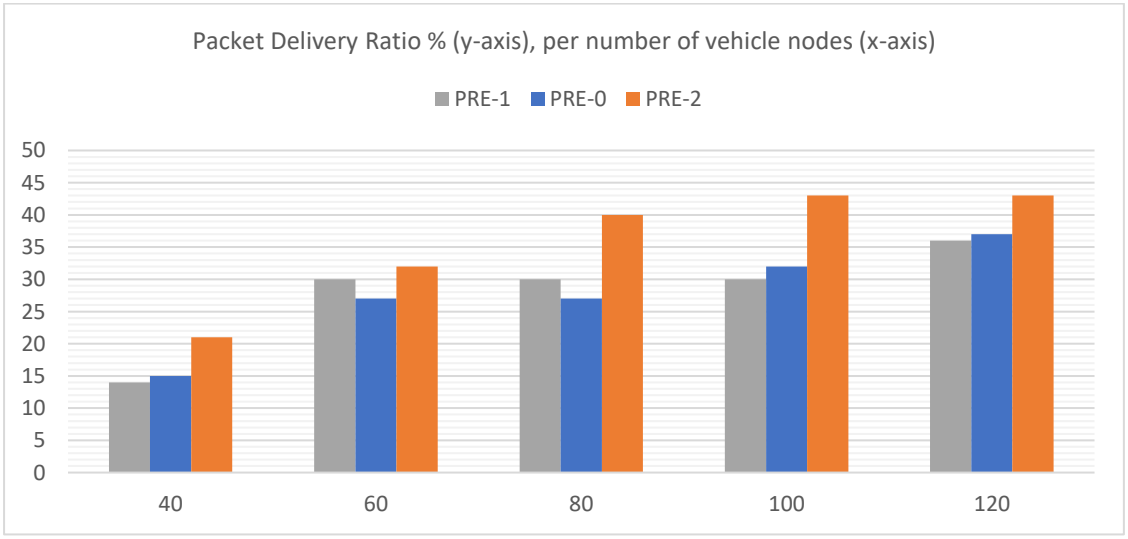

Figure 24: Packet Delivery Ratio: 200 ReEnc

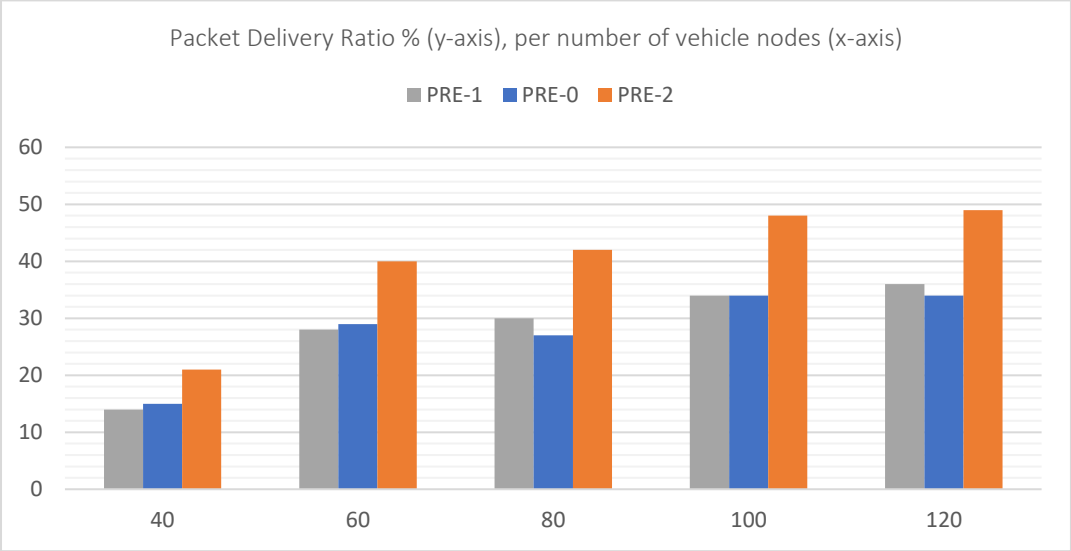

Figure 25: Packet Delivery Ratio: 10 ReEnc

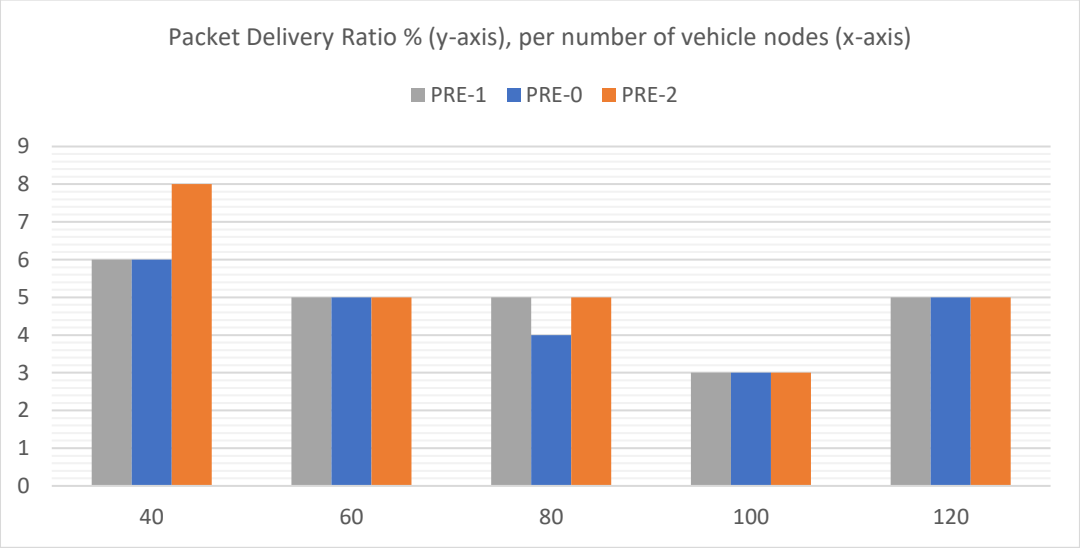

Figure 26: Packet Delivery Ratio: 1 ReEnc

- **Average Number of Hops**

Here we analyze the average number of hops that packets take from the source node to reach their respective destination node. Moreover, this metric provides us with an excellent guideline to explain other simulation results. Notice that the end-to-end delay of the packets is closely related to the average number of hops, where more hops imply a bigger end-to-end delay, as we will later observe. Also, it is important to acknowledge that each hop has a certain associated cost that changes according to the previously described situations: PRE-0, PRE-1, and PRE-2.

In the following graphs, we can perceive that for cases 1 and 2, the average number of hops increases with the number of nodes. Such is expected since there exist more nodes willing to participate in the packet forwarding process. Moreover, in both cases (1 and 2), the outputs are very similar, with

values in the range [3, 6] hops. Therefore, we can conclude that the maximum recommended number of hops in the IoV should not surpass ten hops, corroborating our cryptosystem configuration setup.

As a consequence of having a similar number of hops, we can observe that all outputs will also be identical. We can conclude that the number of hops a packet requires to reach the destination highly impacts the IoV since it models its performance. Enabling multi-hop is crucial to achieving good performances, and this should be limited to ten hops.

Because cases 1 and 2 have similar outputs, from now on, we will only present case-2 results, with the recommended maximum number of re-encryptions and hops limit. Yet, we emphasize that case-2 results were slightly better according to our defined metrics.

On the other hand, for case-3, we can verify that the average number of hops has low values, always below 1.30, which are expected since there is a 2-hop limit or one re-encryption. Thus, we can conclude that source vehicles hardly ever connect to the destination vehicles when leveraged solely by one proxy or direct connections. Furthermore, the value of 1.30 hops tells us that most of the delivered packets only used one-hop even with two hops possibility, which indicates that two hops cover a very short distance to route packets in our scenario.

Later we will see that the low average number of hops results in low packet end-to-end delays, as expected since the end-to-end delay measures the sum of all connection's delays. Consequently, the direct links (one hop/ one encryption) or one proxy connection (two hops/ one re-encryption) will have the lowest packet delays.

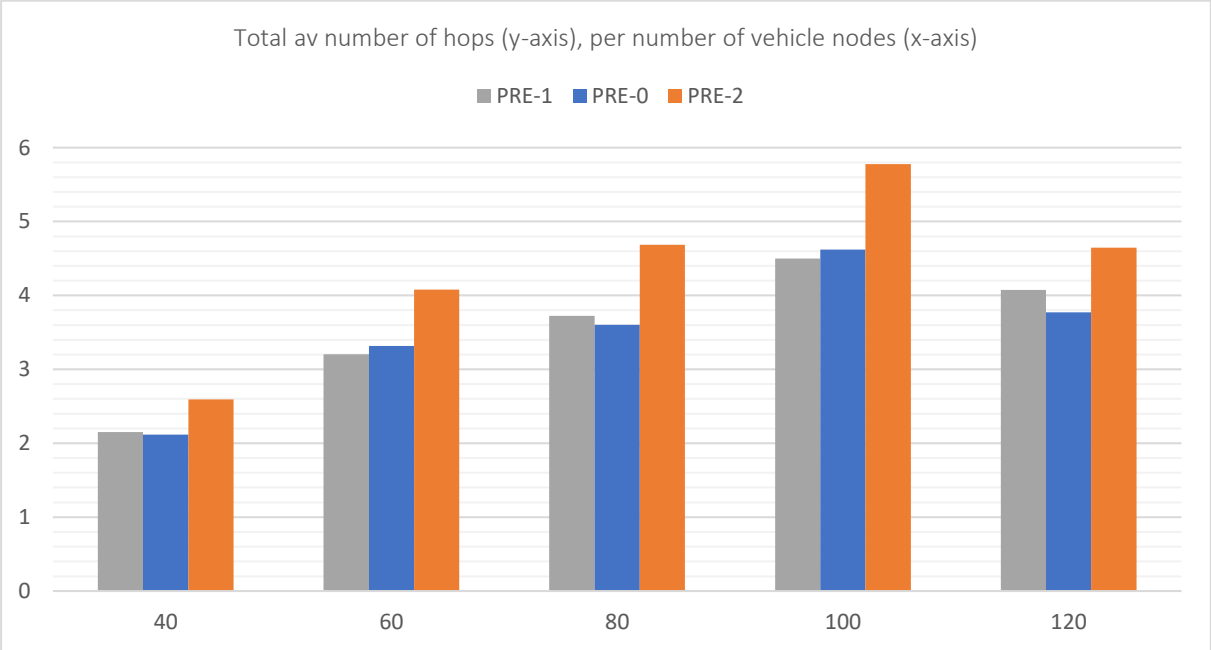

Figure 27: Av. Number of Hops: 200 ReEnc

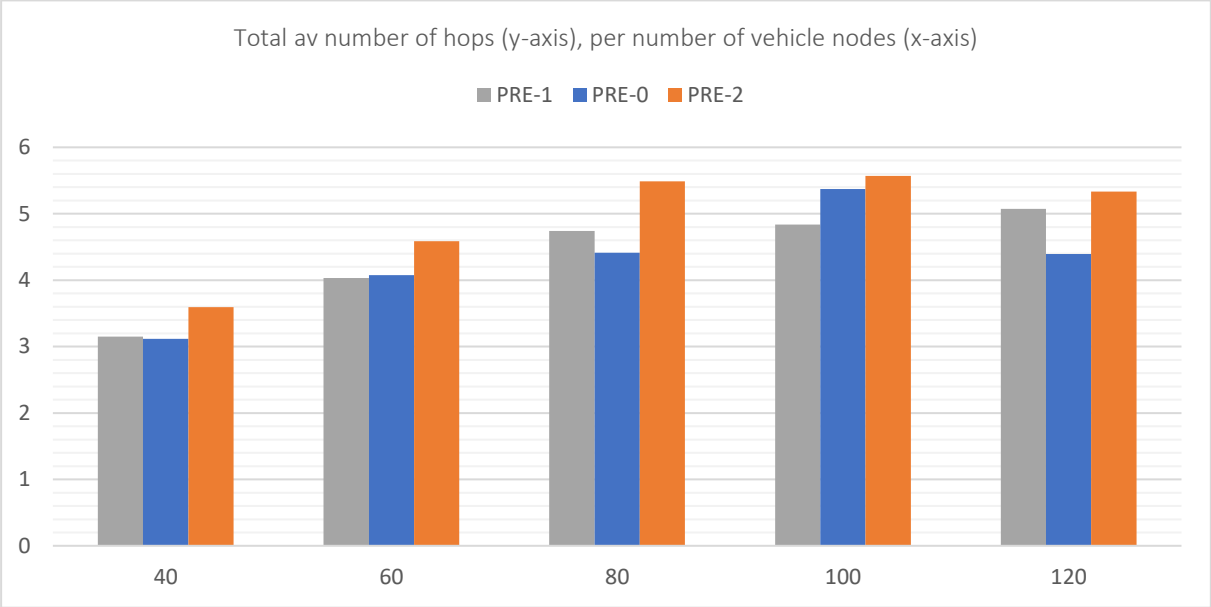

Figure 28: Av. Number of Hops: 10 ReEnc

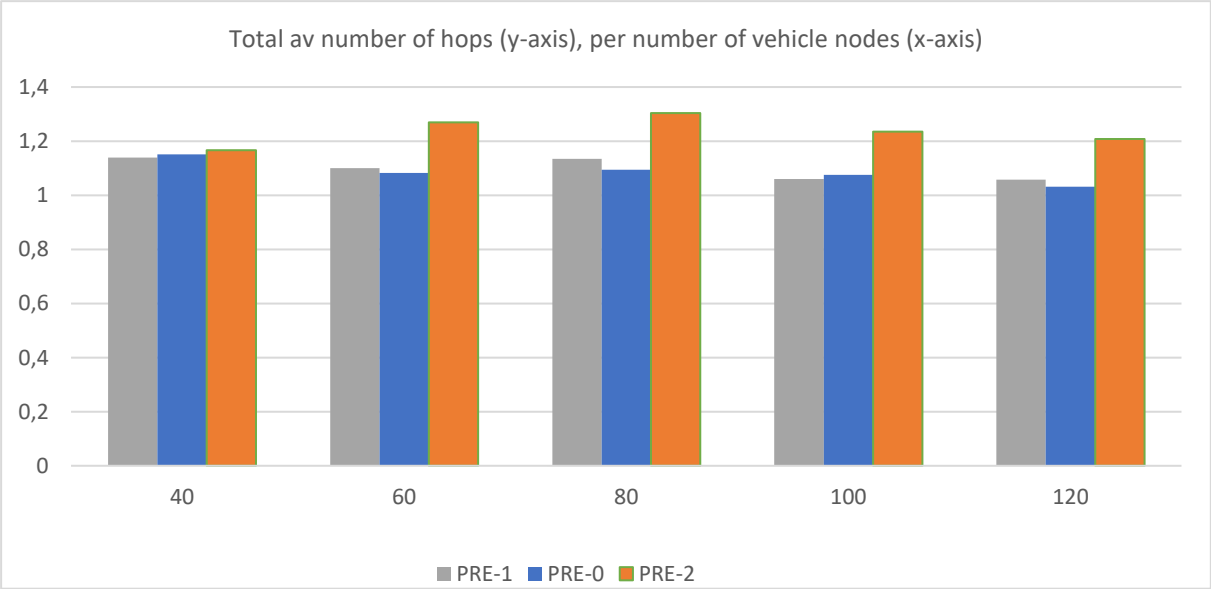

Figure 29: Av. Number of Hops: 1 ReEnc

- **Total Number of Transmitted Packets**

The total number of transmitted packets is always the same for each PRE case, which is expected since App1 continuously transmits at the same data rate and for the same duration in all the corresponding simulations.

Furthermore, we can observe that PRE-1 and PRE-2 cases send precisely the same number of packets, i.e., 3443 packets. Thus, allied to the similar PDR outcome, it indicates that our PRE cryptosystem processing time overhead does not impact the number of transmitted packets nor the received ones.

For case 3, we see that the number of sent packets is highly reduced, i.e., 440 packets. Such can be explained by the fact that the packets' size considered in case 3 increases by a factor of 6.72 concerning cases 1 and 2. However, we can notice that the number of sent bytes is similar in all cases:

- PRE-0 and PRE-1: $3443\ (packets) * 220\ (bytes) = 757\ 460\ bytes$
- PRE-2: $440\ (packets) * 1677\ (bytes) = 737\ 880\ bytes$

The difference between PRE-0 or PRE-1 and PRE-2 is $19.580\ KB$. This difference can be explained by the fact that more packets come with extra overhead regarding the headers each packet requires (with fixed size), in addition to the actual consumed data.

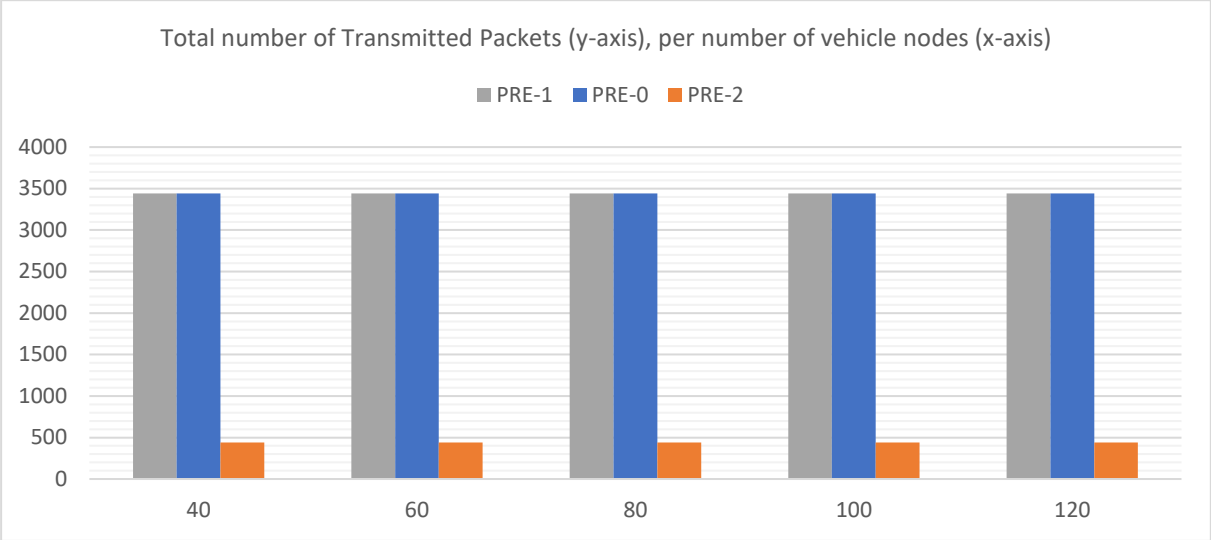

Figure 30: Number of Tx Packets

- **Total Number of Received Packets**

The total number of received packets sums the total number of packets received by all 20 flows or source-sink connections that we are analyzing. Accordingly, we can immediately observe that for case-2, the number of received packets increases with the number of added nodes (i.e., vehicles or proxies). Precisely, the number of received packets almost doubles when we go from 40 to 60 nodes. Subsequently, we keep adding 20 nodes, the number of received packets keeps rising, yet with decreased ratio. It could be indicating a possible saturation point from which adding more nodes would not impact the IoV in terms of required paths to forward packets. It is expected that a dense IoV has multiple ways to reach the same destination, and adding more nodes will not make a big impact on building new paths. Still, if we consider the importance of having backup routes, it is necessary to have dense networks.

In case-3, we observe that the number of received packets remains relatively constant, even decreasing with the number of nodes, as expected considering the PDR metric and equally explained.

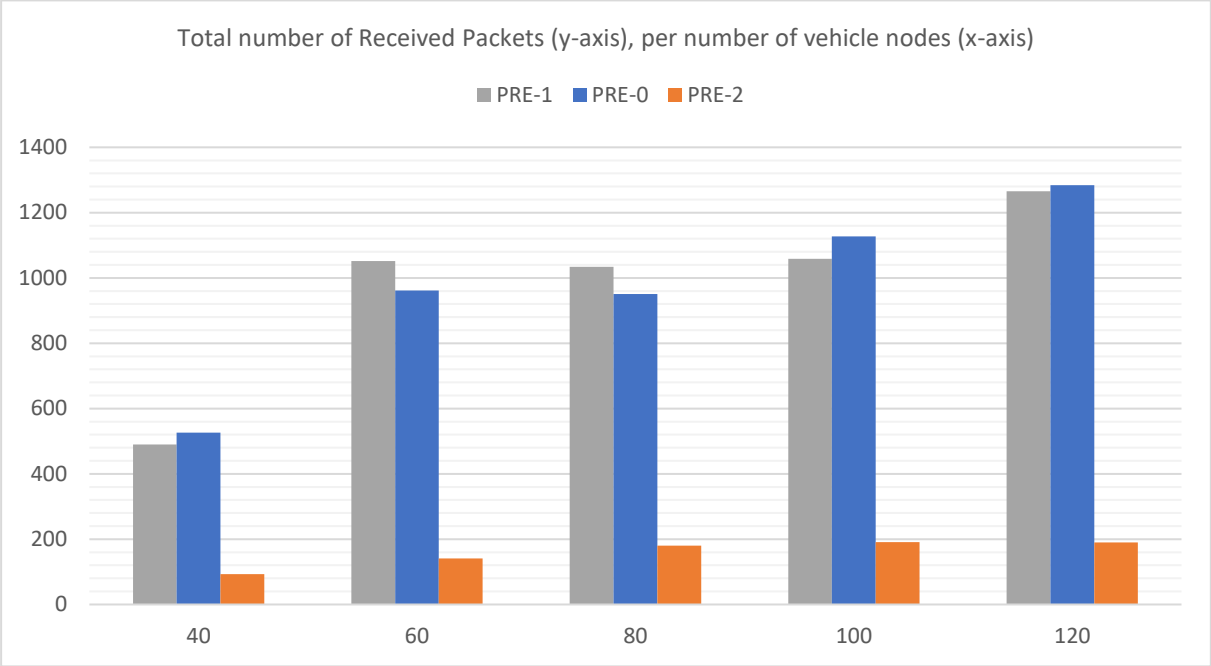

Figure 31: Total Number Rcv Packets: 10 ReEnc

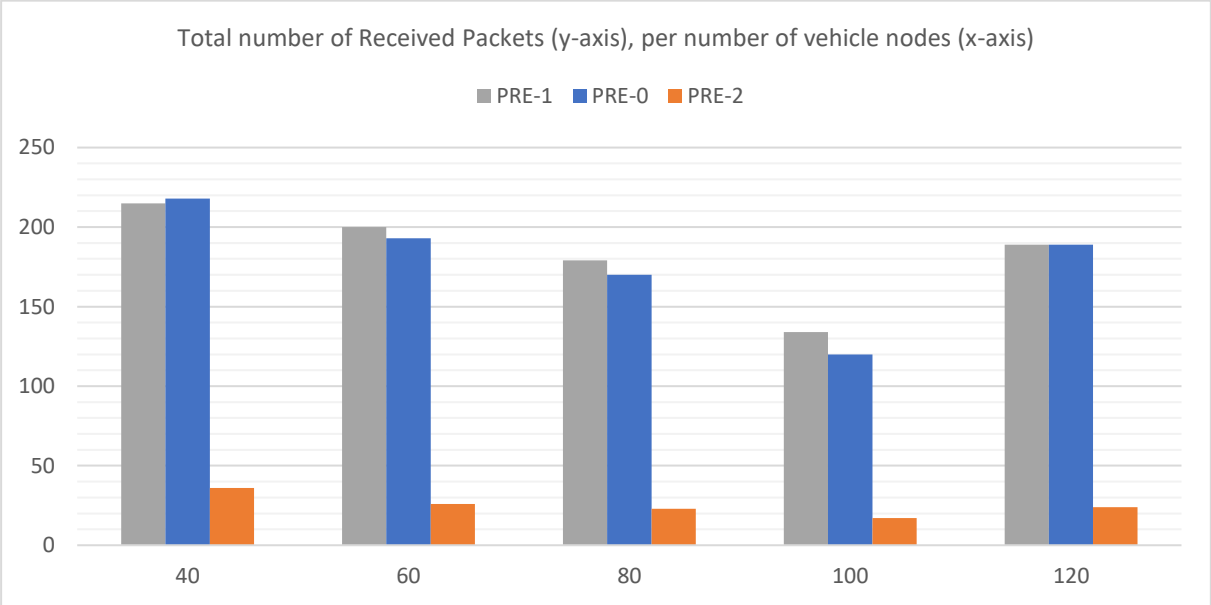

Figure 32: Total Number Rcv Packets: 1 ReEnc

Furthermore, by comparing PRE-0 with PRE-1, we again recognize the low impact that the NTRU-PRE implementation has. We find it logical because the end-to-end packet delay is in the order of magnitude of milliseconds. Therefore, the processing time for the cryptographic functions is three orders of magnitude lower in the nanoseconds, resulting in a low impact on the overall delay, as shown in the subsequent figures.

- **Per Packet Delay**

The packet delay measures the total delays since a packet is sent from the source node until it reaches the sink node, also referred to as the end-to-end packet delay.

In case-2, we can witness that the packet delay slightly increases with the number of additional nodes in the IoV. This phenomenon can be explained by the previously analyzed fact that the number of hops, or re-encryptions, also increases, implying higher delay values. Additionally, we correspondingly highlight the similar performance when comparing with and without PRE time overheads, PRE-0 and PRE-1.

However, in both cases, there is a big difference when comparing PRE-1 with PRE-2. This is due to the packet size difference, and we associate it with how NS3 calculates the delay to transmit packets. The total packet size, the available channel bandwidth, and the transmission loss model will impact the delay values. We conclude that the packet size severely affects the end-to-end packet delay. Yet, we see that the high end-to-end delay does not degrade the packet delivery when considering the PDR. This result is the most unexpected because we could foresee that an increased end-to-end delay would impair the packet delivery in a highly dynamic network since this would increase the probability of breaking a connection before sending the whole packet.

In case-3, the end-to-end delay decreases with an increase of node density. We can explain this performance by recognizing that adding more nodes we are statistically putting vehicles closer to each other. Therefore, when multi-hop is not enabled, we can observe more significant impact and conclude that packets will travel shorter distances between source and sink, which implies reduced end-to-end packet delay. Please note that a reduced end-to-end delay does not necessarily mean a higher PDR, as we can observe in our results.

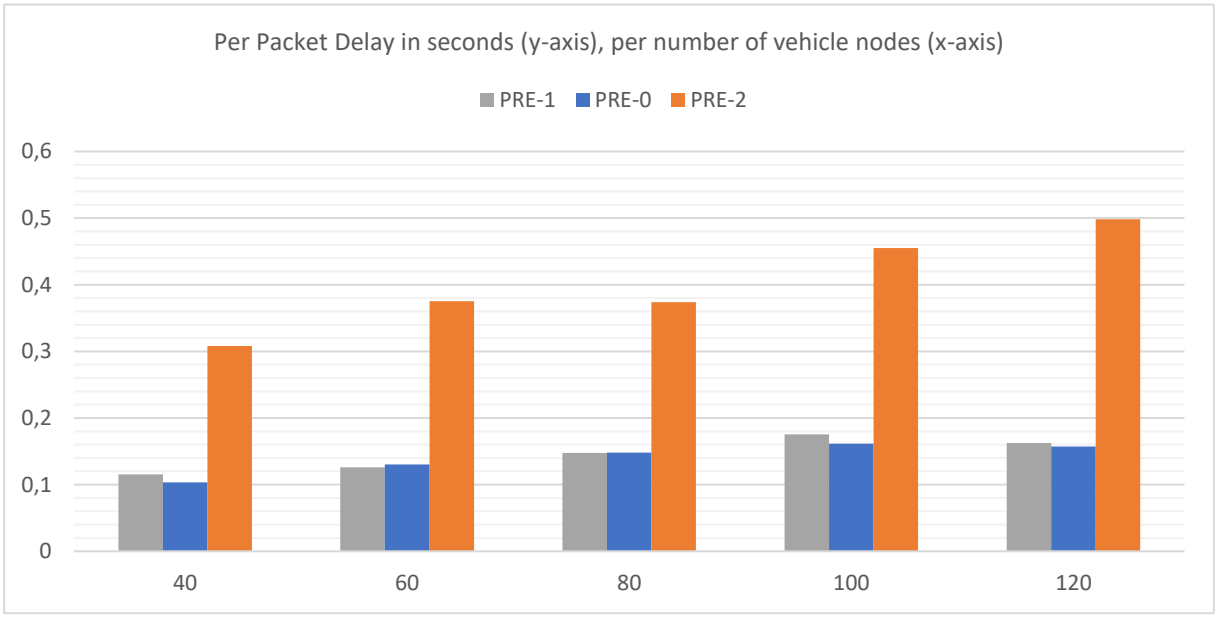

Figure 33: Packet End-to-End Delay: 10 ReEnc

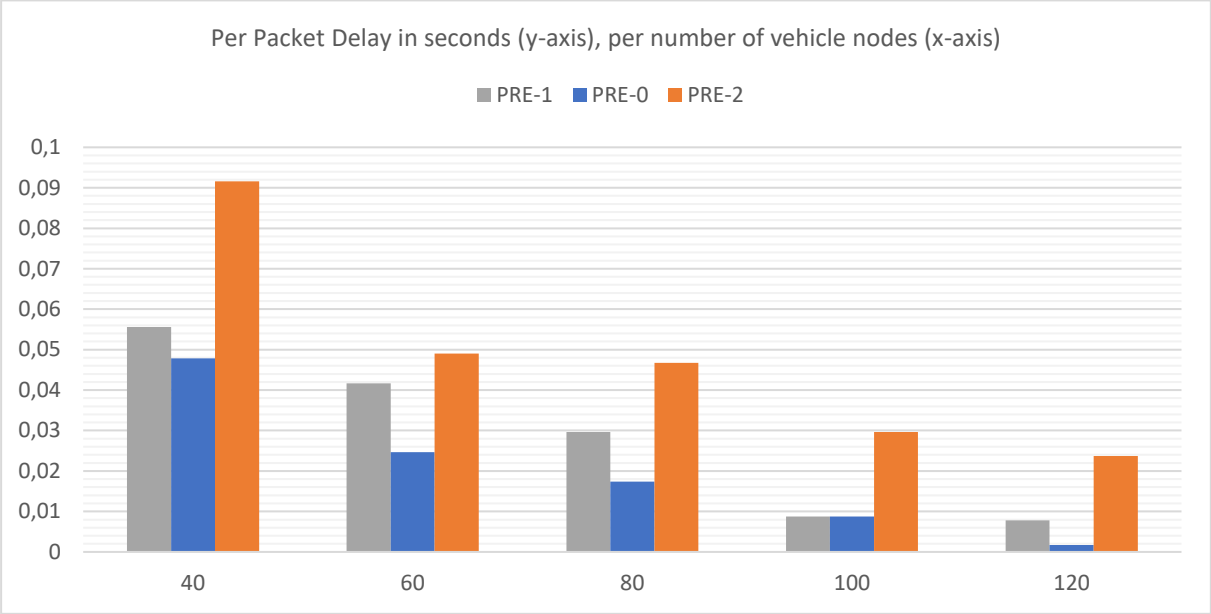

Figure 34: Packet End-to-End Delay: 1 ReEnc

6. Conclusions

As we stated along this work, OppNets are an alternative to consider where the Internet infrastructure is unavailable or saturated and should be integrated into the IoV concept. Due to the nature of these cooperative wireless networks, cryptography must be deployed to ensure data security.

The main goal of this thesis was to make a depth evaluation of the main factors that impact the message dissemination performance when adding the overhead imposed by the cryptosystem. Based on these results, we can conclude and foresee its practical usage in real-case scenarios.

6.1 Achievements and Contributions

We have implemented reasonably basic implementation of an IoV scenario, considering mobile vehicles. The scenario is implemented in the NS3 simulator and focuses on message dissemination using the WAVE device technology (based on the 802.11p standard). It is straightforward that there are undesired overheads when implementing cryptography. Nevertheless, we can observe that the overall network is mildly impacted, which indicates a positive perspective for future integration. Also, given the signs of progress in lattice-based cryptography and wireless communications, we can expect that the negative impact of cryptosystems will be reduced.

Moreover, this work also provides an extensive study about the PRE cryptosystems development. In this way, we are leaving our contribution to all researchers within the scientific community. We hope to have captured the interest of researchers and the general audience towards learning about the IoV, its importance, and the challenges ahead.

We have also shown the importance of cryptographic systems in securing the IoV. Precisely we have selected the NTRU PRE cryptosystem and explained its advantages, such as being faster than other typical cryptosystems or being lattice-based and therefore resistant to quantum attacks.

Finally, we leave our contribution related to the NS3 network simulator. Our scenario implementation is explained in detail and pretends to let the reader explore and build new scenarios.

6.2 Future Work

This work provides an extensive theoretical explanation about the IoV and a diversity of possible scenarios. However, it lacks the practical implantation and analysis of more scenarios, for example, considering RSU nodes.

Some approaches of future work are pointed out:

- A deep study of other PRE cryptosystems or hybrid implementations aiming for improved performance while maintaining high security.
- Build and evaluate new IoV scenarios with NS3. For example, considering RSU and vehicle nodes.
- Integrating different communication technologies in the scenarios, specifically the NS3 mmWave module, to compare it with the WAVE module and learn about its practical deployment pros and cons.

References

[1] J. Dede, A. Förster, E. Hernández-Orallo, J. Herrera-Tapia, K. Kuladinithi, V. Kuppusamy, P. Manzoni, A. bin Muslim, A. Udugama and Z. Vatandas, "Simulating Opportunistic Networks: Survey and Future Directions," *IEEE Communications Surveys & Tutorials,* vol. 20, no. 2, pp. 1547-1573, 2018.

[2] N. David, A. Isaac and L. Javier, "NTRUReEncrypt: An efficient proxy re-encryption scheme based on NTRU," *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS'15),* p. 179–189, 2015.

[3] L. D'Orazio, F. Visintainer and M. Darin, "Sensor networks on the car: State of the art and future challenges," *2011 Design, Automation & Test in Europe,* pp. 1-6, 2011.

[4] C. R. Storck and F. L. Duarte-Figueiredo, "A Survey of 5G Technology Evolution, Standards, and Infrastructure Associated With Vehicle-to-Everything Communications by Internet of Vehicles," *in IEEE Access,* vol. 8, pp. 117593-117614, 2020.

[5] M. Lee and T. Atkison, "VANET applications: Past, present, and future," *Vehicular Communications,* vol. 28, no. 2214-2096, p. 100310, 2021.

[6] O. Kaiwartya, A. H. Abdullah, Y. Cao, A. Altameem, M. Prasad, . C.-T. Lin and X. Liu, "Internet of Vehicles: Motivation, Layered Architecture, Network Model, Challenges, and Future Aspects," *IEEE Access,* vol. 4, pp. 5356-5373, 2016.

[7] R. Gasmi and M. Aliouat, "Vehicular Ad Hoc NETworks versus Internet of Vehicles - A Comparative View," *2019 International Conference on Networking and Advanced Systems (ICNAS),* pp. 1-6, 2019.

[8] S. Ashfaq Hussain, K. Mohammad Yusof, S. Mazhar Hussain and A. Vikram Singh, "A Review of Quality of Service Issues in Internet of Vehicles (IoV)," *2019 Amity International Conference on Artificial Intelligence (AICAI),* pp. 380-383, 2019.

[9] Y. Sun, L. Wu, S. Wu, S. Li, T. Zhang, L. Zhang, J. Xu, Y. Xiong and X. Cui, "Attacks and countermeasures in the internet of vehicles," *annals of telecommunications,* vol. 72, p. 283–295, 2017.

[10] L. Silva, N. Magaia, S. Breno, A. Kobusinska, A. Casimiro, C. X. Mavromoustakis, G. Mastorakis and V. Hugo C. de Albuquerque, "Computing Paradigms in Emerging Vehicular Environments: A Review," *IEEE/CAA Journal of Automatica Sinica,* vol. 8, no. 3, pp. 491-511, March 2021.

[11] M. Kangas, M. Hippi, J. Ruotsalainen, S. Näsman, R. Ruuhela, A. Venäläinen and M. Heikinheimo, "The FMI road weather model," *Proceedings of 16th International Road Weather Conference, SIRWEC 2012,* p. 117–23, 23-25 May 2012.

[12] T. Sukuvaara and C. Pomalaza-Ráez, "Vehicular Networking Pilot System for Vehicle-to-Infrastructure and Vehicle-to-Vehicle Communications," *International Journal of Communication Networks and Information Security (IJCNIS),* vol. 1, no. 3, pp. 1-11, December 2009.

[13] T. Sukuvaara, R. Ylitalo and M. Katz, “IEEE 802.11p Based Vehicular Networking,” *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS/SUPPLEMENT,* vol. 31, no. 9, pp. 409-417, September 2013.

[14] Y. L. Morgan, “Managing DSRC and WAVE Standards Operations in a V2V Scenario,” *Hindawi Publishing Corporation,* 2010.

[15] Douglas Aguiar do Nascimento, Yuzo Iano, Hermes José Loschi, Navid Razmjooy, Robert Sroufe, Vlademirde Jesus Silva Oliveira, Diego Arturo Pajuelo Castro and Matheus Montagner, “Sustainable Adoption of Connected Vehicles in theBrazilian Landscape: Policies, TechnicalSpecifications and Challenges,” *Transactions on Environment and Electrical Engineering ,* vol. 3, no. 1, pp. 44-62, March 2019.

[16] R. Molina-Masegosa and J. Gozalvez, ““LTE-V for Sidelink 5G V2X Vehicular Communications: A New 5G Technology for Short-Range Vehicle-to-Everything Communications,” *IEEE vehicular technology magazine,* vol. 12, no. 4, pp. 30-39, December 2017.

[17] S. A. A. Shah, E. Ahmed, M. Imran and S. Zeadally, “IMMINENT COMMUNICATION TECHNOLOGIES FOR SMART COMMUNITIES: 5G for Vehicular Communications,” *IEEE Communications magazine,* p. 111–117, January 2018.

[18] Edward J. Oughton, William Lehr, Konstantinos Katsaros, Ioannis Selinis, Dean Bubley and Julius Kusuma, “Revisiting Wireless Internet Connectivity: 5G vs Wi-Fi 6,” *Telecommunications Policy,* vol. 45, no. 5, p. 102127, 2021.

[19] Ullah, Dr. Saleem, KHAN, Dost, Rehman, Aqeel, Siddique, M. and Khan, Abbas, “AN EFFECTIVE MODELING TO MINIMIZE THE TRANSMISSION TIME IN VANET BY REDUCING ROUTING OVERHEAD,” *Science International Lahore,* vol. 28, pp. 3041-3047, 2016.

[20] Cindy Goh and Cheng Leong Lim, “A Review of Network Models for Internet of Vehicles,” *VEHICULAR 2017 : The Sixth International Conference on Advances in Vehicular Systems, Technologies and Applications,* 2017.

[21] P. Alvares, L. Silva and N. Magaia, “Blockchain-Based Solutions for UAV-Assisted Connected Vehicle Networks in Smart Cities: A Review, Open Issues, and Future Perspectives,” *Telecom,* vol. 2, no. 1, pp. 108-140, 2021.

[22] F. Arena and G. Pau, “An Overview of Vehicular Communications,” *Future Internet,* vol. 11, no. 2, pp. 1-12, 2019.

[23] H. Liang and W. Zhuang, “Cooperative data dissemination via roadside WLANs,” *IEEE Communications Magazine,* vol. 50, no. 4, pp. 68-74, April 2012.

[24] P. R. Pereira, A. Casaca, J. J. P. C. Rodrigues, V. N. G. J. Soares, J. Triay and C. Cervello-Pastor, “From Delay-Tolerant Networks to Vehicular Delay-Tolerant Networks,” *IEEE Communications Surveys & Tutorials,* vol. 14, no. 4, pp. 1166-1182, 2012.

[25] F. Yang, S. Wang, J. Li, Z. Liu and Q. Sun, “An overview of Internet of Vehicles,” *China Communications,* vol. 11, no. 10, pp. 1-15, 2014.

[26] W. Wu, Z. Yang and K. Li, “Chapter 16 - Internet of Vehicles and applications,” in *Internet of Things*, Rajkumar Buyya and Amir, 2016, pp. 299-317.

[27] H. Peng, Le Liang, X. Shen and G. Y. Li, “Vehicular Communications: A Network Layer Perspective,” *IEEE Transactions on Vehicular Technology,* vol. 68, no. 2, pp. 1064-1078, 2019.

[28] M. Hossain, R. Hasan and S. Zawoad, “Trust-IoV: A Trustworthy Forensic Investigation Framework for the Internet of Vehicles (IoV),” in *2017 IEEE International Congress on Internet of Things (ICIOT)*, USA, 2017.

[29] Deepak Puthal, R. Ranjan and Jinjun Chen, “Big Data Stream Security Classification for IoT Applications,” in *Encyclopedia of Big Data Technologies*, Springer International Publishing, 2018, pp. 1-5.

[30] Sanjeev Kumar Mandal and A R Deepti, “A Review Paper on Encryption Techniques,” *International Journal of Research and Analytical Reviews (IJRAR),* vol. 6, no. 2, pp. 70-75, June 2019.

[31] V. Sharma and M. Sharma, “A Hybrid Cryptosystem approach for file security by using merging mechanism,” *2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT),* pp. 713-717, July 2016.

[32] R. Hunt, “PKI and digital certification infrastructure,” in *Proceedings. Ninth IEEE International Conference on Networks, ICON 2001*, 2001.

[33] F. Haidar, A. Kaiser, B. Lonc and P. Urien, “C-ITS PKI protocol: Performance Evaluation in a Real Environment,” *2019 15th Annual Conference on Wireless On-demand Network Systems and Services (WONS),* pp. 52-55, 2019.

[34] P. Beckett, “GDPR compliance: your tech department's next big opportunity,” *Computer Fraud & Security,* vol. 2017, no. 5, pp. 9-13, 2017.

[35] L. Wu, X. Yang, M. Zhang and L. Liu, “New identity based proxy re-encryption scheme from lattices,” *China Communications,* vol. 16, no. 10, pp. 174-190, 2019.

[36] M. Blaze, G. Bleumer and M. Strauss, “Divertible protocols and atomic proxy cryptography,” *Advances in Cryptology — EUROCRYPT'98,* pp. 127-144, 1998.

[37] Gelareh Taban, Alvaro A. Cárdenas and Virgil D. Gligor, “Towards a secure and interoperable DRM architecture,” Alexandria, Virginia, USA, 2006.

[38] H. Xiong, Z. Chen and F. Li, “Efficient privacypreserving authentication protocol for vehicular communications with trustworthy,” *Security and Communication Networks,* vol. 5, no. 12, pp. 1441-1451, 2012.

[39] Y. Chen, J. D. Tygar and W. Tzeng, “Secure group key management using uni-directional proxy re-encryption schemes,” *2011 Proceedings IEEE INFOCOM,* p. 1952–1960, April 2011.

[40] Giuseppe Ateniese, Kevin Fu, Matthew Green and Susan Hohenberger, “Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,” *ACM Trans. Inf. Syst. Secur.,* vol. 9, no. 1, pp. 1-30, February 2006.

[41] Anum Khurshid, Fiaz Gul Khan and Abdul Nasir Khan, “A Comparison of Proxy Re-Encryption Schemes – A Survey,” *International Journal of Computer Science and Information Security (IJCSIS),* vol. 14, no. 6, pp. 392-397, June 2016.

[42] Z. Qin, H. Xiong, S. Wu and J. Batamuliza, “A Survey of Proxy Re-Encryption for Secure Data Sharing in Cloud Computing,” *IEEE Transactions on Services Computing,* pp. 1-1, 2016.

[43] N. Jayashree and B. S. Babu, “ElGamal-based Privacy-Preseving Scheme (EPPS) for Edge-Cloud-of-Things (ECoT),” *2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS),* vol. 4, pp. 1-4, December 2019.

[44] J. Markus, “On Quorum Controlled Asymmetric Proxy Re-encryption,” in *Public Key Cryptography*, Berlin, Heidelberg, Springer Berlin Heidelberg, 1999, pp. 112-121.

[45] Anca Ivan and Yevgeniy Dodis, “Proxy Cryptography Revisited,” in *Proceedings of the Network and Distributed System Security Symposium*, 2003.

[46] B. Reddaiah, “A Study on Pairing Functions for Cryptography,” *International Journal of Computer Applications (0975 –8887),* vol. 149, no. 10, September 2016.

[47] M. Ayoub Khan and Y. P. Singh, “On the security of joint signature and hybrid encryption,” *2005 13th IEEE International Conference on Networks Jointly held with the 2005 IEEE 7th Malaysia International Conf on Communic,* vol. 1, pp. 4 pp.-, 2005.

[48] Ran Canetti and Susan Hohenberger, “Chosen-ciphertext secure proxy re-encryption,” in *Proceedings of the 14th ACM conference on Computer and communications security*, Alexandria, Virginia, USA, 2007.

[49] Ran Canetti, Hugo Krawczyk, Jesper B. Nielsen and Dan Boneh, “Relaxing Chosen-Ciphertext Security,” in *Advances in Cryptology - CRYPTO 2003*, Berlin, Heidelberg, 2003.

[50] Green, Matthew and Ateniese, Giuseppe, “Identity-based proxy re-encryption,” in *Proceedings of the 5th International Con-ference on Applied Cryptography and Network Security*, 2007.

[51] Dan Boneh and Matt Franklin, “Identity-Based Encryption from the Weil Pairing,” *Advances in Cryptology — CRYPTO 2001,* pp. 213-229, 2001.

[52] S. Adi, “Identity-Based Cryptosystems and Signature Schemes,” *Advances in Cryptology. CRYPTO 1984,* vol. 196, pp. 47-53, 1985.

[53] Glenn Durfee, Dirk Balfanz, Narendar Shankar, Diana Smetters, Jessica Staddon and Hao-Chi Wong, “Secret handshakes from pairing-based key agreements,” *Proceedings of the 2003 IEEE Symposiumon Security and Privacy (SP ’03),* p. 180, 2003.

[54] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky and Giuseppe Persiano, “Public key encryptionwith keyword search,” *Advances in Cryptology - EUROCRYPT 2004,* vol. 3027, p. 506–522, 2004.

[55] Brent R. Waters, Dirk Balfanz, Glenn Durfee and D. K. Smetters, “Building an encrypted and searchable audit log,” *Proceedings of Network and Distributed System Security Symposium 2004 (NDSS’04),* February 2004.

[56] Ran Canetti, Shai Halevi and Jonathan Katz, “Chosen-ciphertext security from Identity Based Encryption,” *Proceedings of Eurocrypt ’04,* vol. 3027 of Lecture Notes in Computer Science, p. 207–222, 2004.

[57] Dan Boneh and Matt Franklin, “Identity-based encryption from the Weil Pairing,” *SIAM Journal of Computing,* vol. 32, no. 3, p. 586–615, 2003.

[58] J. Weng, S. Liu, K. Chen and R. H. Deng, “Chosen-ciphertext secure proxy re-encryption without pairings,” *Proceedings of the 7th International Conference on Cryptology and Network Security,* pp. 1-17, 2008.

[59] B. Libert and D. Vergnaud, “Unidirectional Chosen-Ciphertext Secure Proxy Re-encryption,” *Public Key Cryptography–PKC’08,* pp. 360-379, 2008.

[60] J. Shao and Z. Cao, “Cca-secure proxy re-encryption without pairings,” *Public Key Cryptography–PKC’09,* p. 357–376, 2009.

[61] X. Liang, Z. Cao, H. Lin and J. Shao, “Attribute based proxy re-encryption with delegating capabilities,” *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security,* p. 276–286, 2009.

[62] J. Weng, R. H. Deng,, X. Ding, C.-K. Chu and J. Lai, “Conditional proxy re-encryption secure against chosen-ciphertext attack,” *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security,* p. 322–332, 2009.

[63] J. Weng, Y. Yang, Q. Tang, R. H. Deng and F. Bao, “Efficient conditional proxy re-encryption with chosen-ciphertext security,” *Proceedings of the 12th International Conference on Information Security,* p. 151–166, 2009.

[64] G. Ateniese, K. Benson and S. Hohenberger, “Key-private proxy re-encryption,” *Topics in Cryptology–CT-RSA’09,* p. 279–294, 2009.

[65] J. Weng, M. Chen, Y. Yang, R. Deng, K. Chen and F. Bao, “Cca-secure unidirectional proxy re-encryption in the adaptive corruption model without random oracles,” *Science China Information Sciences,* vol. 53, no. 3, p. 593–606, 2010.

[66] S. S. Chow, J. Weng, Y. Yang and R. H. Deng, “Efficient unidirectional proxy re-encryption,” *Progress in Cryptology–AFRICACRYPT’10,* p. 316–332, 2010.

[67] T. Matsuda, R. Nishimaki and K. Tanaka, “Cca proxy re-encryption without bilinear maps in the standard model,” *Public Key Cryptography–PKC’10,* p. 261–278, 2010.

[68] C. Sur, C. D. Jung, Y. Park and K. H. Rhee, “Chosen-ciphertext secure certificateless proxy re-encryption,” *Proceedings of the 11th International Conference Communications and Multimedia Security,* p. 214–232, 2010.

[69] S. S. Al-Riyami and K. G. Paterson, “Certificateless public key cryptography,” *Advances in Cryptology–ASIACRYPT’03,* p. 452–473, 2003.

[70] S. Lee, H. Park and J. Kim, “A secure and mutual-profitable drm interoperability scheme,” *Proceedingsof IEEE Symposium on Computers and Communications,* p. 75–80, 2010.

[71] Jun Shao, Zhenfu Cao, Xiaohui Liang and Huang Lin, “Proxy re-encryption with keyword search,” *Information Sciences,* vol. 180, no. 13, pp. 2576-2587, 2010.

[72] B. Libert and D. Vergnaud, “Unidirectional Chosen-Ciphertext Secure Proxy Re-Encryption,” *IEEE Transactions on Information Theory,* vol. 57, no. 3, pp. 1786-1802, March 2011.

[73] J. Shao, P. Liu, Z. Cao and G. Wei, “Multi-use uni-directional proxy re-encryption,” *IEEE International Conference on Communications,* p. 1–5, 2011.

[74] G. Hanaoka, Y. Kawai, N. Kunihiro, T. Matsuda, J. Weng, R. Zhang and Y. Zhao, “Generic construction of chosen ciphertext secure proxy re-encryption,” *Topics in Cryptology–CT-RSA’12,* p. 349–364, 2012.

[75] J. Shao, P. Liu and Y. Zhou, “Achieving key privacy without losing cca security in proxy re-encryption,” *Journal of Systems and Software,* vol. 85, no. 3, p. 655–665, 2012.

[76] T. Isshiki, M. H. Nguyen and K. Tanaka, “Proxy re-encryption in a stronger security model extended from ct-rsa2012,” *Topics in Cryptology–CT-RSA’13,* p. 277–292, 2013.

[77] L. Fang, W. Jiandong, G. Chunpeng and R. Yongjun, “Fuzzy conditional proxy re-encryption,” *SCIENCE CHINA Information Sciences,* vol. 56, no. 5, p. 1–13, 2013.

[78] J. Zhang, Z. Zhang and Y. Chen, “Pre: Stronger securitynotions and efficient construction with non-interactive opening,” *Theoretical Computer Science,* vol. 542, no. 0, p. 1–16, 2014.

[79] E. Kirshanova, “Proxy re-encryption from lattices,” *Public Key Cryptography–PKC’14,* p. 77–94, 2014.

[80] H. Guo, Z. Zhang and J. Zhang, “Proxy re-encryption with unforgeable re-encryption keys,” *Proceedings of the 13th International Conference on Cryptology and Network Security,* p. 20–33, 2014.

[81] Q. Liu, G. Wang and J. Wu, “Time-based proxy re-encryption scheme for secure data sharing in a cloud environment,” *Information Sciences,* vol. 258, p. 355–370, 2014.

[82] Fei Tang, Hongda Li and Jinyong Chang, “Multi-hop unidirectional proxy re-encryption from multilinear maps,” *IEICE Transactions on Fundamentals of Electronics,* vol. 98, no. 2, p. 762–766, 2015.

[83] Sanjam Garg, Craig Gentry and Shai Halevi, “Candidate Multilinear Maps from Ideal Lattices,” *International Association for Cryptologic Research 2013,* p. 1–17, 2013.

[84] D. Nunez, I. Agudo and J. Lopez, “A parametric family of attack models for proxy re-encryption,” *IEEE 28th Computer Security Foundations Symposium (CSF 2015),* p. 290–301, 2015.

[85] D. Hua, W. Qianhong, Q. Bo, S. Willy, L. Joseph and S. Wenchang, “Asymmetric cross-cryptosystem re-encryption applicable to efficient and secure mobile access to outsourced data,” *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security (ASIA CCS ’15),* p. 393–404, 2015.

[86] Z. Yunya, D. Hua, W. Qianhong, Q. Bo, L. Jianwei and D. Yong, “Identity-based proxy re-encryption version 2: Making mobile access easy in cloud,” *Future GenerationComputer Systems, p. in press,* 2015.

[87] Yang Lu and Jiguo Li, “A pairing-free certificate-based proxy re-encryption scheme for secure data sharing in public clouds,” *Future Generation Computer Systems,* vol. 62, pp. 140-147, 2016.

[88] P. Xu, T. Jiao, Q. Wu, W. Wang and H. Jin, “Conditional Identity-Based Broadcast Proxy Re-Encryption and Its Application to Cloud Email,” *IEEE Transactions on Computers,* vol. 65, no. 1, pp. 66-79, 2016.

[89] David Nunez, Isaac Agudo and Javier Lopez, “Proxy Re-Encryption: Analysis of constructions and its application to secure access delegation,” *Journal of Network and Computer Applications,* vol. 87, pp. 193-209, 2017.

[90] Linmei Jiang and Donghui Guo, “Dynamic Encrypted Data Sharing Scheme Based on Conditional Proxy Broadcast Re-Encryption for Cloud Storage,” *IEEE Access,* vol. 5, pp. 13336-13345, 2017.

[91] Y. Yasumura, H. Imabayashi and H. Yamana, “Attribute-based proxy re-encryption method for revocation in cloud storage: Reduction of communication cost at re-encryption,” *2018 IEEE 3rd International Conference on Big Data Analysis (ICBDA),* pp. 312-318, 2018.

[92] N. Attrapadung and H. Imai, “Conjunctive broadcast and attribute-based encryption,” *Pairing-Based Cryptography-Pairing 2009,* vol. 5671, pp. 248-265, 2009.

[93] M. Horváth, “Attribute-Based Encryption Optimized for Cloud Computing,” *SOFSEM 2015: Theory and Practice of Computer Science,* vol. 8939, pp. 566-577, 2015.

[94] S. Yu, C. Wang, K. Ren and W. Lou, “Attribute based data sharing with attribute revocation,” *Proc. of the 5th ACM Symp. on Information Computer and Communications Security,* pp. 261-270, 2010.

[95] Zhiwei Wang, “Leakage resilient ID-based proxy re-encryption scheme for access control in fog computing,” *Future Generation Computer Systems,* vol. 87, pp. 679-685, 2018.

[96] Qinlong Huang, Yixian Yang and Jingyi Fu, “PRECISE: Identity-based private data sharing with conditional proxy re-encryption in online social networks,” *Future Generation Computer Systems,* vol. 86, pp. 1523-1533, 2018.

[97] Mohamed Nahri, Azedine Boulmakoul, Lamia Karim and Ahmed Lbath, "IoV distributed architecture for real-time traffic data analytics," *Procedia Computer Science,* vol. 130, pp. 480-487, 2018.

[98] S. C. Linga Reddy and Chandrakala B M, "Proxy Re-Encryption using MLBC (Modified Lattice Based Cryptography)," *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC),* pp. 1-5, 2019.

[99] R. P. Meiliasari, A. Syalim and S. Yazid, "Performance Analysis of the Symmetric Proxy Re-encryption Scheme," *2019 International Workshop on Big Data and Information Security (IWBIS),* pp. 91-96, 2019.

[100] L. Wu, X. Yang, M. Zhang and L. Liu, "New identity based proxy re-encryption scheme from lattices," *China Communications,* vol. 16, no. 10, pp. 174-190, 2019.

[101] A. Manzoor, M. Liyanage, A. Braeke, S. S. Kanhere and M. Ylianttila, "Blockchain based Proxy Re-Encryption Scheme for Secure IoT Data Sharing," *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC),* pp. 99-103, 2019.

[102] Iuon-Chang Lin, Chen-Yang Cheng and Hsiang-Yu Chen, "A real-time parking service with proxy re-encryption in vehicular cloud computing," *Engineering Applications of Artificial Intelligence,* vol. 85, pp. 208-213, 2019.

[103] S. Maiti and S. Misra, "P2B: Privacy Preserving Identity-Based Broadcast Proxy Re-Encryption," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 5, pp. 5610-5617, 2020.

[104] S. Sharma, A. Swarnakar, C. J. Babu, R. Padmavathy and R. Kumar, "An Authenticated Keyword Searchable Conditional Proxy Re-encryption Scheme in Cloud Services," *2020 5th International Conference on Computing, Communication and Security (ICCCS),* pp. 1-8, 2020.

[105] Priyanka Dutta, Willy Susilo, Dung Hoang Duong and Partha Sarathi Roy, "Collusion-resistant identity-based Proxy Re-encryption: Lattice-based constructions in Standard Model," *Theoretical Computer Science,* 2021.

[106] Fucai Luo, Saif Al-Kuwari, Fuqun Wang and Kefei Chen, "Attribute-based proxy re-encryption from standard lattices," *Theoretical Computer Science,* vol. 865, pp. 52-62, 2021.

[107] C. Peikert, "A Decade of Lattice Cryptography," *Now Publishers Inc.,* vol. 10, no. 4, 2016.

[108] Palazzo, Sergio, Campbell, Andrew T. and Dias de Amorim, Marcelo, "Opportunistic and Delay-Tolerant Networks," *EURASIP Journal on Wireless Communications and Networking,* 2011.

[109] C. B. a. G. D. a. A. G. a. A. Zacharopoulos, "Comparison of 4G and 5G Network Simulators," *The Fifteenth International Conference on Wireless and Mobile Communications (ICWMC 2019),* pp. 13-18, 2019.

[110] "NS3 Network Simulator," [Online]. Available: https://www.nsnam.org/. [Accessed 09 2021].

[111] S. Damien and R. Steinfeld, "Making NTRU as Secure as Worst-Case Problems over Ideal Lattices," *Advances in Cryptology -- EUROCRYPT 2011,* pp. 27-47, 2011.

[112] J. Hoffstein, J. Pipher and J. H. Silverman, “NTRU: A ring-based public key cryptosystem,” *Lecture Notes in Computer Science Algorithmic Number Theory,* pp. 267-288, 1998.

[113] Chris Peikert, “A Decade of Lattice Cryptography,” *Foundations and Trends in Theoretical Computer Science,* vol. 10, no. 4, pp. 283-424, 2016.

[114] Larissa Cherckesova, Olga Safaryan, Pavel Razumov, Veronica Kravchenko, Sergey Morozov and Alexey Popov, “Post-Quantum Cryptosystem NTRUEnCrypt and Its Advantage over Pre – Quantum Cryptosystem RSA,” in *E3S Web of Conferences*, Russia, 2020.

[115] Bo Mi, Darong Huang and Shaohua Wan, “NTRU Implementation of Efficient Privacy-PreservingLocation-Based Querying in VANET,” *Wireless Communications and Mobile Computing,* 2018.

[116] Hoffstein J., Pipher J., Schanck J.M., Silverman J.H., Whyte W. and Zhang Z., “Choosing Parameters for NTRUEncrypt,” *Topics in Cryptology – CT-RSA 2017,* Vols. Lecture Notes in Computer Science, vol 10159, 2017.

[117] William Whyte, Nick Howgrave-Graham, Jeff Hoffstein, Jill PIpher, Joseph H. Silverman and Phil Hirschhorn, “IEEE P1363.1 Draft 10: Draft Standard for Public Key Cryptographic Techniques Based on Hard Problems over Lattices,” in *IACR Eprint archive*, 2008.

[118] J. Hermans, F. Vercauteren, B. Preneel and J. Pieprzyk, “Speed Records for NTRU,” *Topics in Cryptology - CT-RSA 2010,* pp. 73-88, 2010.

[119] W. Whyte, “NTRU,” *In: van Tilborg H.C.A. (eds) Encyclopedia of Cryptography and Security,* pp. 427-430, 2005.

[120] Y. Aono, X. Boyen, L. T. Phong and L. Wang, “Key-private proxy re-encryption under LWE,” *Progress in Cryptology–INDOCRYPT 2013,* p. 1–18, 2013.

[121] Niklas Kuse and Benedikt Jaeger, “Network Simulation with ns-3,” *Seminar IITM SS 20, Network Architectures and Services,* November 2020.

[122] K. Katsaros, “Vehicular communication simulations with NS-3,” [Online]. Available: https://www.nsnam.org/tutorials/consortium15/Vehicular-Comms.pptx.

[123] N.-3. nsnam, “NS-3 WAVE Module,” [Online]. Available: https://www.nsnam.org/doxygen/group__wave.html.

[124] K.Sjöberg, “Standardization of Wireless Vehicular Communications within IEEE and ETSI,” *IEEE VTS Workshop on Wireless Vehicular Communications,* p. 25, November 2011.

[125] n. NS-3, “https://apps.nsnam.org/app/mmwave/,” [Online].

[126] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan and M. Zorzi, “End-to-End Simulation of 5G mmWave Networks,” *IEEE Communications Surveys & Tutorials,* vol. 20, no. 3, pp. 2237-2263, 2018.

[127] Tommaso Zugno, Michele Polese and Michele Zorzi, “Integration of carrier aggregation and dual connectivity for the ns-3 mmWave module,” *Proceedings of the 10th Workshop on ns-3,* p. 45–52, 2018.

[128] Sourjya Dutta, Menglei Zhang, Marco Mezzavilla, Mustafa Riza Akdeniz and Sundeep Rangan, “Millimeter wave module for ns-3 network simulator,” 2015.

[129] Gustavo Carneiro, Pedro Fortuna and Manuel Ricardo, “FlowMonitor - a network monitoring framework for the Network Simulator 3 (NS-3),” *2nd International ICST International Workshop on Network Simulation Tools,* 2010.

[130] G. Carneiro, P. Fortuna and M. Ricardo, “FlowMonitor: a network monitoring framework for the network simulator 3 (NS-3),” *Proceedings of the Fourth International ICST Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS ‘09),* 2009.

[131] Pei, Zhonghui, Chen, Wei, Zheng, Hongjiang and Du, Luyao, “Optimization of Maximum Routing Hop Count Parameter Based on Vehicle Density for VANET,” *Mobile Information Systems,* 2020.