UNIVERSITY
INSTITUTE
OF LISBON

Post-Quantum Cryptography in Automotive Context

Mariana do Carmo Barata Gonçalves

Master in Telecommunications and Computer Engineering

Supervisor:
PhD Valderi Reis Quietinho Leithardt, Assistant Professor with Aggregation,
Iscte – Instituto Universitário de Lisboa

May, 2026

TECHNOLOGY
AND ARCHITECTURE

Department of Information Science and Technology

Post-Quantum Cryptography in Automotive Context

Mariana do Carmo Barata Gonçalves

Master in Telecommunications and Computer Engineering

Supervisor:
PhD Valderi Reis Quietinho Leithardt, Assistant Professor with Aggregation,
Iscte – Instituto Universitário de Lisboa

May, 2026

To my parents who supported me every step of my academic journey.

Acknowledgments

I would like to express my deepest appreciation to my supervisor, Valderi Leithardt, who guided me throughout this process and permitted me to explore this topic.

Resumo

Este estudo analisa a viabilidade da criptografia pós-quântica no contexto automóvel, tendo em conta a crescente ameaça representada por algoritmos quânticos, tais como o Algoritmo de Shor e o Algoritmo de Grover. A simulação é composta pelo OMNeT++, Veins, INET, SUMO, dois hardwares distintos e três esquemas de PQC. A avaliação ocorre em dois cenários: V2V, representando um ambiente rural, e V2X, simulando um ambiente urbano juntamente com o suporte de uma Unidade de Beira de Estrada. Os esquemas selecionados foram: o CRYSTALS-Kyber, que apresenta baixa latência durante a fase de handshake, tornando-o adequado para cenários com requisitos em tempo real, o CRYSTALS-Dilithium, que oferece desempenho equilibrado e o Falcon que implica altos custos computacionais para a geração de chaves e assinatura, mas se destaca na verificação de assinaturas. No geral, o Falcon revela-se muito adequado para protocolos futuros devido à sua eficiência na operação de verificação. Os algoritmos analisados são viáveis para integração em sistemas automóveis, embora seja necessário considerar os compromissos entre latência, custo computacional e os requisitos específicos de cada cenário. Todos os resultados foram compilados num conjunto de dados, o que poderá ajudar futuros investigadores em estudos relacionados com este campo.

Palavras Chave: *Criptografia Pós Quântica*, *Criptografia*, *Automóvel*, *Vehicle-to-Vehicle*, *Vehicle-to-Everything*

Abstract

This study analyzes the feasibility of post-quantum cryptography in the automotive context, considering the growing threat posed by quantum algorithms such as the Shor Algorithm and Grover's Algorithm. The simulation is composed by OMNeT++, Veins, INET, SUMO, two distinct hardwares and three PQC schemes. The evaluation occur in two scenarios: V2V, representing a rural environment, and V2X, simulating an urban environment supported by a roadside unit. The selected schemes were, CRYSTALS-Kyber which exhibits low latency during the handshake phase, making it suitable for scenarios with real-time requirements, CRYSTALS-Dilithium which offers balanced performance and Falcon that incurs high computational costs for key generation and signing, but excels in signature verification. Overall Falcon proofs to be very suitable for future protocols due to its efficiency on verification operation. The analyzed algorithms are viable for integration into automotive systems, though it is necessary to consider the trade-offs between latency, computational cost, and the specific requirements of each scenario. All the results were compiled into a dataset, which may assist future researchers in studies related to this field.

KEYWORDS: *Post Quantum Cryptography*, *Cryptography*, *Vehicle-to-Vehicle*, *Vehicle-to-Everything*, *Automotive*

Contents

Acknowledgments i

Resumo iii

Abstract v

List of Figures ix

List of Tables xi

List of Symbols xiii

List of Acronyms xv

Chapter 1. Introduction 1

1.1. Motivation 2

1.2. Research Questions and Objectives 2

1.3. Document Structure 3

1.4. Methodology 3

Chapter 2. Literature Review 5

2.1. Cryptography 5

2.1.1. Transition of Quantum to Post-Quantum Cryptography 5

2.1.2. Types of Post-Quantum Cryptography (PQC) 7

2.1.3. NIST Standard PQC algorithms 10

2.2. Automotive 15

2.2.1. Vehicle-To-Everything (V2X) 15

2.2.2. Attacks 16

Chapter 3. Related Work 19

3.1. Post-Quantum Secure Architeture for Automotive HSM (Wang et al., 2020) 19

3.2. Post-Quantum Secure Over-the-Air Update of Automotive Systems (W.Bos Joppe et al., 2023) 21

3.3. Impacts of PQC on Automotive Security (Rathweg et al., 2024) 22

3.4. Post-Quantum Authentication for V2V Communications (McCarthy et al., 2024) 24
3.5. Other Relevant Works 26

Chapter 4. Solution 31
4.1. Methodology 31
4.2. Tools 31
4.2.1. Hardware 31
4.2.2. OMNeT++ 33
4.2.3. Selected PQC Schemes 33
4.2.4. PQC Libraries 34

Chapter 5. Results 37
5.1. Rural Scenario (V2V) 37
5.2. Urban Scenario (V2X) 39

Chapter 6. Conclusion 43
6.1. Limitations 43
6.2. Future Works 44

References 45

Appendix A. Architecture 51

Appendix B. Submitted Article 53

List of Figures

Figure 1.1	DSR Methodology Process Model [8]	3
Figure 2.1	BB84 protocol scheme shown in [10]	5
Figure 2.2	Example of good and bad bases	7
Figure 2.3	Alice and Bob's private keys according to [23]	9
Figure 2.4	Alice and Bob's shared secret on [23]	10
Figure 3.1	Performance of the different security levels of Key Management Systems (KEM) schemes demonstrated in [35]	23
Figure 3.2	Performance of the different security levels of signature schemes observed in [35]	23
Figure 4.1	Architecture of the experimental simulation	35
Figure 5.1	Rural Scenario - Car alerting others cars of his accident	38
Figure 5.2	Urban Scenario - Car alerting the Roadsite Unit (RSU) and others vehicles	39
Figure 5.3	Comparison of PQC performance on both scenarios	41
Figure A.1	Full architecture of the developed application	51

List of Tables

Table 2.1	Each PQC scheme strengths and weaknesses according to [19] and [20]	10
Table 3.1	Related Work papers	30
Table 4.1	Hardware 1 - Components used in the simulation	31
Table 4.2	Hardware 2 - Components used in the simulation	31
Table 4.3	Main characteristics of each scheme approached in this work	33
Table 5.1	PQClean Schemes - Sizes (Bytes)	37
Table 5.2	Kyber Rural Scenario	38
Table 5.3	Dilithium - Rural Scenario	38
Table 5.4	Falcon - Rural Scenario	39
Table 5.5	Kyber - Urban Scenario	40
Table 5.6	Dilithium - Urban Scenario	40
Table 5.7	Falcon - Urban Scenario	41

List of Symbols

a, A: Alpha

$\cong$: Approximately Equal

$||$: Concatenation

E: Epsilon

$f^{-1}(x)$: Inverse Function

$\log_a$: Logarithm to Base a

μ: Mu

ϕ: Phi

ψ: Psi

$\sum$: Sigma

$\surd$: Square Root

T: Tau

v: Ypsilon

List of Acronyms

BHT: Brassard–Høyer–Tapp

CAN: Controller Area Network

CVP: Closest Vector Problem

C-V2X: Cellular-Vechile-to-Everything

DH: Diffie-Hellman

DSR: Design Science Research

DSRC: Dedicated Short-Range Communication

ECC: Elliptic Curve Cryptography

ECDH: Elliptic Curve Diffie-Hellman

ECDSA: Elliptic Curve Digital Signature Algorithm

ECU: Electrical Control Unit

FIPS: Federal Information Processing Standards

FORS: Forest of Random Subsets

GCD: Greatest Common Divisor

HNDL: Harvest now, decrypt later

HQC: Hamming Quasi-Cyclic

HSE: Hardware Secure Engine

HSM: Hardware Secure Modules

IoT: Internet of Things

KEM: Key Encapsulation Mechanism

KEM: Key Management Systems

LIDAR: Light Detection and Ranging

LLL: Lenstra, Lenstra and Lovász

LWE: Learning With Errors

MLWE: Module Learning with Errors

mMTC: massive Machine-type Communication

MSIS: Module Short Integer Solution

NTRU: Nth-degree Truncated polyminal Ring Units

NTT: Number Theoric Transform

OTA: Over-the-Air

PQC: Post-Quantum Cryptography

PRF: Pseudorandom Function

QC-MDPC: Quasi-Cyclic Moderate Density Parity-Check

RSA: Rivest-Shamir-Adleman

RSU: Roadsite Unit

SDV: Software Defined Vehicle

SIDH: Supersingular Isogeny-Based Diffie-Hellman

SVP: Shortest Vector Problem

URLLC: Ultra-Reliable Low-Latency Communication

V2X: Vehicle-To-Everything

V2V: Vehicle-to-Vehicle

V2I: Vehicle-to-Infrastructure

V2P: Vehicle-to-Pedestrian

V2N: Vehicle-to-Network

V2D: Vehicle-to-Device

WOTS: Witernitz One Time Signatures

CHAPTER 1

Introduction

As time passes, everything around us tends to evolve, becoming more intelligent, although it comes with its disadvantages. At this date, we are dealing with constant attacks to our communications and now with new quantum era reaching its peak, it is only rational to come up with ways to defend ourselves. Intelligent devices that require cryptography, operate with algorithms which depend on factorizing large numbers for example, Rivest-Shamir-Adleman (RSA) and Elliptic Curve Cryptography (ECC) which depends on discrete logarithm problem. These algorithms may be in serious danger due to two quantum algorithms that were developed which surpass traditional cryptographies: Shor's algorithm and Grove's algorithm. These two algorithms are the main reason to start building up defenses agaisnt quantum era, with that Post-Quantum Cryptography (PQC) was created. Even though we currently are very scarce on quantum computer power, with the most powerful ones being Magne from Denmark with exactly 1200 physical qubits [1], [2], 50 logical qubits and ABCI-Q from Japan having 2020 NVIDIA H100 GPUs and it is an huge step in order to protecting us from quantum hazards. According to the media [3], "Quantum computing is witnessing rapid advance toward commercial and research uses", many companies show an huge interest in quantum power making them invest billions in order to complete their company's timeline objectives. For example, IBM has updated their roadmap and intends to reach a quantum-centric supercomputer by the end of 2025 with over 4000 qubits [3]. Throughout the development of quantum computers, NIST has taken action and announced a contest with the intent of standardization of various PQC schemes and in 2024, several were officially approved. And now has come the time to implement these schemes in real-life scenarios. Since 1885, humans rely on vehicles to locomote one place to another, and with all the technogical evolution, they became smarter e.g, capable of making calls, open the doors with the key fob. But all of that requires extensive communication between systems, and for our own privacy, these systems involve cryptographic algorithms as mentioned before (RSA, ECC) [4], it's time to elaborate different methodologies involving PQC. The automotive PQC market is all about hardware, software and services idealized to defend components from quantum threats. These attacks can affect Vehicle-To-Everything (V2X) communication, Over-the-Air (OTA) updates, autonomous driving systems and Electrical Control Unit (ECU) communication [5]. It's expected that the market will continue to grow significantly until 2034, with the factors of growth being Electrification of vehicles and Advanced Driver Assistance Systems (ADAS) [1], [5].

1.1. Motivation

Automotive is a rising industry used by billions every single day. It should be a priority to interconnect vehicles utilizing V2X, specially when it comes to emergencies (i.e., ambulances or fire trucks) where road safety is critical. Future functionalities like autonomous mobility and traffic management could be innovative or even life changing. Another reason for engaging with this topic, is the inevitable economically growth of quantum computing. Multiple organizations are investing billions in quantum, so we can expect in a couple of years, quantum computing will play a big role in the technology business [6].

At this technogical stage, it is crucial to combine V2X with PQC. PQC is becoming necessary security-wise, since traditional methods are struggling to protect our information due to quantum algorithms. To this day, there are not many researchers working on this topic since it is still fresh, with NIST only standardizing algorithms in 2022 [7]. This work motivation comes from a deep interest for cybersecurity and the desire to assemble two big concepts: PQC and V2X. There is a lack of comprehensive studies evaluating both performance and feasibility of PQC in automotive context. The purpose of this paper is to provide further insight into this topic.

1.2. Research Questions and Objectives

PQC is still a big question mark when it comes to implementing to the automotive industry. Many wonder how can they otimize quantum algorithms in order to make them perform in embedded systems like a ECU without having issues. Since automotive networking is evolving so fast with V2X and soon we can reach autonomous driving and preventing accidents, it leaves us with the task of testing how do PQC algorithms perform under V2X by evaluating the latency and size of the data they need to complete their role. That brings on some possible questions to move forward with our research:

- How does PQC affect V2X communications? (will it worsen performance by increasing security?)
- Which are the most effective PQC algorithms for V2X?
- How can we otimize latency concerning the of automotive network when implementing PQC?

In a simulated scenario, our objective is to test multiple PQC schemes (CRYSTALS-Kyber, CRYSTALS-Dilithium and Falcon) and observe parameters like key sizes, time of funcionalities (i.e., key generation), measure the overhead. Before analyzing the different parameters, we can state some concrete objectives for this study:

- Analyze how PQC affects V2X performance.
- Determine which are the implementations that success more on V2X communications.

1.3. Document Structure

This document is composed by six chapters. The first chapter explains what is objetive of the work, the problem, its questions that intend to solve, what methodology is used and the author's motivation for perform this reasearch. In chapter 2 we focus mainly on explaining to the reader the main concepts that will be approached to contextualize the reader. Chapter 3 performs a research on articles within the topic, to analyze and use as a guide to assist our writing and studying the subject. Section 4 provides a step by step solution for the problem mentioned in chapter one, by addressing all the tools used. In 5 is all about results from the solution from previous chapters, we analyze each result and take and compare them until we reach a conclusion. Finally in chapter 6 regards to the conclusions of everything declared on all previous chapters and sugesting possible future works. If there is any limitations, this is the place to expose them.

1.4. Methodology

This work will implement Design Science Research (DSR), proposed by Brocke Hevner, Maedche [8]. It focus on the process of solving problems by expanding knowledge (by developing artifacts) on a specific field, so that experts can use it to formulate possible new solutions for their specific problems possibly, real-world problems.

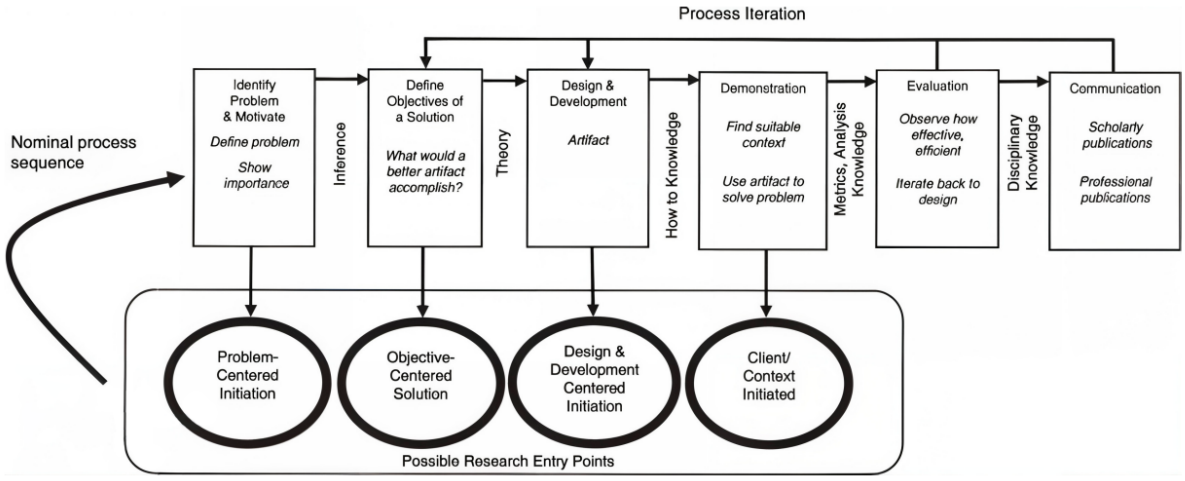


FIGURE 1.1. DSR Methodology Process Model [8]

According to Figure 1.1, DSR is composed by 6 steps:

(1) Identify the Problem and Motive: Defining a problem and state one possible solution. The defined problem is going to be used to develop a artifact that can provide a solution.

(2) Define Objetives of a Solution: Focused on establishing the objetives of the previous solution and understanding if is feasible. There are two types of objectives:

quantitative (e.g., according with solution would them be better than current ones) and qualitative (e.g., report how the newly artifact is going to assist on solution's problems).

(3) Design and Development: This step is about the artifact that we intend of making. First we idealize the artifact's functionality, architeture and then we start the development. Knowledge theory is fundamental to this step in order to move forward.

(4) Demonstration: Once the artifact is complete, it is ready for the demonstration step, here we show how the artifact is able to solve one or more instances of the problem. Involving experiments, simulations, cases studies, or any appropriate activity.

(5) Evaluation: After demonstrating, we need to evaluate our results, measure how well the artifact assists on solving the problem. This step is all about comparing the solution's objetives to the obtained results from the previous activity. Evaluation can be done by many ways, by comparing the artifact's functionality with solution's objetives as mentioned before. Quantitative performance can estimate statistics as satisfaction, utility.

(6) Communication: Relate the problem and its importance. Report about the developed artifact, his functionalities, design and how functional is for researchers and others who share similar objetives. This methodology's structure is also very suitable for academic research papers since it has the same structure.

CHAPTER 2

Literature Review

This section of the work is dedicated to explore essential concepts regarding PQC in order to contextualize the reader. It is divided by 3 sections, we commence by briefly explaining the evolution of cryptography, then proceed to state types of PQC and finally addressing NIST standardization contest winners.

2.1. Cryptography

2.1.1. Transition of Quantum to Post-Quantum Cryptography

Quantum cryptography objetive is not to replace classic cryptography but to ensure security between two users. Weisner [9] was the first to introduce Quantum in the cryptographic field by describing the concept of polarization. Followed by Wiesner inspiration came Bennet and Brassard who designed their own cryptography protocol known as BB84.

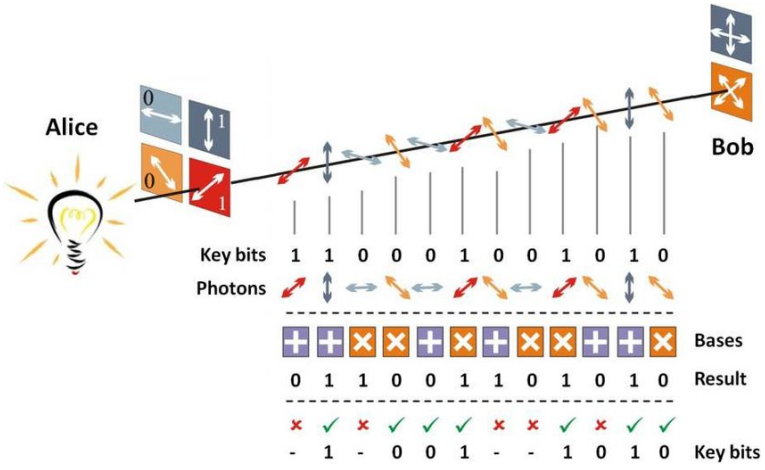


FIGURE 2.1. BB84 protocol scheme shown in [10]

BB84 protocol utilizes the concept of polarization, Wiesner mentioned in his work, that results can varie depending on photons and bases. As observed Alice determines the polarization (horizontal, vertical, left-circular or right-circular) of each burst of photons she is going to deliver to Bob [11]. When a light is emitted, the photons are filtered conformally their polarization. Bob generates his base sequence and measures each photon polarization. Bob then proceeds to inform Alice about his base sequence, without caring about eavesdroppers. Alice declares publicly which bases were correctly chosen. Alice and Bob focus solely on correct the right bases then convert them into binary code. This protocol is capable of correcting errors as well.

Unfortunally this method is very far from perfect. It is impossible to utilize single photons to transmit information, instead small bursts of light are used. So Eve could possibly isolate photons from the light beam and measure them. Current limitations include expensiveness, there are only a few banks that already took advantage of Quantum Cryptography. Not only that, this security method has to change in order to function in the real-world, since there are some practical issues: how are they going to maintain the photons entangled long enough. The bigger issue, for longer distances, noise becomes predominant that the error rates gets extremely high, leaving the channel very susceptible of eavesdropping and equally it is impossible to transmit information [11].

The Quantum era also introduced two algorithms that could potentially ruin all the classic cryptography by solving the mathematical problems in polinomial time: Shor's algorithm and Grover's algorithm. Shor's Algorithm consists on fiding prime factors of large numbers by selecting a random integer x, smaller than the number to be factored n [12], ensuring that x is coprime to n (i.e., their greatest Greatest Common Divisor (GCD) equals 1). The GCD of the two numbers is utilized to calculate whether target number has already been factored. It runs $O((\log n)^2 \cdot \log\log n)$ on a quantum computer and must run $O(\log n)$ on a classical computer [13]. This algorithm stems from the number theory:

$$F(a) = x^a \mod n \tag{2.1}$$

It is a periodic function [13]. Quantum Parallelism ("ability of a quantum memory register to exist in a superposition of base states" [13]) is utilized to execute exponential number of operations in one step to determine the period of n. And by determinating that period, we can calculate the factors of n.

Grover's algorithm was designed to accelarate searching tasks of quantum computers. It is renowned for their efficiently detailed search through unsorted databases, obviously better than the classic methods that use $O(\log n)$, due to their use of quantum power requiring only $O(\log \sqrt{n})$. It all begins with the creation of a quantum superposition (i.e,. uniform distribution where every solution has equal probability) [14] of every possibility. Proceeding by analyzing all solutions simultaneously. Basically this algorithm repeats two operations: oracle and diffusion [14]. The oracle operation is responsible for fliping the phase of marked items, those which satisfy your criteria. It divides the search by two sections, solutions and non-solutions. And the diffusion operation makes an amplification on the probabilities, so that the values can difer from the mean [14]. Since cryptography depends on computational difficult that rely on fiding certain values, it is understandable how Grover's algorithm can be dangerous, specially on brute-force attacks.

Researchers have taken a step ahead and developed PQC, a way to overcome these strong quantum algorithms. Implementing PQC now is necessary, since various governments and standards bodies are preparating quantum readiness. For example in [15], there's a clear limitation: no form of post-quantum cryptography is used. Therefore, the author

of [16] proposed an "adaptive middleware architecture" that incorporates the features of the PRISEC model, but now with integrated post-quantum cryptography. Through supervised, unsupervised and reinforcement learning, Q-OPSEC analyzes the information and dynamically selects the most appropriate scheme for that situation. This architecture consists of both classical cryptography and post-quantum cryptography.

Unfortunally Harvest now, decrypt later (HNDL) is a way to intercepting and recording encrypted traffic now for later decryption once a quantum computer can break existing cryptographic algorithms.

2.1.2. Types of PQC

Here we explain most of the current PQC which have been developed and successfully securing agaisnt quantum threats. Each of presents advantages and vulnerabilities, which we will approach later on. The stable methods, all differ from their correspondent mathematical foundations.

2.1.2.1. *Lattice-based*

In a two-dimensional space, a lattice is represented by two vectors, these vectors form a base. Bases can be defined as good and bad. Good bases require vectors to be orthogonally aligned as observed in 2.2, they are ideal for solving computational problems. Meanwhile bad bases are practically parallel [17].

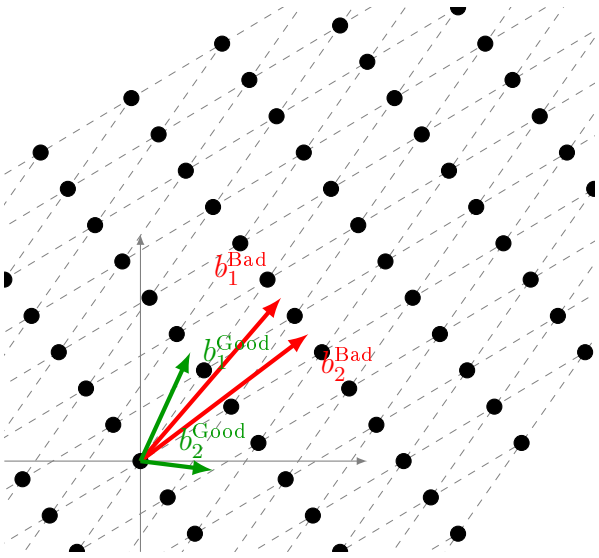


FIGURE 2.2. Example of good and bad bases

In cryptographic context, lattices are generally in 500-dimensional space. In this environment it would require a lot of time to find a good base. Lattice-based problems include Shortest Vector Problem (SVP), given a basis of a lattice, find a non-zero lattice vector that reaches the destination point. And Closest Vector Problem (CVP), given a basis of a lattice, and a target vector, find a lattice vector that reaches the destination [18], [19], [20]. Lattice reduction is used to solve these problems. It consists on transforming a

given lattice base into a good base, as mentioned before, a base with short and orthogonal vectors. The Lenstra, Lenstra and Lovász (LLL) algorithm is capable of achieving that lattice transformation [18].

2.1.2.2. *Hash-based*

Mathematically speaking, hash-based cryptography is much simpler when compared with the others types of PQC, but only can be utilized for digital signatures, being unsuitable for encryption. Not only that but part of the private key must be made public of each signature, which limits the number of signatures per key [17], [19].

Basically hash functions are functions that use input data of a certain size and produce fixed-length output, known as hash. Advantages of using hash functions are the one-way property and collision resistance. The one-way property refers to the difficulty of recovering the input data from the hash. Collision resistance means that an input only produces a unique hash, no other input will produce the same hash, having no hash collision [21], [20]. When protecting the integrity of data and detecting tampering, hashes functions are ideal due to their properties.

2.1.2.3. *Code-based*

Code-based cryptography focuses on the study of cryptosystems based on error-correcting codes. The corruption of one bit due to an unreliable transmission channel is sufficient to cause a disaster. Transmission reliability can be enlarged, by encoding the data package with extra redundant bits which can be designated as a code word and only some values in that space are considered valid code words. The modification of some bits would provide an invalid code word, when that happens an error is detected. The Hamming distance is defined by the space between code words measured by the numbers of bits that are different in each. Hamming weight is another important concept. It refers to the number of bits set to 1 in a binary representaiton of a value [22], [20]. In order to make a code word w, it is necessary to multiply the base value a by a matrix G designated by generator.

$$w = a.G \tag{2.2}$$

Matrix G is composed by the concatenating of two arrays: identity matrix I (squared matrix with all zeros execept the diagonal values) and parity matrix P which has the error coding values. Not every values of size w map are valid code words, so we can use w to determine if there was any sign of corruption if the received value on the channel w' isn't a valid code word. With the aim of correcting w', we can leverage P to produce a parity check matrix H as observed in equation 2.3 [22], [19].

$$H = [P^T] \cdot [I] \tag{2.3}$$

H is utilized to calculate the syndrome s (2.4), if s equals 0 then we can assume that w' is a valid code word. Unfortunally, it is not capable of correcting more than 1 error, since 2 errors will provide a syndrome that already corresponds to a error in syndrome table

and correcting the wrong column creating yet another error.

$$s = H \cdot w' \tag{2.4}$$

2.1.2.4. *Isogeny-based*

An elliptic curve is defined as a curve in a 2 dimensional space which contains all points within $y^2 = x^3 + ax + b$. It is used in ECC, which isn't secure enough against a quantum adversary. So it flourished a new concept, an isogeny, it consists of a rational map between elliptic curves. The reason this cryptography is so secure agaisnt quantum power, relies on the problem which is fiding an explicit isogeny between two isogenous supersingular elliptic curves over a finite field [19], [20].

Within Isogeny-based cryptography there are multiple variations of algorithms, some based on ECC and Diffie-Hellman (DH). They wanted to create a PQC algorithm similar to what they previously had as Elliptic Curve Diffie-Hellman (ECDH) so they established Supersingular Isogeny-Based Diffie-Hellman (SIDH). To produce private keys, SIDH uses isogenies. As described in 2.3, the isogenies ϕ_A and ϕ_B are identical to the isogenies ψ_A and ψ_B, but started from different curves. E as an elliptic curve.

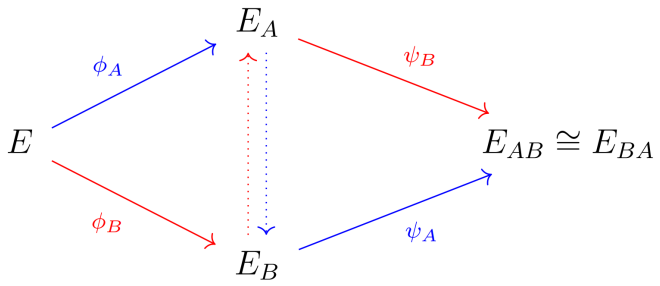


FIGURE 2.3. Alice and Bob's private keys according to [23]

Alice's private key will be the isogeny ϕ_A and Bob's private key, the isogeny ϕ_B. Alice's private key is capable of calculating ψ_A and the same for Bob in order to obtain ψ_B [23]. As for public keys, it is necessary introduce new concepts. Attached to each elliptic curve is a number denominated as j-invariant.

$$j(E) = \frac{6912a^3}{4a^3 + 27b^2} \tag{2.5}$$

This number is used to determine if two elliptic curves are isomorphic [23]. If two elliptic curves possess the same j-invariant, then they're considered isomorphic. The public key it derived from the elliptic curve that results from the secret. Since the j-invariant uniquely identifies isomorphic elliptic curves, so the j-invariant efficiently becomes the public key. Finally to form a pair of keys, it is mandatory to randomly selecting isogenies as private keys, we can achieve that by using the notion of kernel of an isogeny. The kernel of an

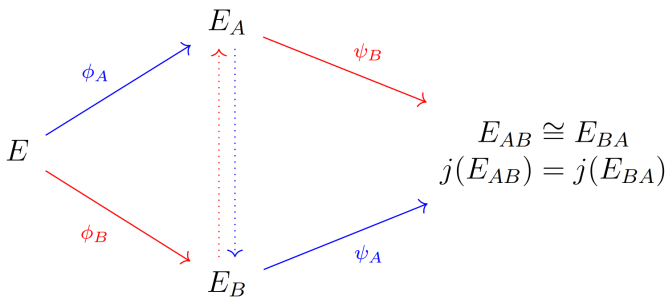


FIGURE 2.4. Alice and Bob's shared secret on [23]

isogeny is the set of points that map to the identity of a certain elliptic curve. And the public key is derived by computing the image curve resulting from this secret isogeny [23].

TABLE 2.1. Each PQC scheme strengths and weaknesses according to [19] and [20]

Scheme	Advantages	Disadvantages
Lattice-based	High security, fast and convinient	Vulnerable implementation on average computers
Code-based	High security	Extra data required to verify digital signatures
Isogeny-based	Public keys and digital signatures with small sizes	Mathematically complex
Hash-based	Simple algorithm and fast execution	Security depends on hash function

Advantages about isogeny cryptography, it requires less computational power than others schemes, since it produces smaller keys, which reduces bandwidth needs, making it more efficient overall as shown in table 2.1.

After reviewing each PQC, we can conclude certain advantages and disadvantages. We decided to approach these specific schemes because it were NIST's choice after 8 years of researching. While lattice-based schemes dominates NIST with multiple of schemes such as Kyber (ML-KEM), Dilithium (ML-DSA), Falcon (FN-DSA) there is an hash-based scheme with very much potential, $SPHINCS^+$ (SLH-DSA). Next section we will approach each NIST PQC standards.

2.1.3. NIST Standard PQC algorithms

This section is dedicated to NIST PQC schemes according to Federal Information Processing Standards (FIPS) 203, 204, 205 and soon to come 206, how they function and their type of PQC.

2.1.3.1. *CRYSTALS-Kyber*

CRYSTALS-Kyber is a Key Management Systems (KEM), whose security is based on the hardness of solving the Learning With Errors (LWE) and CVP problems. Unlike RSA, Kyber does not directly encrypt messages. Instead, it uses a KEM approach, which facilitates secure key exchange between two parties. The shared secret key obtained through this process can then be used to efficiently encrypt data using a symmetric encryption scheme [17]. Before we deepen on Kyber, we must understand the concept of KEM. KEM is operated to send a symmetric key between two parties using asymmetric algorithms. It consists on the sender encapsulating the symmetric key within a cipher-text utilizing the receiver's public key. Once received the cipher-text, the receiver decapsulates and recovers the symmetric key using their private key guaranteeing an authenticated exchange without directly sharing the symmetric key during transmission.

In this public key system, we need to establish a integer *modulus* q and a polynomial *modulus* $f(x)$ to keep the numbers small and polynomials manageable, thereby defining a ring structure. There are three main operations in Kyber: Key Generation, Encryption and Decryption. Starting with Key Generation, polynomials with small coefficients incorporate the private key s. By randomizing the small coefficients, we generate effortlessly the private key. Meanwhile the public key consists of a matrix of random polynomials A and a vector of polynomials t [24]. In order to determine t we need an extra error vector e. Vector e is composed of polynomials with small coefficients, similar to the private key. Now we can calculate t as observed in 2.6.

$$t = A \cdot s + e \tag{2.6}$$

For an attacker to obtain s from (A, t) would require solving Module Learning with Errors (MLWE) and it is very complex even for a quantum computer.

In Encryption, when the party possesses a public key is capable of encrypting the message. On the other hand, decryption can only be achieved by parties who have the private key. The process of encryption utilizes two polynomial vectors error e_1 and e_2 and one vector whose random r. Note that all the polynomials within e_1, e_2 and r have the same characteristics as the polynomials in s. The message needs to be turned into a polynomial and by converting every bit of binary's message into a coefficient. Next, the upscaling of the message by multiplying with $\lceil \frac{q}{2} \rceil$, this is necessary because polynomial coefficients are require to be large. After obtaining the encrypted message m, we proceed to determine Kyber ciphertexts u and v with 2.7 and 2.8 respectively.

$$u = A^T \cdot r + e_1 \tag{2.7}$$

$$v = t^T \cdot r + e_2 + m \tag{2.8}$$

Finally, Decryption step starts by recovering the message with noise m_n. So after computing m_n, we proceed to remove the noise created during transmission. After following

2.8, we can recover the original message by simply converting the coefficients into binary. Coefficients whose values are closer to $[\frac{q}{2}] = 9$, are rounded to 9, while if they are closer to 0 are rounded to 0. By doing that, simply multiple that converted message to $\frac{1}{9}$ and the binary of the original message is successfully recovered [24].

$$m_n = v - s^T \cdot u \tag{2.9}$$

$$m_n = e^T \cdot r + e_2 + m + s^T \cdot e_1 \tag{2.10}$$

2.1.3.2. *CRYSTALS-Dilithium*

Dilithium also a member of the CRYSTALS family since it's a lattice-based digital signature scheme. Its difficulty on solving relies on MLWE and Module Short Integer Solution (MSIS). Unlike Kyber, Dilithium focuses on applications of authentication to ensure authenticity and integrity. Dilithium utilizes Number Theoric Transform (NTT) for faster multiplication with polynomials, crucial when dealing with lattices schemes. It assists on the optimization of the CPU usage which can be useful when handling with sensitive environments (e.g. Internet of Things (IoT)). To minimize the size of the signature, Dilithium uses a rejection technique, where the signature needs to meet specific requirements for the purpose of being valid.

Starting with the signature, first the key must be generated randomly, using a vector y with small coefficients. Assuming that A is the MLWE array and m is the message we intend to sign and s is the MLWE private key. In order to proceed, the signer must verify that the z coefficients are bellow a certain threshold, if not, a whole new y is created again [25].

$$w = A \cdot y \tag{2.11}$$

$$c = HASH(m||w) \tag{2.12}$$

In 2.13, b is considered the MLWE public key. To verify if the signature is valid, two conditions must be true, $c = HASH(m||w')$ shortly $w = w'$ and the coefficients of z must be under the threshold mentioned before.

$$b = A \cdot s + e \tag{2.13}$$

$$z = y + c \cdot s \tag{2.14}$$

$$w' = A \cdot z - c \cdot b \tag{2.15}$$

The signer knows s, which permits him to generate various z with small coefficients worth of being potential valid contenders.

2.1.3.3. *FALCON*

Along with the CRYSTALS family, FALCON also deals with Lattice Reduction and is also a digital signature algorithm, unfortunally it proofs to be slower than Dilithium due to its low bandwidth and fast verification according to NIST. So it becomes a popular choice if it needs to be implemented into a protocol. FALCON uses Nth-degree Truncated

polyminal Ring Units (NTRU) as an assymetric encryption method, benchmarks proof that it is twice as fast than RSA to encrypt and even more efficient decrypting. NTRU is a public key method not based on factorization, discrete logs or elliptic curves [26].

Three parameters are selected N, p and q. To generate a key pair, we also require two polynomials f and g. Assuming that f and f_q are our private keys. While h is the public key.

$$f_q = f^{-1} \pmod{q} \tag{2.16}$$

The two parties agree to share N (limiting the largest polynomial power), p and q. So one of the parties, generates two short f and g (as mentioned previously) and with the equations 2.16 and 2.17 we obtain our key pair.

$$h = p \cdot f_q \cdot f \pmod{q} \tag{2.17}$$

2.1.3.4. *SPHINCS*$^+$

SPHINCS$^+$ is a hash-based PQC stateless algorithm for digital signatures. The biggest advantage about hash based algorithms is not existing good attacks against it yet. The best so far was Brassard–Høyer–Tapp (BHT) algorithm which utilized Grover's algorithm (2.1.1) in order to decrease the computational effort to find a collision from $2^{\frac{n}{2}}$ to $2^{\frac{n}{3}}$ [27]. Its framework includes two kinds of hash functions: Tweakable hash functions and Pseudorandom Function (PRF). Tweakable hash functions, F and H:

$$F : \{0,1\}^n \times \{0,1\}^{256} \times \{0,1\}^n \to \{0,1\}^n, \qquad F \stackrel{\text{def}}{=} T_{h,1};$$

$$H : \{0,1\}^n \times \{0,1\}^{256} \times \{0,1\}^{2n} \to \{0,1\}^n, \qquad H \stackrel{\text{def}}{=} T_{h,2}.$$

We have PRF for pseudorandom key generation, PRF_{msg} to generate randomness for the message compression and additionally H_{msg} which is responsible of the compression of the message to be signed that can process various length messages:

$$\text{PRF} : \{0,1\}^n \times \{0,1\}^{256} \to \{0,1\}^n.$$

$$\text{PRF}_{\text{msg}} : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^n,$$

$$H_{\text{msg}} : \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^n \times \{0,1\}^* \to \{0,1\}^m,$$

SPHINCS$^+$ is composed by a hypertree of keys signed by: Forest of Random Subsets (FORS), Witernitz One Time Signatures (WOTS)$^+$.

WOTS$^+$ is utilized for the inner trees since they will only be used once to sign the root of the next tree. This algorithm is a one-time signature scheme that requires a private key to sign one message, if used multiple times, its security reduces significantly. It relies on the concept of "hash chains" where the hash function is implemented various times and the function encodes the signature value of many bits [28]. WOTS$^+$ possesses two parameters: n and w. Parameter n is responsible for the security, being the length of

the message and also the private key, public key and signature in bits. Parameter w is known as the Winternitz parameter (it is constricted to 4, 16 and 256), it is used to make a trade-off between signing time and signature size: a higher value equals a smaller and slower signature [27], [28]. Defining $len_1 = \lceil \frac{n}{\log(w)} \rceil$ and $len_2 = \lfloor \frac{\log(len_1(w-1))}{\log(w)} \rfloor + 1$. The sum of the lengths represents the number of n-bit value in an uncompressed WOTS$^+$ private key, public key and signature [27].

The WOTS$^+$ key pair derives from a secret seed $SK.seed$ that is part of SPHINCS$^+$ private key, and the address of the WOTS$^+$ key pair within the hypertree while operating PRF [27], [28]. The public key is obtained through implementing the F function (responsible of creating an hash block) repeatedly to each value of the private key, forming multiple hash chains. Finally in order to compact the public key, SPHINCS$^+$ utilizes function T_h (adjustable hash function) instead of Merkle trees according to older versions of WOTS$^+$.

As for the signature and verification the message is interpreted as len_1 integers m_i between 0 and $w-1$. We perform a checksum (necessary to prevent forgery of messages), $C = \sum_{i=1}^{len_1}(w-1-m_i)$ we checked these values until C_{len_2}. Each m_i value defines how many times F is applied on the element that corresponds to the private key, that generates the signature. In order to recover the awaited public key, the verifier applies F until the end of the chain, and then checks if they are a match.

Hypertree is a essential concept to the SPHINCS$^+$, it is an hierarchical structure composed of various trees, where each layer authenticates the one below, enabling stateless signatures. The single tree is capable of signing $2^{h'}$ messages, since the signer derives $2^{h'}$ WOTS$^+$ public keys [27]. Each leaf is a WOTS$^+$ public key and its root is a global public key. To sign one message, we select one of the leaves, use that key to generate a signature and add that as an authentication path (nodes needed for the verifier to reconstruct the tree root and confirm its signature). Tree of trees is a hypertree, created to surpass the limitations of a single tree, since it allows to sign many messages effectively instead of increasing h' value. Each layer is a subtree, so the tree base signs the message, intermediary trees sign the roots of the trees under them and the top (these trees are known as certifiers) tree provides the final public root which is the global public key.

FORS is an improvement of HORST (solves the issue with the public key distribution by making the chunks of the public key leaves of a merkle tree). It is used for its ability to decrease the number of keys needed for a message, since there can be reutilized (few-time signature scheme) [27]. FORS has defined in terms of integers k and $t = 2^a$ used to sign ka bits. To generate a pair of keys, starting with the private key, grouping kt random n-bits together into k sets of t values each. These values are derived from SK.seed while using PRF and the address of the key in the hypertree [27]. For the public key, for each k set we build a binary tree with height, $a = \log_2 t$. Each private value is a tree. The nodes are combined utilizing the H function previously mentioned, using FORS key address in

the hypertree as a parameter. The roots of that k are compressed with Th_k to generate FORS public key (one single n-bit value) [27], [28].

The signature process ocurrs for one message of ka bits, it divides itself in k blocks of a bits. Each block indicates one leaf in each k trees. That signature includes: those leaves and the authentication path. The verifiers then reconstruct each root utilizing the authentication path. The roots are compressed with Th_k to reconstruct the awaited FORS public key. This signature is not verified, instead its value is an input for a WOTS$^+$ signature, and only then it is checked [27], [28].

Now we can fully understand how SPHINCS$^+$ really works, since all the concepts that complement it, are covered. SPHINCS$^+$ public key consists of two n-bit values: the root node of the top tree in the hypertree and a random public seed $PK.seed$. Like the public key, the private key also derive from two n-bit random seeds: $SK.seed$ to generate the WOTS$^+$ and FORS secret keys and $SK.prf$ to use for the randomized message digest [27]. As for the signature process is composed by one FORS signature to the message digest, next a WOTS$^+$ signature to FORS public key and then the authentication path and other WOTS$^+$ signatures we are able to reach the top of the hypertree where we achieve the public root [27].

Finally we must address the processes of message digest and leaf selection, processes which differ between SPHINCS and SPHINCS$^+$. The creation of the randomizer R pseudorandomly based on the message and SK.prf. We can add additional randomness with `OptRand`, so R= PRF(SK.prf, `OptRand`, M). Deriving from the index of leaf node, R (part of the signature) we can obtain the message digest with: $(MD||idx) = H_{msg}(R, PK.seed, PK.root, M)$. SPHINCS$^+$ is capable of verying the index, that way preventing an attacker from selecting a random index and combining it with a message of their choice [27].

2.2. Automotive

2.2.1. V2X

V2X covers multiple types of communication such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N), Vehicle-to-Device (V2D) and Vehicle-to-Pedestrian (V2P). V2X is an essential technology for full autonomous driving, boosting the vehicle's situational awareness beyond the capabilities from sensors, camera and Light Detection and Ranging (LIDAR). The advantage of using V2X instead of these day-to-day sensors, is V2X capablility of detecting objects without needing a direct line of sight with them [29]. Infrastructure developers are currently utilizing two kinds of V2X wireless communication technologies: Dedicated Short-Range Communication (DSRC) and Cellular-Vechile-to-Everything (C-V2X). DSRC is an older technology that operates within the 5.9GHz spectrum for Wi-Fi communication. As for C-V2X, a recent technology that leverages cellular networks, including 5G.

DSRC was based on IEE 802.11p standard and built specifically for low-latency communication (designed for V2V and V2I) suitable for safety-critical applications like emergency braking, generally within range of 300 to 1000 meters. Since it is a mature technology it possesses defined protocols which were in development for decades, built-in security and privacy mechanisms (specifically for vehicular communication), providing strong protections with the slightest computational overhead, making vehicular systems efficient [29]. As a consequence of DSRC being based on open standards, further increases its interoperability which provides communication with different vehicle manufacturerss and infrastructure providers without constraints. Despite all these advantages, there is still some notable limitations, starting with the range in non-line-of-sight on sites like terrains with obstacles can weaken the signal making it less realible. Not only that but, the allocated spectrum dedicated for DSRC has been underused, leading various discussions about reallocating some of the spectrum for more productive uses, possibily influencing the future of DSRC. Another concern is about scalability, since various vehicles are trying to communicate with each other, and DSRC operates in a fixed bandwidth spectrum, so it eventually results in the performance degradation [29].

C-V2X was based on third Generation Partnership Project (3GPP) and 5G cellular standards. It presents with two types of communications: direct short-range communication (PC5 interface) and long-range cellular communications (Uu interface). PC5 interface handles communication similarly to DSRC for V2V and V2I, permitting vehicles to communicate with each without needing cellular networks. Uu interface uses existing cellular network infrastructures, allowing vehicles to transmit information with cloud-based services for real time updates [29]. By utilizing existing cellular networks, it is capable of covering more area than DSRC which is advantegeous for V2N applications like dynamic traffic management and OTA software updates. Unlike DSRC, Uu interface has better performance in non-linear-of-sight conditions thanks to its advanced modulation and coding schemes, making it more reliable in urban environments. While operating on cellular networks, it supports a growing number of connected vehicles without causing any spectrum congestion, therefore its scalability is superior. Finally its compatibility with 5G enables Ultra-Reliable Low-Latency Communication (URLLC) and massive Machine-type Communication (mMTC), which can be a huge step on V2X scenarios. The major disadvantage is network dependency, as mentioned C-V2X relies on existing cellular networks, which may be sparce in rural and unwarranted regions, limiting its effectiveness in those areas [29]. By comparing these two communication technologies, we can determine which is the best for a specific scenario, DSRC is strong and robust for local V2X applications, while C-V2X seems a great potential for the future of connected transportation.

2.2.2. Attacks

As observed V2X may be vulnerable to an broad type of network attacks. According to [30], there are five main types of attacks: Attacks on Infrastructure, Attacks on Software,

Hardware and Attacks on Privacy, Attacks on Data Trust and Attacks based on Behavioral Patterns. Attacking the Infrastructure presents a severe threat to the V2X communication since it is necessary in order to the network to functionate correctly. For each type we mention a specific attack and how can we mitigate it:

Attacks on Infrastructure

- ❏ DDoS attacks –> Denial of service attacks consists on an entity known as the coordinator, responsible by a big number of bots that behave like an hivemind. The main objective of this attack is to flood the victim's system, leaving the system completetly inoperative. The use of strong authentication and digital signatures mitigates the effects caused by the attack [30].

- ❏ Hardware Tampering –> Attackers manipulate the automotive hardware. By doing this the attacker can easily access the veicle's data, specifically network and particular data. A mitigation strategy includes implementing a Trusted Platform Module (TPM), a piece of hardware, that necessitates a software infrastructure capable of protecting and storing data in shielded locations. It also includes cryptographic mechanics such as SHA1 engine, an RSA engine and a random number generator [30] [31].

- ❏ Masquerade Attack –> The cybercriminal passes by an authorized identity, its objetive is to create a Blackhole (routing packets with the intention of losing them) or forge fake data packets that seem to be gotten from legitimate sources. Other possibility is the attacker masquerade as a emergency vehicle and force other vehicles to change directions or even slow down their speeds.

Attacks on Software

- ❏ Spoofing –> Giving the wrong position information to various vehicles present in the current network which needs correct information, even the smallest error could generate an harmful incident. In [32], the authors explored this attack on real-life traffic scenarios and one of the conclusions was that the drive distance under the influence of spoofing increased. The use of certificates can assist on detecting fake vehicles.

- ❏ Man-in-the-Middle (MiM) –> By inserting itself between two vehicles communicating, the attacker can then extract and manipulate the data. MiM can be avoided by heavy authentication process between the two parties before communicating.

- ❏ Brute force –> The possibility of compromising the privacy of V2X communication and authentication strategies by executing this attack may not be very

likely due to resource limitation and network instability but we must prevent any breaches to the system. By reducing the period of availability to the attacker to execute this scripts and reenforcing the security of secrets, attackers will find it harder to breach into the network and eventually give up.

Attacks on Privacy

- ❏ Location tracking –> Attacker tracks victim's vehicle route and position during a certain period of time. It is very important that the keys are constantly being changed and never using permanent keys, avoiding the share of sensitive information about location.

Attacks on Data Trust

- ❏ Hidden vehicle –> The attacker's objective is to provoke accidents and it achieves that by utilizing hidden vehicles to generate false location alerts.

Attacks based on Behavioral Patterns

These attacks intent to influence users based on their behaviour in order to make them launch the attacks. There are two types: selfish and malicious. The selfish type causes the nodes to act selfishly, not sending packets on purpose for example. As for the malicious attacks, the attackers carry out malicious actions in the system.

CHAPTER 3

Related Work

This chapter focuses on analyzing previous research on the integration of PQC into automotive communication systems. Every segment will address aspects like adoption V2X and PQC into automotive environments, including secures updates, authentication and hardware-based security mechanisms. Each section will follow the same structure to highlight the fundamentals:

(1) Context and Objective

(2) Methodology

(3) Results

(4) Final considerations

The final section of this chapter will be a summary of other worthy mentions about this topic, complemented with a table with all works that were used to write this section.

3.1. Post-Quantum Secure Architeture for Automotive HSM (Wang et al., 2020)

The evolution of automotive technology lead to an increase of requirements on security incoporations functionalities in in-vehicle architeture, one way to achieve it, is by adding Hardware Secure Modules (HSM) in ECU. So that way, we can enforce secure communications within the HSM by possessing cryptographic computations using embedded hardware accelarators. Unfortunally there is not any standard for architeture design for an automotive HSM. This design is desired since it will protect agaisnt future attacks by attackers exploiting large-scale quantum computers. The authors objective with this paper is to analyze the NIST final candidates in PQC standardization process and to propose a new sets of hardware accelarators for the future generation of the automotive HSMs.

The authors needed to choose specific PQC schemes according to their level of security, so they separated into two parts: medium security level and high security level. For the medium level, they selected CRYSTALS Dilithium and BIKE which is a code-based KEM scheme. They were chosen due to their small key sizes and efficiency. BIKE is based on Quasi-Cyclic Moderate Density Parity-Check (QC-MDPC) codes, NIST affirmed that BIKE is very promising scheme compared to others like Hamming Quasi-Cyclic (HQC), all due to its small sizes of the public key and ciphertext, also the bandwidth it is better. For

high security level, were chosen XMSS which is a stateful hash-based signature scheme standardized by IETF and CRYSTALS-Kyber. XMSS relies on the underlying hash function which can be constructed even with the presence of large quantum computers [33]. Both of these schemes possess small keys, as their main operations can accelarated by using dedicated hardware accelarators. In addition, we can combined the two for applications aiming a high security level [33]. In order to advance into determining which hardware accelarators are a right fit for each scheme, we must perform a software profiling. For this process, the authors used a tool designed to analyze the operations that require more computational power, *Gprof*. Starting with XMSS, most of the time in the verification operation is consumed by the underlying hash function [33], so SHA2-512 or SHAKE-256 can be used as the hash function. Next, polynomial multiplication is the most expensive operation there is in BIKE, so the hardware involves the instantiation of AES-256 and the hash function SHA-384. The results for other two schemes, Dilithium and Kyber, were similar due to their protocols being based of the same building blocks [33]. Dilithium uses AES-256 while Kyber uses SHAKE. With this process done, we can proceed to idealize different HSM architetures.

Next step, we focus on the architetures, the authors proposed two, one modern (without PQC support) and the other one dedicated to PQC. We observe the different blocks used for each HSM. Both share blocks like: SHA2-256, AES-128 (PQC HSM uses advanced versions), TRNG, PRNG. SHA2-256 used as a general-purpose hash function, AES-128 includes cryptographic operations like key generation, encryption and decryption. TRNG is used as a true number generator thus the abbreviation and PRNG as a pseudo random number generator, generally is seeded by TRNG and expands the randomness by using AES engine [33]. The Modern also utilizes ECC-256 for 256-bit elliptic curve arithmetics in an asymmetric crypotosystem. There are two kinds of PQC HSM, as mentioned before the sepparation of the schemes according to their level of security. Each scheme has a specific block in order to accelarate it. Starting by the medium level of security, Dilithium-Poly-Mult, a polynomial multiplier based on NTT supporting the security parameters of Dilithium, AES-256 expands randomness in AES-based implementation of Dilithium (PRNG and Dilithium can share the same AES-256 core). For BIKE, BIKE-Poly-Mult, polynomial multiplier used for accelerating both dense and sparse polynomial operations [33]. The high level security, Kyber-Poly-Mult similar to Dilithium, is a dedicated NTT based polynomial multiplier that accelarates Kyber. SHAKE-256 is chosen for both XMSS and Kyber aiming high security level can be constructed with SHAKE-256.

Finally we reached the results, after comparing the three HSMs: Modern HSM, PQC PQC Medium Security Level and High Security Level. The Modern HSM and the Medium Security achieve similar areas of utilization, meanwhile the High Security demonstrated a small overhead in usage area, about 13%. In terms of memory, both PQC approachs are lightweight since ECC-256 (intensive asymmetric cryptographic core) is eliminated from

the architeture. The authors did not acknowledge the extra overhead generally common in embedded systems due to interface communication logic, memory controller logic, top-level controller logic, etc. This paper intends to present the feasibility of designing a time-area efficient HSM embedded on vehicles and by the results we can conclude that is in fact feasible, without bearing big area overhead [33].

This paper has a creative approach, introducing an embedded hardware into an automotive system. It is relevant since it obtained great results, the PQC architetures proved to be lightweight and secure than the Modern HSM architeture. The authors mentioned that in the future it would be interesting to explore real hardware implementations for these architetures and eventually have a prototype for analyzing the performance metric, essential to obtaining concrete results about the efficiency of these architetures [33].

3.2. Post-Quantum Secure Over-the-Air Update of Automotive Systems (W.Bos Joppe et al., 2023)

There are various ways to expose applications to dangerous risks due to the usage of pre-quantum cryptographic algorithms as mentioned previously, this work intends to study more in depth, ways to ensure that applications like secure boot and secure OTA update are not easily manipulated through software or firmware modification. The main focus is investigating the impact of Dilithium on OTA, working with a vehicle network processor, *S32G* and comparing PQC signature verification in two different settings: The first is performed by Arm Cortex-A53 (knowned for being bigger and faster) based microprocessor and the other by microcontroller which is part of the Hardware Security Engine (HSE) based Arm Cortex-M7 (offers a variety of security features unlike A53).

In order to implement OTA updates the authors take advantage of AUTOSAR Adaptive Platform which offers the "Update and Configuration Management" (UCM) cluster, responsible for guaranteeing that the update process goes accordingly by following the next steps: Download the update package by UCM client, verifying the integrity of the package, evaluate its manifest, interact with state management and implementing the update. The authors consider the verifcation step as the most vulnerable step, this because potential attackers can infiltrate into the infrastructure through various ways e.g attempting to obtain the digital signature over a malicious package. Other attacks work by using power spikes, clock glitches, temperatures attacks, optical attacks or eletromagnetic injection to induce faults during the execution [34]. As for the implmentation, cryptographic algorithms can be applied in ECUs in two ways: HSE or in a software library executed on a host application core. The HSE method offers protection mechanisms against physical attacks but less flexible when it comes to execute software on the host processor. CryptoAPI is utilized to implement the digital signature verification. The main problems that main occur while sending OTA updates have to do with performance issues due to the quantity of data that is transmitted through 4G LTE to 5G. So a small performance loss

when migrating to PQC era is not expected to impact on the user side because it will be very minimal in update time [34].

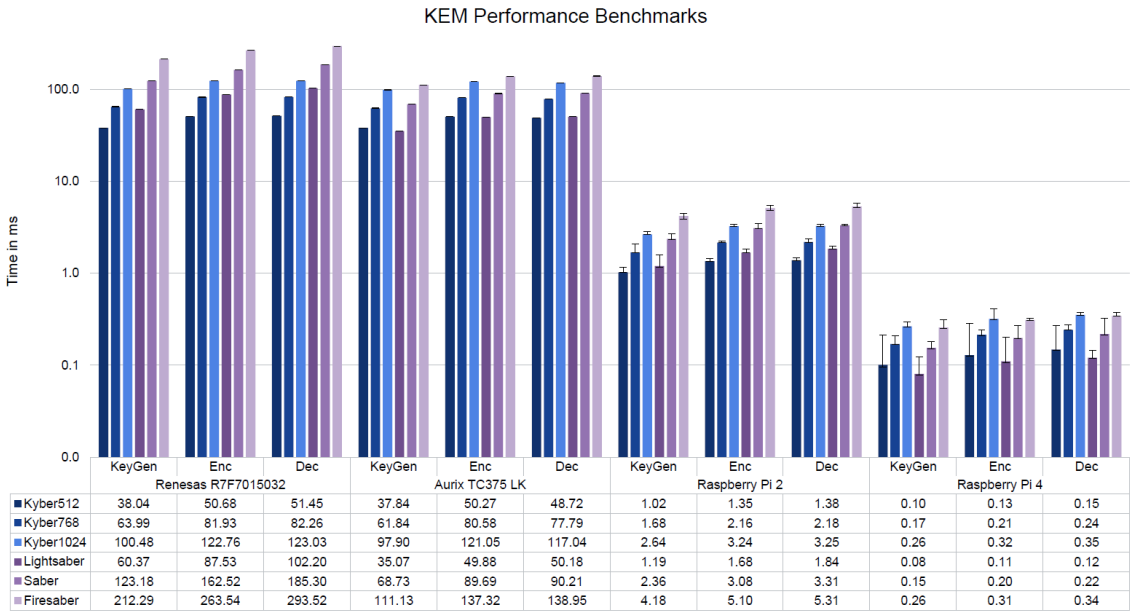
The authors explain in the Results section that despite the PQC migration not interfere much on the overall performance, is still important to measure the latency of the verification algorithm. Note that latency is not only determined by the choice of signature scheme, it is also by the length of the software update. The OTA update protocol was adjusted in order to utilize PQC Dilithium algorithm, in this especific experiment the authors chose Dilithium-2 parameter set: public key of 1312 bytes and veryfing a signature of 2420 bytes. One side, host processor was running the verification and the Dilithium verification on the HSE. The benchmark conclude that Dilithium-2 verification took 0.5 ms on the host processor and 11.1 ms on the HSE. This difference can be explained by the fact that the host processor and the HSE differ when it comes to performance (different clock speeds and instruction sets). Experimenting with diverse microprocessors Cortex, we clearly observe an alarming difference sometimes even 5-6 times. The host may be a desirable choice but do not forget that it is unprotected against advanced attacks relevant in the automotive domain, while HSE carries the protection (time-memory trade-offs), so that explains the slowness that presents performance wise [34].

3.3. Impacts of PQC on Automotive Security (Rathweg et al., 2024)

With the rise of large-scale quantum computers, classical asymmetric cryptography will not be powerful enough to protect our data. So we should start to consider the quantum-secure transtion. So the authors study the impact of this possible transtion to PQC schemes by analyzing them in protocols and functions like KEM, Secure Boot, Secure Diagnostic Access, Transport Layer Security. The authors selected four PQC algorithms from NIST third-round finalists: CRYSTALS-Kyber and Saber as KEMs and CRYSTALS-Dilithium and FALCON as digital signature schemes. They tested these algorithms on different hardware targets: Aurix TC375, Renesas R7F7015032, Raspberry Pi 2 and 4. Testing on this kind of hardware aims to represent different classes of ECU in today's vehicles [35]. To compare the algorithms we need focus on specific aspects: Execution time (ms), Stack Consumption (bytes), Code Size (bytes). The authors utilized PQClean, which is a collection of integratable clean implementations of post-quantum schemes. By using an API, the authors can call the PQC functions with short abbreviations:

- ❏ KeyGen() –> Key Generation
- ❏ Enc() –> Key Encapsulation
- ❏ Dec() –> Key Decapsulation
- ❏ Sign() –> Signing
- ❏ Verify() –> Verification

We observe that CRYSTALS-Kyber has a better performance overall in all different security levels compared to Saber.



	Renesas R7F7015032			Aurix TC375 LK			Raspberry Pi 2			Raspberry Pi 4		
	KeyGen	Enc	Dec	KeyGen	Enc	Dec	KeyGen	Enc	Dec	KeyGen	Enc	Dec
Kyber512	38.04	50.68	51.45	37.84	50.27	48.72	1.02	1.35	1.38	0.10	0.13	0.15
Kyber768	63.99	81.93	82.26	61.84	80.58	77.79	1.68	2.16	2.18	0.17	0.21	0.24
Kyber1024	100.48	122.76	123.03	97.90	121.05	117.04	2.64	3.24	3.25	0.26	0.32	0.35
Lightsaber	60.37	87.53	102.20	35.07	49.88	50.18	1.19	1.68	1.84	0.08	0.11	0.12
Saber	123.18	162.52	185.30	68.73	89.69	90.21	2.36	3.08	3.31	0.15	0.20	0.22
Firesaber	212.29	263.54	293.52	111.13	137.32	138.95	4.18	5.10	5.31	0.26	0.31	0.34

FIGURE 3.1. Performance of the different security levels of KEM schemes demonstrated in [35]



	Renesas R7F7015032			Aurix TC375 LK			Raspberry Pi 2			Raspberry Pi 4		
	KeyGen	Sign	Verify	KeyGen	Sign	Verify	KeyGen	Sign	Verify	KeyGen	Sign	Verify
Dilithium2	125.84	360.30	127.62	133.71	420.24	137.53	3.26	9.28	3.37	0.29	1.17	0.31
Dilithium3	333.89	964.97	324.19	236.50	715.33	233.77	5.90	17.08	5.83	0.51	2.26	0.50
Dilithium5	567.31	1,093.92	564.01	401.62	774.43	403.24	9.81	18.91	10.20	0.78	2.27	0.81
FALCON512	15,712.34	3,324.37	49.14	8,504.46	2,338.48	27.47	238.28	62.11	0.92	46.46	13.74	0.16
FALCON1024	27,094.31	7,016.86	95.70	24,087.47	5,121.28	54.01	694.45	132.82	1.75	138.26	29.75	0.33

FIGURE 3.2. Performance of the different security levels of signature schemes observed in [35]

In term of consumption in bytes, we observe that CRYSTALS-Kyber requires up to 1.3 times less memory than Saber at each security level. Signature schemes demonstrate stronger differences in memory consumption. FALCON require less memory than CRYSTALS-Dilithium [35].

In conclusion, as we know quantum does not impose any threats for automotive security currently. So the transition to quantum secure algorithms is not justified just yet. We analyzed that many ECU support signature verification due to their constrained space, so the original equipment manufacturer does not implement it. According to the results the PQC algorithms depend on multiple factors: one is the consideration of security requirements and time restrictions of the specifics of automotive use cases. The other consideration is the available resources provided by the corresponding ECU, depending on its performance class, limits the possible algorithm choices [35].

3.4. Post-Quantum Authentication for V2V Communications (McCarthy et al., 2024)

The median life of a vehicle is between 12-15 years, so they will operate until 2039 or beyond, but many experts believe there is a 50% or greater chance that quantum computer power will surpass the classical cryptography currently used on vehicles to garantee a secure communication with everything around it. Standars like IEEE 1609.2 AND 1609.2.1 that secure connected vehicles rely on ECC, especifically Elliptic Curve Digital Signature Algorithm (ECDSA) used to sign V2V messages to protect against spoofing, alteration, and replay attacks in both DSRC and C-V2X will be very weak agaisnt quantum power. The V2V system must broadcast ten or more safety messages per second over a shared radio spectrum. To achieve the spectrum requirements, it is necessary to ensure that we minimize the transmission lentgh and message size, in order for others systems who utilize the spectrum can also operate, which means maximizing the system capacity. Due to current use of spectrum to support signature verification in V2V, it is considered impossible to apply PQC algorithms into this network. So the authors propose a scheduling technique for mesage signing certificate transmissions that adaptively reduce the use of radio spectrum. This combination, is the first integration of PQC and V2V which satisfies the constraints given the available spectrum. The authors present their solution for spectrum optimization technique and threat model and *Partially Hybrid* design [36].

First in order to otimize the spectrum, authors investigated the excessive transmission of redudant information under the current security standards. Uitlizing VEINS, the authors tested how many Broadcast Safety Messages (BSM) they could send in a determined interval. So every 100 ms they send a BSM with a full ECDSA certificate in every Secure Protocol Data Unit (SPDU) according to current standards. They observed within a 5 minute period, 100 iterations were made, with a median of redudant certificates received to total certificates receveid across all vehicles turns out to be 99.3%. This result suggests a safer option to maintain the spectrum mainly free. Also leveraging from a protocol defined by IEEE 1609.2, Peer-to-Peer Certification Distribution (P2PCD), because it allows vehicles to request another vehicle's pseudonym certificate from others who already have it, consequently compensating the time from any delays from an increased interval between SPDU's transmissions.

They relied on machine learning for examine for various scenarios with different types of conditions. Random forest classifier is enforced for V2V due to their light weight, minimal storage space requirements, and logarithmic time predictions once developed. Training the model through several VEINS simulation's samples, experimenting with distinct vehicle density, speed, transmission rate, etc. The model includes various variable environment features, and communication channel metrics like signal-to-interference-and-noise ratio (SINR), but the authors noticed that two specific environment parameters had a strong connection to determine how strained the channel is going to be and measuring the interference: vehicle density around the transmitter and distance between the transmitter and neighbour vehicles. Relatively about the threat model within of *Partially Hybrid* design, the authors had to assume certain aspects like, the attacker's capabilities, security of the computations, the quality of certificate's operations (generation and fragmentation), etc. Also assuming that quantum computer cannot break ECDSA immediately. On the other hand the certicate operations, the authors describe an hybrid certificate, C_S which have multiple types, indicated by the field 'version', so they proposed a new 'version', combining an ECDSA certificate with a Post-Quantum certificate over the same ECDSA verification key. Fragmentation is the solution to reassure the size constraints on the frames, the first few SPDUs will include fragments of the sender's hybrid certificate, after receiving all fragments, the receiver can then verify the integrity of the verification key using ECDSA and PQC, before utilizing the ECDSA key to verify signatures [36].

They conducted indoor and outdoor experiments utilizing *PQ-V2Verifier*, a testbed for V2V security, an extension of *V2Verifier*, an open-source testbed that supports transmitting BSMs over the air with SDRs while using DSRC and 1609.2-compliant BSM signing and verification using ECDSA. *PQ-V2Verifier* is composed by various cryptographic libraries for the algorithm implementations, Botan (ECDSA, Dilithium and XMSS), *liboqs*(FALCON, $SPHINCS^{+}$) and certificate verification functionalities according to IEEE 1609.2.1, as well as reworking some signature generation verification schemes in order to accomodate larger volumes of BSMs. As mentioned before, the authors also had the assistance of VEINS, useful for simulations of real-world wireless communication scenarios. The biggest restriction in terms of scalability of the system capacity of a V2V security protocol (v_{max}) is due to the frame duration and signature verification time. Frame duration is very difficult to manage due to the possibility of the BSM overlapping, inevitably blocking multipe vehicle's transmissions. According to the authors, the only PQC scheme able to fullfil standards of *Partially Hybrid* design is FALCON, since $v_{max} \geq 100$, which means, the system needs to support at least 100 transmissions per interval, "a viable PQ algorithm must be capable of verifying at least 100 signatures per 100 ms interval", if the scheme does not allow that it is unfeasible ($v_{max} \leq 100$). Regarding the signature verification time, for a scheme to be feasible it needs to complete at least 1000 verifications per second, and as we can observe the results, FALCON has

$v_{max} = 2243$ and Dilithium has $v_{max} = 5299$ unfortunally Dilithium it is not eligible because it did not pass the frame duration. So FALCON is the most eligible, the authors also observed that it can sign 32 messages per second. Efficiency-wise, it was noticed that the *Partially Hybrid* design added a 0.6 ms of delay to each BSM in both light and dense traffic scenarios, this value is very optimistic since PQC considers the elevated overhead. Spectrum optimization also reduces per-BSM delay to 0.25 ms and 0.39 ms on the same traffic scenarios as previous. Finally we need to analyze if *Partially Hybrid* design causes any kind of BSM losses. They define frame loss ratio (FLR), when just implement the design, FLR increases notably by 0.11 (+28.9%) and 0.26 (+61.9%) in both scenarios. Meanwhile if we implement the design plus the spectrum optimization, FLR increases vs.ECDSA by just 0.03% (+7.8% and +7.1%) in the same scenarios. Further experiments in a real environment using two vehicles, placing an Universal Software Radio Peripheral (USRP) and a lapton on each. After configuring the USRP to transmit BSMs at 10 Hz rate and driving both vehicles on a circuit around their campus, maintaining 15-20 meters spacing between them and average speed of 32 km/h. Reviewing the logs, we observe that *Partially Hybrid* deisgn performed well with only a FLR of 7.4% comparing with 1.2% for ECDSA [36].

Overall this article presents with a possible deisgn that probably will be reused and improved in future works. Improve in a sense of making the others PQC algorithms be feasible according to V2V standards. Like the authors, we believe that more theoretical and experimental testing (including CAs and their respectively infrastructures in order to secure V2V against quantum threat) will eventually envolve into something useful for everyday real life scenarios.

3.5. Other Relevant Works

There are yet many papers exploring this vast topic, many evaluating the cost of certain methods, others implementing AI into their work to assist on decisive moments and to apply them into real life scenarios. In this section we approach some papers worthy of mentioning due to their impact on this new area of research.

In [37] the authors focus on solving a big problem of lattice-based PQC that is Error-Correcting Codes (ECC), it helps to increase security and decrease the failure rate. But unfortunally ECC are vulnerable to timing attacks, so the authors present possible solutions to avoid these malicious attacks. ThreeBears is a lattice-based KEM based on a variant of the Integer MLWE problem, it generates three instances of the Bear family: MamaBear, PapaBear and BabyBear. Authors chose BabyBear due to its efficiency and key sizes. It is one of few PQC algorithms that includes ECC, so the authors chose this one to solely focus on strengthening its security. Bose-Chaudhuri-Hocquenghem (BCH) codes are a classic ECC codes choice, known to improve the performance, capable of correcting multiple erros and flexible when it comes to length and code rate. We observe that BCH codes could increase both lattice security (2^{141} –> 2^{152}) and decrease

failure probability (2^{-54} –> 2^{-58}). The anti-time attack implementation was dependent on certain aspects of the decoding process: decoder implementation, number of errors and the received codeword, so the following decoding algorithms (included on BCH decoding process) needed protection: Berlekamp-Massey, Chien-Search and Fix-Error. It was necessary to apply some modifications to these algorithms to make them comply with certain requirements: avoid conditional branch operations which depends on the syndrome or on the number of errors in the codeword, unroll loops where the number of iterations depend on the secret data and to use constant-time binary operations such as & instead of &&. For Berlekamp-Massey the syndrome dependent branch was completetly removed also the maximum number of correctable errors is now 6. In Chien-Search the branch to different solvers and their execution was removed due to its exploitability by timing attacks, the algorithm was modified to behave as in the maximum number of possible roots is 4. And Fix-Error also changed its behaviour to act as the maximum number of errors, iterating as many times as if t = 6. After these alterations, the results present that the protection methods integration increase the runtime of the BCH-decoder, when none errors are produced the execution time of the protected ECC is 151.12 times slower than the unmodified ECC. On the other hand when 6 errors are produced the protected ECC is 7.66 times slower than normal ECC. Proceding to the hardware implementation, TC297TF Application Kit from Infineon was the chosen one for the experiment, equipped with AURIX. Concluded by the authors that BabyBear achieves an overall clock reduction of 55.98% compared to the baseline. Now concerning security, BabyBear paired with a strengthen ECC can achieve lower failure rates by a good amount, the downside is the time required to protect from these errors, as mentioned before the execution time can be 7 times slower than the baseline ECC with also a small performance overhead.

As mentioned before on [34] regarding the Automotive OTA updates, now it is time to analyze the costs of such thing. The authors in [38] explored multiple PQC algorithms and tested them on an automotive-oriented evaluation board, Xilinx Zynq Ultrascale+ ZCU102. Update and Configuration Manager (UCM) is the responsible ECU for cryptographic operations concerning updates. The system model proposed starts with the OEM (possesses a Signature Key (SK) known as the private key), it distributes these keys to the cloud and consequently it delivers to various edge nodes existing in scenario (smart city). These edges nodes give these keys to the end-user, vehicles. Vehicles in turn, have their own public key called Verification Key (VK) inside of the UCM. That way, the UCM can verify the OEM's signature over the update using the VK. It is also presented a Threat Model with algorithms such as Shor's algorithm and Grover's algorithm, exploring divergent attacks like rollback attack and malicious update attack. Only 4 digital signature verification were subjected to performance evaluation, 2 non-quantum resistant (RSA and ECDSA) and 2 quantum resistant (Dilithium and FALCON). Obviously the authors point out the huge difference between pre-quantum and post-quantum, being pre-quantum cannot garantee proctection against the Threat Model. In terms of

execution time of the verification algorithms, pre-quantum achieved 19 ms while post-quantum 22 ms (FALCON) and 28 ms (Dilithium). When the update size increases the verification time of post-quantum algorithms extremely increases while the pre-quantum does not vary much. The reason of this happening is due to the performance of SHA-256 which is hardware-accelerated while post-quantum does not possess an embedded hardware-accelerated hash function. The other interesting trend that authors mention is that the post-quantum algorithm verification is not affected by the security level when it comes to time. FALCON has a constant time conformally the security level, as for Dilithium there is a little raise (about 1 ms for each update size). Between these two, FALCON has a sligth advantage due to its execution time and smaller digital signatures.

We previously discussed types of attacks that can occur when operating a V2X network from DDoS to spoofing, in [39] the authors analyze different standards for various V2X EU projects and explicitly mention security issues like the use of certificates: "Certificate Based Message Verification", "Certificate Usage and Change Policy", "Certificate Reloading" and so on. They also bring up communication modes, Broadcast-Oriented Communication, Unicast explaining how vehicles could leverage from these types of communications. They list all security issues and concluded that the gaps related to security aspects are significant, the specifications that exist are described conflicting and undefined. This could lead to current communications to be unreliable and insecure and unusable. So in 2025, the authors of [40] focused on implementing PQC into Controller Area Network (CAN) protocol with the intetion of securing V2X communications on a low level. The workflow proposed by the authors demonstrates de communication of only two ECUs (to facilitate the assimilation of the concept) and the KeyManagement. The indivual stars by turning on the vehicle and this automatically sends a message to the KeyManagement entity that consequently alerts the first ECU and this one alerts the second one. KeyManagement generates the key (verification key) and then encapsulates it. Transmits the key to the ECUs and they start decapsulating it, storing it. Only then the ECUs can start communcating with each other safely. Results show that it is impossible to apply MiM and eavesdropping attacks with this approach due to the secret key provided by the KeyManagement entity. Overall this integration was very successfull, obtaing great results although with still a bit of delay from the Secure CAN, the model proposed could ensure data confidentially and comply with the requirements outlined in the ISO/SAE 21434 standard.

Concerning real life scenarios, when it comes to transportation, researchers implemented AI in [41] into Transportation Cyber-Physical Systems (TCPS) for adaptive algorithm selection, threat prediction and resource optimization. For example the adaptive algorithm selection basically consists on selection the most adequate cryptographic algorithm for that specific context. A machine learning model could be trained on parameters such message size, required latency, channel conditions to decide whether to use Dilithium or

Falcon for a given signature or to participate on an hybrid RSA+Kyber handshake or a pure Kyber handshake for a key exchange. That way the model could determine which algorithm would be the most appropriated for that moment. The model can also be teached how to deal with certain attack scenarios, when anomaly scores spike in a specific period of time, the AI policy engine automatically locks that segment into a "PQC-only, extended key". While [42] concentrates on strengthening their defenses against AI-assisted physical attacks against a Intelligent Transportation Systems (ITS).

TABLE 3.1. Related Work papers

Paper	Key Focus	Methodology	Advantages	Limitations	Published
[33]	PQC	PQC on HSM	High Security	Performance	2020
[37]	PQC	ECC into PQC	Optimization	Performance	2020
[38]	PQC for Automotive OTA Updates	Implementing PQC Dilithium and Falcon on Hardware	Falcon achieves better results than Dilithium	PQC doesn't have embedded hardware-accelerated hash function	2021
[43]	PQC	Implementing PQC on Hardware	Low Failure Rate	Performance	2021
[39]	V2X	Survey	Comprehensive	Standards	2023
[34]	PQC for Automotive OTA Updates	Implementing PQC Dilithium for Secure OTA Updates	High Security	Performance	2023
[35]	PQC	Implementing PQC on Hardware	Security	Performance	2024
[44]	PQC for V2V	Partially Hybrid Protocol	Reliable and Low-Latency	Slight Frame Loss	2024
[36]	PQC for V2X	Dataset Generation for PQC attacks	Promote further research	Doesn't cover real-world effects (i.e. jitter)	2025
[41]	AI-PQC for V2X in Transportation	Enhancing PQC with AI	Self-Adapting and Intelligence-driven Security Architecture	Little Explored	2025
[40]	PQC for in-vehicle security	Segmented-based CAN model with PQC Kyber	Efficiency and Feasibility	Alternatives like SIDH, BIKE or Saber were not feasible	2025
[45]	V2X Attacks	Simulation of Jamming attacks on NR-V2X	Proved that Jamming attacks degrade communication	Lack of dynamic vehicular interactions	2025
[46]	Key Agreement Protocol for V2X with PQC	Protocol based on PQC Kyber and Saber	Low handshake latency and moderate resource consumption	Experiment conducted in a virtual environment	2025
[30]	V2X Attacks	Survey	Compiled works to facilitate researchers' studies	Developments with machine learning were not included	2025
[42]	PQC for Intelligent Transportation Systems	Review on V2X and vulnerabilities	Explicit priorities for the implementation roadmap	Bandwidth and hardware	2026
[47]	PQC for V2X	Comprehensive Overview of PQC	Compiled state-of-art PQC in Automotive Context	Necessity of lighter and faster PQC schemes	2026
This thesis	PQC for V2X	Dataset of PQC scheme's performances on different environments	Viable for automotive use and promising for future protocols	Simulated on virtual environment	2026

CHAPTER 4

Solution

4.1. Methodology

To achieve meaningful results, it is necessary to define the scenarios and the PQC schemes to be evaluated. In order to analyze different conditions, two distinct scenarios are considered: an urban environment (characterized by high vehicle density) and a rural environment (with low vehicle density), both of which are implemented and available online[1]. This approach aims to assess how PQC algorithms perform under varying levels of network stress. Further details regarding these scenarios are provided in Sections 5.1 and 5.2.

4.2. Tools

This section serves to explaining which tools were used for the experimental part and how were they used.

4.2.1. Hardware

The hardware displayed on table 4.1 and 4.2 were utilized for the experiment. To perform such simulation, it isn't necessary huge computer power for the rural scenario, as for the urban one, we assume that, by the fact of multiple vehicles interacting with each other and with the Roadsite Unit (RSU), it will require more processing. Both hardwares are decent for both scenarios.

TABLE 4.1. Hardware 1 - Components used in the simulation

Operative System	Windows 11 Home 64 bits
CPU	Intel(R) Core(TM) i5-8400 @ 2.80GHz
Memory	16384MB RAM
GPU	NVIDIA GeForce GTX 1060 3GB
Motherboard	HP 84FD

TABLE 4.2. Hardware 2 - Components used in the simulation

Operative System	Windows 11 Home 64 bits
CPU	AMD Ryzen 5 5500U @ 2.10GHz
Memory	15769MB RAM
GPU	AMD Radeon Graphics
Motherboard	LNVNB161216

[1] `https://github.com/MarianaCGoncalves/V2X-PQC-Tese`

For future works, it would be interesting to test in real vehicular hardware. Joppe W.Bos et al in [34], explore the S32 vehicle network processor fabricated by NXP. At the moment, NXP offers PQC-secure hardware: S32N (Vehicle Super-Integration Processors), S32K5 (Automotive General-Purpose Microcontrollers) and i.MX 94/95 (Applications Processor) [48]. Before deepen more into the implementaiton of PQC, we must approach how these processors are capable to secure agaisnt HNDL. One of the main the features of i.MX 94 is EdgeLock Secure Enclave which is a subsystem dedicated to manage security to ensure that the security keys remain safe in case the main application processor is compromised. These processors support NIST latest standardized algorithms (FIPS 203, 204 and 205) which represent ML-KEM (Kyber), ML-DSA (Dilithium) and SLH-DSA (SPHINCS$^+$). To complement this, we must address the overhead of PQC algorithms without degrading performance, which is why NXP implements PQC accelerators, EdgeLock Accelerator (Prime) and SHA-3 Acceleration. EdgeLock Accelerator provides high speed message signing, authentication and encryption. Due to the complexity of pure PQC transition, NXP has employed hybrid solutions by combining traditional cryptography (ECC or RSA) with post-quantum algorithms (ML-KEM). This strategy ensures that the data is protected, if the PQC fails, we have a backup algorithm to garantee that no data is compromised. It depends on what is the purpose of the investigation and scenario of the future works in order to assign a concrete Software Defined Vehicle (SDV) hardware. We have S32K5 which serves as gateway between simple nodes and central computes, as for S32N, it is a processor designed to operate as a central hub and finally i.MX 94/95 is used for smart gateways, connectivity domain controllers and advanced telamatics, things which require high performance computing. To be exact we proceed to analyze the specifications of NXP S32K566. It consists of a junction of microcontrollers based on Arm Cortex-M7 and microprocessors based on Arm Cortex R52. As for Memory, 1.5 MB SRAM for Cortex-M7 with ECC, 1 MB of SRAM connected to the Cortex-R52 with ECC, 32 MB of Code Magnetic RAM for all cores with ECC. Safety-wise, it is classified as ASIL D safety compliance (ISO 26262). The Hardware Secure Engine (HSE)2 provides advanced capabilities and high performance when it comes to PQC operations. Overall it is a optimized solution if we focus on serving as a gateway between simple nodes and central computers. Because it runs up to 800 MHz using a 16nm Fin Field-Effect Transistor (FinFET) process, it is considered power efficient. In comparison with the processors series, S32N, which consists of high-performance "super integration" processors designed for central vehicle controler. S32N7 is composed with 8x Arm Cortex-A78AE cores (up to 1.8 GHz) with split/lock capability (Application Processors), 12x Arm Cortex-R52 cores (up to 1.4 GHz) (Real-Time Processors). Memory-wise 36 MB of low latency on-ship SRAM. It also includes an eIQ Neutron NPU for neural network acceleration, along with RISC-V based accelerators for networking and math intense problems. At last, i.MX 94/95, has up to 6x Arm Cortex-A55 cores (up to 1.8 GHz), 32-bit LPDDR5/LPDDR4X

up to 16 GB. Arm Mali-G310 and NXP-developed Image Signal Processor (ISP) for the graphic component.

4.2.2. OMNeT++

Also known as Objective Modular Network Testbed in C++[2], its an object-oriented simulator of abudant types of network, like V2X. It was developed solely for academic and educational purposes. There multiple frameworks compatible with OMNeT++, like Veins[3], it includes a comprehensive suite of models to recreate vehicler network as real as possible. Road traffic simulation is performed by SUMO[4] which is well-established in traffic engineering. Recently the OMNet++ developers created a new way to setup more easily OMNeT++ with *oppenv*[5]. This avoids dealing with manually installing the dependencies and compatibility issues, it also comes with extensive model library and it is cross-platform. Both simulations will be running on Windows 11 via WSL2. With different hardware, we will obtain a wider range of results, allowing us to analyze which option performs best, since the results will depend on the hardware.

4.2.3. Selected PQC Schemes

For this experiment, we focus exclusively on lattice-based schemes due to their strong performance in current post-quantum standardization efforts. After reviewing the literature, a selection was made among the most well-established candidates. In particular, we selected three widely recognized lattice-based schemes: Kyber, Dilithium and Falcon. These algorithms have been standardized by NIST, making them highly relevant for practical deployments. To enable a comparative analysis, we included two digital signature schemes (Dilithium and Falcon), which exhibit different design trade-offs in terms of performance and signature size. Additionally, we selected a single KEM scheme, Kyber due to its strong security guarantees consistent performance reported across multiple studies [35], [46].

TABLE 4.3. Main characteristics of each scheme approached in this work

Kyber	Dilithium	FALCON
❏ Key Encapsulation Management (KEM) ❏ Learning with Errors (LWE)	❏ Digital Signature ❏ Number Theoretic Transform (NTT) ❏ Rejection Method	❏ Digital Signature (Slower) ❏ Nth-degree Truncated Polynomial Ring Units

[2] https://omnetpp.org/
[3] https://veins.car2x.org/
[4] https://omnetsimulator.com/sumo-omnet/
[5] https://oppenv.omnetpp.org/

4.2.4. PQC Libraries

To implement PQC algorithms, we must include a diverse and detailed library. This section intends to explore multiple libraries, analyze each one, and selecting one that will satisfy the requirements for the experiment. The requirements necessary for the experimental part of this work include: have schemes such as ML-KEM (Kyber), ML-DSA (Dilithium) and Falcon, compatible with the OMNeT++ environment and the usefulness of the library in other works.

4.2.4.1. *PQClean*

PQClean[6] [49] is an open-source and a clean C implmentation of various PQC schemes such as ML-DSA (Dilithium), ML-KEM (Kyber),Falcon, SPHINCS$^+$ and others. It follows a strict quality criteria, no dynamic memory allocations, side-channel resistant and it can be integrated into other libraries, like *liboqs* and support projects such as Open Quantum Safe (OQS) that want to transistion to quantum-resistant cryptography. Beyond that, they can be incorporated into embedded systems, very interesting factor for possible integration of PQC on automotive's hardware. Works such as [34] and [46] use this library and obtain decent results, asserting that ML-DSA and ML-KEM schemes are very optimized.

4.2.4.2. *liboqs*

Like PQClean, libqos is also a C library for quantum-safe cryptographic algorithms. Provides ML-KEM and ML-DSA (schemes other than these two are subject to changes). The highlight of libqos is the schemes use a common API for easily switching between each other. It also includes a test harness and benchmarking routines to compare performance of PQC implementations in a common framework.

4.2.4.3. *OpenSSL*

OpenSSl is a robust Open Source Toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL). It's widely used by software developers to implement secure communication and encryption in various applications such as web servers, VPNs and more. Most of applications make use of OpenSSL. It's been the library with the most active user community, which implies, better support and bug fixes. The release of OpenSSL 3.5 introduced NIST-standardized PQC algorithms into the mainstream library's toolkit, ML-KEM, ML-DSA and SLH-DSA. Because how recent it is, not many works utilize this library.

After evaluating the available options, PQClean was selected as the most suitable library for this experiment. This choice is motivated by its comprehensive collection of post-quantum cryptographic schemes, including implementations of the algorithms selected and standardized by NIST, as well as other candidates. Another important advantage of PQClean is its focus on clean, portable, and well-documented C implementations,

[6] `https://github.com/PQClean/PQClean`

which facilitates integration into simulation environments. To incorporate this library into OMNeT++, only a wrapper layer is required, as PQClean is implemented in C, whereas OMNeT++ is based on C++. This interoperability allows seamless use of the cryptographic primitives within the simulation framework. And not only that, but also multiple works reviewed in section 3 utilized PQClean, for example [34], [46].

Figure 4.1 provides a clearer overview of the architecture of this work, summarizing all the details described in this section.

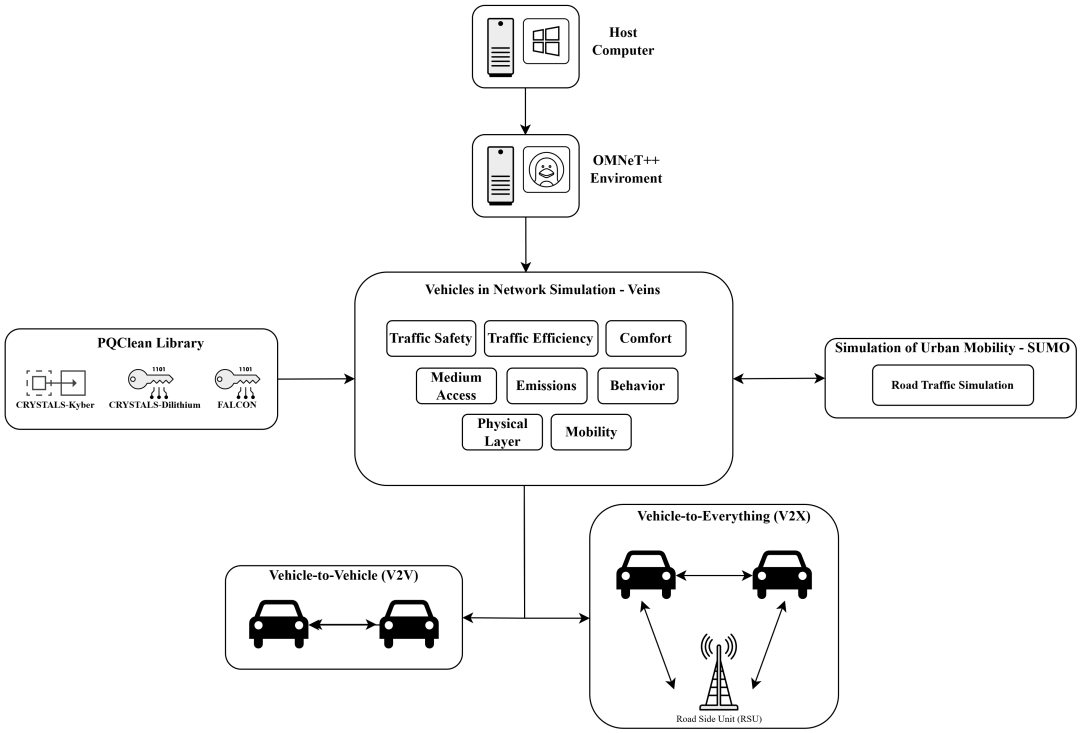


FIGURE 4.1. Architecture of the experimental simulation

There are numerous parameters that we can modify to obtain different results. These parameters are very extensive and can alter the simulation duration, the exact time of the accident, the number of accidents, and the duration of the accidents, all of which are found in the omnetpp.ini file in the Veins example. On the other hand, to modify variables more specifically related to road traffic, you will need to do so in SUMO.

CHAPTER 5

Results

The values displayed on the next tables (sections 5.1 and 5.2) come from a calculated average of the total of vehicles, in V2V scenario there are three cars and V2X with above fifty approximately. We run the simulations multiple times to assure that the results are coherent, of course sometimes we get some outliers, but we include them in the calculations to get a more "real" result. The selected PQC schemes differ significantly in the size of their key material and ciphertexts, which directly impacts performance and communication overhead. In the PQClean implementations, Kyber, Dilithium, and Falcon exhibit different trade-offs. Kyber, optimized for balanced performance, offering relatively small public keys and ciphertexts along with a fixed shared secret size. Dilithium, designed for digital signatures, prioritizes simplicity resulting in larger public and secret keys as well as sizable signatures, but with efficient verification. Falcon, also a signature scheme, achieves much smaller signature sizes compared to Dilithium, though at the cost of more complex arithmetic.

TABLE 5.1. PQClean Schemes - Sizes (Bytes)

Scheme	Secret Key	Public Key	Shared Secret	Cipher Text
Kyber-512	1632	800	32	768
Kyber-768	2400	1184	32	1088
Kyber-1024	3168	1568	32	1568
Dilithium-44	1312	2560	2420	—
Dilithium-65	1952	4032	3309	—
Dilithium-87	2592	4896	4627	—
Falcon-512	1281	897	752	—
Falcon-1024	2305	1793	1462	—

5.1. Rural Scenario (V2V)

The Veins-Inet example consists of three vehicles. One of them is involved in an accident and subsequently broadcasts a warning message to the others. Upon receiving this message, the remaining vehicles adjust their trajectories accordingly.

Every vehicle acts like trust center, they send each other the public key and ciphertext. The vehicle which sends the accident alert, encrypts the message with AES-256 with the

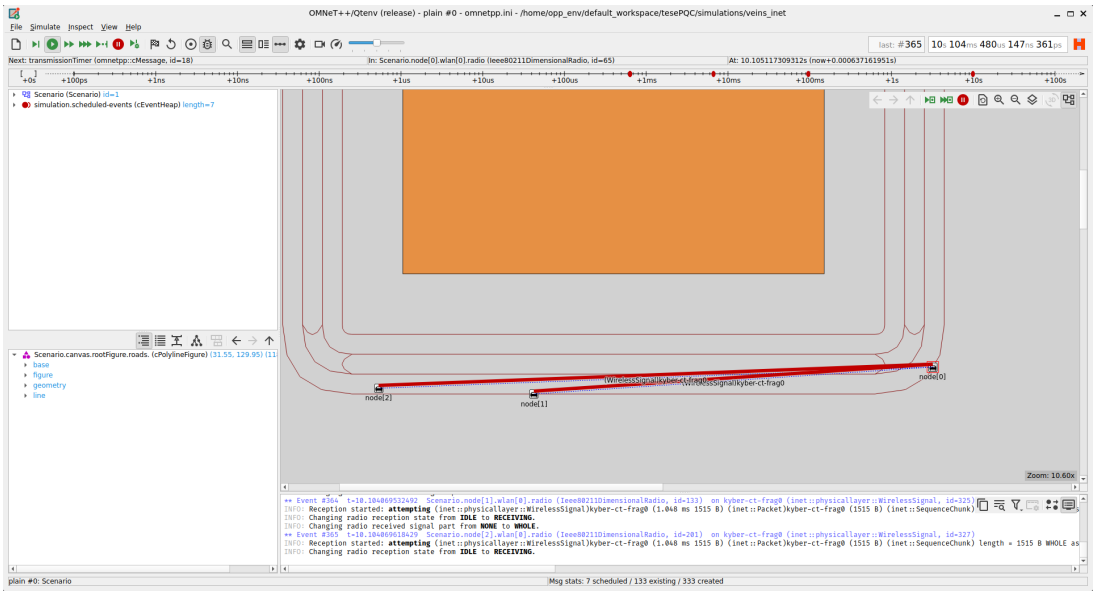


FIGURE 5.1. Rural Scenario - Car alerting others cars of his accident

now shared secret. The tables in this section present Hardware 1 (above) and 2 (under) results.

TABLE 5.2. Kyber Rural Scenario

Scheme	KeyGen (μs)	Encapsulation (μs)	Decapsulation (μs)
Kyber-512	46,33	53,28	62,33
Kyber-768	76,56	75,89	92,11
Kyber-1024	106,56	119,11	133,33
Kyber-512	39,67	49	63,22
Kyber-768	61,11	70,44	88,72
Kyber-1024	99	106,11	130,72

TABLE 5.3. Dilithium - Rural Scenario

Scheme	Key Generation (μs)	Signature (μs)	Verification (μs)
Dilithium-44	122,33	306,33	124,91
Dilithium-65	221,56	367,67	200,62
Dilithium-87	316,56	554,33	318,86
Dilithium-44	122,22	274,33	111,74
Dilithium-65	167,11	284,67	178,22
Dilithium-87	250,11	498,67	275,81

TABLE 5.4. Falcon - Rural Scenario

Scheme	Key Generation (μs)	Signature (μs)	Verification (μs)
Falcon-512	17533	7186,33	57,76
Falcon-1024	41544,44	13139	110,47
Falcon-512	14344,11	3848,33	48,29
Falcon-1024	36992,44	8308,33	94,81

5.2. Urban Scenario (V2X)

In this V2X scenario, approximately fifty vehicles are considered, along with a single RSU. Following the same logic as in the V2V scenario, when a vehicle is involved in an accident, it broadcasts an alert message to nearby vehicles and the RSU. Upon receiving the message, the RSU relays it over a wider range, allowing vehicles that were outside the initial broadcast coverage to also receive the alert.

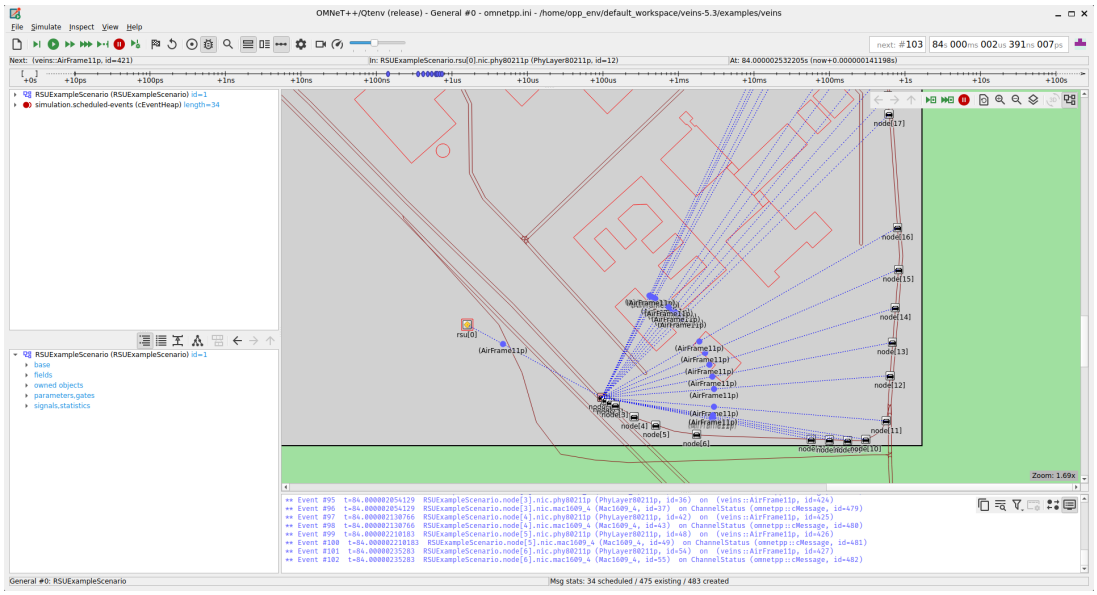


FIGURE 5.2. Urban Scenario - Car alerting the RSU and others vehicles

In this scenario, the RSU acts like a trust center by generating a key pair. It then broadcasts its public key, then vehicles proceed with the encapsulation of ciphertext with the received public key. And concluding this KEM system, the RSU receives the ciphertext and would decapsulate, and both parties would have access to the secret. As the number of vehicles in the network increases, the volume of transmitted packets also grows significantly. This leads to increased packet collisions at the MAC layer (e.g., under IEEE 802.11p), resulting in the loss of multiple ciphertext packets. This would happen because vehicles would send them simultaneously after receiving the broadcast public key. It created bursts of transmissions and would flood the channel and because of that many

handshakes wouldn't be completed. To mitigate this issue, we had to induce a random transmission interval before sending the ciphertexts, that way it lowered the probability of collisions. Before implementing this method, the handshake success rate was 23,07% and after implement it, we observed a 40% improvement (still many weren't completed due to the limited simulation time). The latency considers the network, MAC, propagation and random backoff of the scenario, overall latency measured 11,04 ms (Hardware 1) and 9,87 ms (Hardware 2).

TABLE 5.5. Kyber - Urban Scenario

Scheme	Key Generation (μs)	Encapsulation (μs)	Decapsulation (μs)
Kyber-512	49,67	63,98	68,82
Kyber-768	72	87,54	109,41
Kyber-1024	111,33	133,93	153,33
Kyber-512	47,67	55,27	62,32
Kyber-768	69,67	77,57	101,29
Kyber-1024	90,67	118,98	135,41

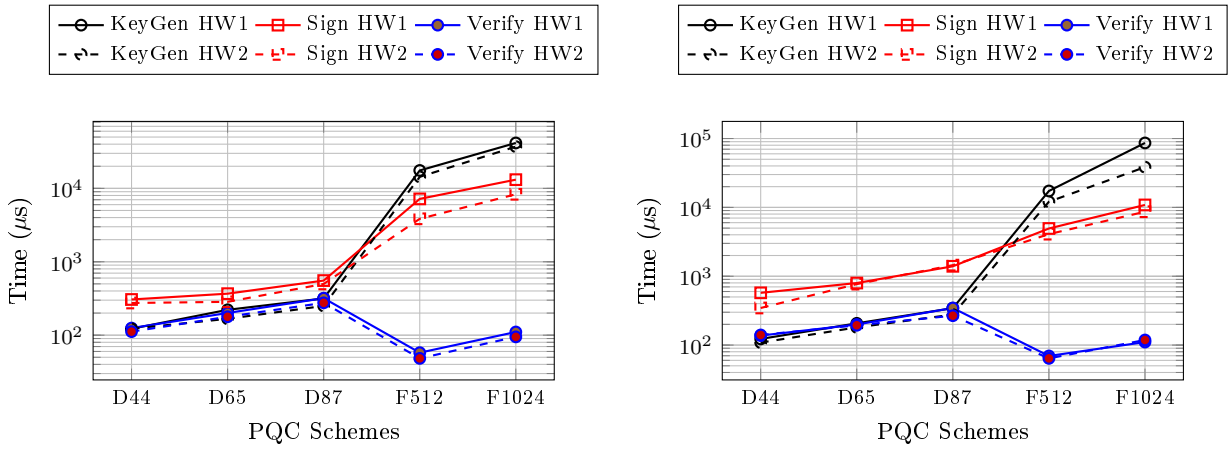
In the digital signature scenario, once the vehicle has an accident, it signs the message and broadcasts it to all vehicles and RSU present. Once the vehicles receive the message, they verified it, if it's valid they change the lane to avoid the accident, on the other hand, they simply drop the message. As for the RSU, it proceeds with the same verification process, but if message is valid, it rebroadcasts it, to reach other vehicles that are further away. The lantecy measured for Dilithium was 2,12 s, sligthly slower than Falcon. Falcon demonstrates a relatively low overall latency of 2,08 s when accounting for signing, verification, MAC operations, and networking overhead. However, Dilithium is capable of performing more efficiently on the second hardware and outperforms Falcon with a latency of 2,0062 s , whereas Falcon has a latency of 2,0127 s.

TABLE 5.6. Dilithium - Urban Scenario

Scheme	Key Generation (μs)	Signature (μs)	Verification (μs)
Dilithium-44	122,79	574,17	137,38
Dilithium-65	206,95	796,67	198,91
Dilithium-87	343,84	1394	345,39
Dilithium-44	108,70	345,83	140,20
Dilithium-65	180,67	768,17	193,98
Dilithium-87	277,30	1450	267,16

TABLE 5.7. Falcon - Urban Scenario

Scheme	Key Generation (μs)	Signature (μs)	Verification (μs)
Falcon-512	17316,15	4936,67	69,49
Falcon-1024	86291,81	10849,83	110,55
Falcon-512	12083,79	4086,5	64,09
Falcon-1024	38437,77	8646,5	118,49



(A) Rural scenario (V2V)

(B) Urban scenario (V2X)

FIGURE 5.3. Comparison of PQC performance on both scenarios

Both graphs show the execution times of different cryptographic operations in the digital signature schemes. The x-axis lists each scheme, and the y-axis shows the execution times for the three operations in each scheme: key generation, encapsulation/decapsulation, and signing/verification. As we can see, each scheme has its own execution times for each operation. In graphs 5.3a and 5.3b, Falcon-512 and Falcon-1024 show a relatively fast verification time compared to signing and key generation, as the data point representing verification is in the units and tens range, respectively, while the others are in the thousands. However, Dilithium performs quite consistently across all operations in general.

When it comes to the automotive context, we must ensure that the schemes that we're using don't disturb the fundamental functionalities within a vehicle and as observed in the simulated scenario, every cryptographic function operates in the order of microseconds which is very good sign for V2X. Kyber could be a good candidate to coperate with Dilithium or Falcon. At first after comparing the two digital signature schemes we may discard Falcon. But even though Falcon takes more time in generating keys and signing, its verification average always beats Dilithium. So creating a protocol with Kyber and Falcon, would actually work very efficiently. Since the verification function it's the more frequent during V2X and V2V communications, it would be reasonable to make the effort with Falcon, because we only generate keys once and the signing is more time-consuming,

the verification time could be very beneficial specially in V2X scenarios. Comparing the two scenarios, we could see that V2X requires sligthly more time than V2V because, V2X has more identities interating with each other, affecting the latency.

CHAPTER 6

Conclusion

In this thesis, we evaluated the performance of lattice-based post-quantum cryptographic schemes: CRYSTALS-Kyber, CRYSTALS-Dilithium, and Falcon in V2X and V2V communication scenarios. The analysis considered both rural (V2V) and urban (V2X) environments, with the aim of assessing performance under varying network conditions and levels of communication stress. The results indicate that Kyber is a strong candidate for integration into future protocols, due to its consistently favorable performance across both scenarios. Regarding digital signature schemes, Dilithium and Falcon exhibit distinct trade-offs, however both demonstrate characteristics that make them suitable for deployment in automotive communication systems. In particular, while FALCON achieves lower latency in verification, Dilithium provides a balanced performance on overall operations. To mitigate communication overhead, periodic packet transmission is identified as a potential optimization strategy, reducing channel contention among vehicles. The impact of post-quantum cryptography is more pronounced in V2X scenarios, largely due to the relative efficiency of key encapsulation mechanisms such as Kyber when compared to signature-based approaches. Overall, all evaluated schemes demonstrate effective performance in both V2V and V2X contexts, supporting their feasibility for future adoption. Additionally, this work produced a dataset [50], comprising key generation, encapsulation/decapsulation and signing/verification timings for each scheme. This dataset provides a foundation for future research, including the application of machine learning techniques to dynamically select the most appropriate cryptographic scheme based on system conditions.

6.1. Limitations

The primary limitation lies in the use of a virtual simulation environment. Although the Veins framework enables the modeling of complex scenarios, it cannot fully capture real-world problems such as signal interference, unpredictable channel conditions and hardware-specific constraints. Consequently, the results may not fully reflect performance in real deployment settings. A second limitation concerns the simplicity of the V2X scenario considered. The simulated environment includes only vehicles and a RSU, restricting communication to V2V and V2I interactions. This abstraction does not account for more large-scale V2X ecosystems, which may include additional infrastructure, pedestrians and network.

6.2. Future Works

This work only tested with lattice-based schemes due to their popularity among PQC schemes, because of their mathematical complexity can garantee security as discussed in section 2.1.2.1. But that leaves us with still multiple PQC schemes, like SPHINCS$^+$ (Hash-based), HQC (Code-based) which is KEM whose security is based on the hardness of solving the Quasi-Cyclic Syndrome Decoding (QCSD) problem, Saber (Lattice-based) also a KEM that uses Learning with Rounding (LWR). Now is the time to start and experimenting like the authors of [46] which joined two schemes: Kyber and Syber to create a efficient protocol. Not only joining two PQC schemes but also creating hybrid protocols which can operate at all situations. With the raise of Artificial Intelligence it is only natural to implement it in this context. Complementing the idea of analyzing and testing with every PQC scheme, the AI could train and decide which scheme is the more suitable in that situation. Concluding the segment of section 4.2.1 and section 6.1 references a set of possibilities when it comes to hardware. The study referenced in that exact section only tests the S32G. The hardware mentioned is PQC-recommended by NXP, so it would be very compelling to analyze the results. On top of everything, we still have to deal with attacks of every kind, as reported in papers like [51] which alerts about the efficiency of side-channel attacks on schemes like Kyber, Saber, Dilithium, HQC, we need to be cautions by applying the appropriate countermeasures.

References

[1] F. Vasconcelos, *Quantum Computing @ MIT: The Past, Present, and Future of the Second Revolution in Computing*, arXiv:2002.05559 [physics.pop-ph], Feb. 2020. DOI: 10.48550/arXiv.2002.05559 [Online]. Available: http://arxiv.org/abs/2002.05559

[2] D. Q. Community. "Denmark to acquire the world´s most powerful quantum computer." Accessed: October 2025. [Online]. Available: https://dqc.dk/denmark-to-acquire-the-worlds-most-powerful-quantum-computer/

[3] M. Swayne. "Quantum computing roadmaps & predictions of leading players." Accessed: October 2025. [Online]. Available: https://thequantuminsider.com/2025/05/16/quantum-computing-roadmaps-a-look-at-the-maps-and-predictions-of-major-quantum-players/

[4] E. Manson and R. Mood, *Elliptic Curve Cryptography in Vehicle Security*, Jan. 2025. DOI: 10.36227/techrxiv.173611692.29943420/v1

[5] S. Zoting. "Automotive pqc market driven by rising cybersecurity threats and quantum-safe certification." Accessed: October 2025. [Online]. Available: https://www.precedenceresearch.com/automotive-pqc-market

[6] M. Swayne. "Automotive pqc market driven by rising cybersecurity threats and quantum-safe certification." Accessed: October 2025. [Online]. Available: https://thequantuminsider.com/2025/09/13/how-is-nvidia-betting-on-quantum-a-look-at-nventures-quantum-investment-playbook-from-skepticism-to-strategic-stakes/

[7] NIST. "Post-quantum cryptography." Accessed: November 2025. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/selected-algorithms

[8] J. Vom Brocke, A. Hevner, and A. Maedche, "Introduction to Design Science Research," en, in *Design Science Research. Cases*, J. Vom Brocke, A. Hevner, and A. Maedche, Eds., Series Title: Progress in IS, Cham: Springer International Publishing, 2020, pp. 1–13, ISBN: 978-3-030-46780-7 978-3-030-46781-4. DOI: 10.1007/978-3-030-46781-4_1

[9] S. Wiesner. "Conjugate coding." Accessed: November 2025. [Online]. Available: https://edu.itp.phys.ethz.ch/hs12/qsit/wiesner.pdf

[10] A. Carrasco-Casado, V. Fernández, and N. Denisenko, "Free-Space Quantum Key Distribution," in arXiv:1611.03529 [physics], 2016, pp. 589–607. DOI: 10.1007/978-3-319-30201-0_27

[11] "Quantum cryptography." Accessed: November 2025. [Online]. Available: https://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-05/cryptography/quantum.html

[12] "Quantum cryptography - shor's algorithm explained." Accessed: November 2025. [Online]. Available: https://www.classiq.io/insights/shors-algorithm-explained

[13] A. Bhalla, K. Eguro, and M. Hayward. "Quantum computing, shor's algorithm, and parallelism." Accessed: November 2025. [Online]. Available: https://quantum-algorithms.herokuapp.com/433/shor-par.pdf

[14] L. K. Grover, *A fast quantum mechanical algorithm for database search*, Version Number: 3, 1996. DOI: 10.48550/ARXIV.QUANT-PH/9605043

[15] H. Sohail, V. Leithardt, and A. Trigo, "Prisec iii: Cryptographic techniques for enhanced security," May 2025, pp. 354–361. DOI: 10.1109/CSCS66924.2025.00059

[16] D. Noetzold, J. L. V. Barbosa, J. F. P. Santana, and V. R. Q. Leithardt, "Q-OPSEC: An Architecture Proposal for Adaptive AI Middleware for Quantum Cryptography," en, in *2025 International Conference on Electrical and Computer Engineering Researches (ICECER)*, Antananarivo, Madagascar: IEEE, Dec. 2025, pp. 1–6, ISBN: 978-1-6654-5756-9. DOI: 10.1109/ICECER65523.2025.11401311

[17] V. Preysing. "An introduction to postquantum cryptography." Accessed: November 2025. [Online]. Available: https://page.eviden.com/rs/247-MBJ-716/images/Eviden-An%20introduction%20to%20Post-quantum%20cryptograph%20PQC-cryptovision-Klaus%20Schmeh-White%20paper.pdf?version=0

[18] J. Surin and S. Cohney, *A gentle tutorial for lattice-based cryptanalysis*, Cryptology ePrint Archive, Paper 2023/032, 2023. [Online]. Available: https://eprint.iacr.org/2023/032

[19] S. M. Hosseini and H. Pilaram, *A comprehensive review of post-quantum cryptography: Challenges and advances*, Cryptology ePrint Archive, Paper 2024/1940, 2024. [Online]. Available: https://eprint.iacr.org/2024/1940

[20] D. Noetzold and V. Leithardt, *Fundamentals, recent advances, and challenges regarding cryptographic algorithms for the quantum computing era*, Jan. 2026. DOI: 10.48550/arXiv.2601.18413

[21] V. Srivastava, A. Baksi, and S. K. Debnath, *An overview of hash based signatures*, Cryptology ePrint Archive, Paper 2023/411, 2023. [Online]. Available: https://eprint.iacr.org/2023/411

[22] R. Relyea. "Post-quantum cryptography: Code-based cryptography." Accessed: November 2025. [Online]. Available: https://www.redhat.com/en/blog/post-quantum-cryptography-code-based-cryptography

[23] I. Corporation. "Isogeny-based cryptography tutorial." Accessed: November 2025. [Online]. Available: https://www.isara.com/downloads/crypto_tutorials/Intro-to-Iso-v3.pdf

[24] R. Gonzalez. "Kyber - how does it work?" Accessed: November 2025. [Online]. Available: https://cryptopedia.dev/posts/kyber/

[25] R. Relyea. "Post-quantum cryptography: Lattice-based cryptography." Accessed: November 2025. [Online]. Available: https://www.redhat.com/en/blog/post-quantum-cryptography-lattice-based-cryptography

[26] C.V.Vredendaal, S.Dragone, B.Hess, T.Visegrady, M.Osborne, D.Bong, and J.Bos. "Quantum safe cryptography key information for falcon." Accessed: November 2025. [Online]. Available: https://www.ietf.org/archive/id/draft-uni-qsckeys-falcon-00.html

[27] E. Rahn. "Sphincs+ - step by step." Accessed: November 2025. [Online]. Available: https://er4hn.info/blog/2023.12.16-sphincs_plus-step-by-step/

[28] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, *The SPHINCS+ signature framework*, Cryptology ePrint Archive, Paper 2019/1086, 2019. DOI: 10.1145/3319535.3363229 [Online]. Available: https://eprint.iacr.org/2019/1086

[29] H. L. Tie. "V2x vehicle-to-everything communication – the future of autonomous connectivity." Accessed: November 2025. [Online]. Available: https://www.keysight.com/blogs/en/inds/auto/2024/10/03/v2x-post

[30] E. Farsimadan, L. Moradi, and F. Palmieri, "A review on security challenges in v2x communications technology for vanets," *IEEE Access*, vol. 13, pp. 31 069–31 094, 2025, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2025.3541035

[31] G. Guette and C. Bryce, "Using TPMs to Secure Vehicular Ad-Hoc Networks (VANETs)," in *Information Security Theory and Practices. Smart Devices, Convergence and Next Generation Networks*, J. A. Onieva, D. Sauveron, S. Chaumette, D. Gollmann, and K. Markantonakis, Eds., vol. 5019, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 106–116, ISBN: 978-3-540-79965-8 978-3-540-79966-5. DOI: 10.1007/978-3-540-79966-5_8

[32] Z.-R. Tzoannos, D. Kosmanos, A. Xenakis, and C. Chaikalis. "The impact of spoofing attacks in connected autonomous vehicles under traffic congestion conditions." Accessed: December 2025. [Online]. Available: https://www.mdpi.com/2673-4001/5/3/37

[33] W. Wang and M. Stöttinger, *Post-quantum secure architectures for automotive hardware secure modules*, Cryptology ePrint Archive, Paper 2020/026, 2020. [Online]. Available: https://eprint.iacr.org/2020/026

[34] J. W. Bos, A. Dima, A. Kiening, and J. Renes, *Post-quantum secure over-the-air update of automotive systems*, Cryptology ePrint Archive, Paper 2023/965, 2023. [Online]. Available: https://eprint.iacr.org/2023/965

[35] A. Rathweg, A. Osman, A. Fleig, E. Katsigianni, and A. Tschache. "Impacts of post-quantum cryptography on automotive security: A case study. "[Online]. Available: https://d-nb.info/1316435830/34

[36] G. Twardokus, N. Bindel, H. Rahbari, and S. McCarthy, "When Cryptography Needs a Hand: Practical Post-Quantum Authentication for V2V Communications," en, in *Proceedings 2024 Network and Distributed System Security Symposium*, San Diego, CA, USA: Internet Society, 2024, ISBN: 978-1-891562-93-8. DOI: 10.14722/ndss.2024.24267

[37] T. Fritzmann, J. Vith, and J. Sepulveda, "Strengthening Post-Quantum Security for Automotive Systems," in *2020 23rd Euromicro Conference on Digital System Design (DSD)*, Kranj, Slovenia: IEEE, Aug. 2020, pp. 570–576, ISBN: 978-1-7281-9535-3. DOI: 10.1109/DSD51259.2020.00094

[38] M. L. Manna, P. Perazzo, L. Treccozzi, and G. Dini. "Assessing the cost of quantum security for automotive over-the-air updates. "[Online]. Available: https://ieeexplore.ieee.org/document/9631426

[39] T. Yoshizawa, D. Singelée, J. T. Muehlberg, S. Delbruel, A. Taherkordi, D. Hughes, and B. Preneel, "A Survey of Security and Privacy Issues in V2X Communication Systems," en, *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–36, Sep. 2023, ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3558052

[40] A. Castiglione and T. Elia, "Securing in-vehicle communications through post-quantum cryptography and network segmentation," en, *Computers and Electrical Engineering*, vol. 126, p. 110 488, Aug. 2025, ISSN: 00457906. DOI: 10.1016/j.compeleceng.2025.110488

[41] A. Abrar, S. Dasgupta, M. Rahman, and A. Alsharif, *AI-Driven Post-Quantum Cryptography for Cyber-Resilient V2X Communication in Transportation Cyber-Physical Systems*, Version Number: 1, 2025. DOI: 10.48550/ARXIV.2510.08496

[42] A. A. Mamun, A. Abrar, M. Rahman, M. S. Salek, and M. Chowdhury, *Post-Quantum Cryptography for Intelligent Transportation Systems: An Implementation-Focused Review*, arXiv:2601.01068 [cs], Jan. 2026. DOI: 10.48550/arXiv.2601.01068

[43] T. Fritzmann, J. Vith, D. Flórez, and J. Sepúlveda, "Post-quantum cryptography for automotive systems," en, *Microprocessors and Microsystems*, vol. 87, p. 104 379, Nov. 2021, ISSN: 01419331. DOI: 10.1016/j.micpro.2021.104379

[44] S. A. Abdel Hakeem and H. Kim, "PQ-V2X: A Novel Postquantum Cryptographic Dataset for Secure Vehicular Communications," *IEEE Internet of Things Journal*, vol. 12, no. 24, pp. 53 704–53 729, Dec. 2025, ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2025.3618153

[45] A. S. Da Silva, K. H. M. Gularte, G. A. Santos, D. S. S. Corrêa, L. F. Oliveira De Melo, J. P. J. Da Costa, J. A. R. Vargas, D. A. Da Silva, and T. Fei, *Evaluation of Jamming Attacks on NR-V2X Systems: Simulation and Experimental Perspectives*, Jul. 2025. DOI: 10.20944/preprints202507.0729.v1

[46] Y. Dai, Q. Wang, X. Song, and S. Wang, "A Lightweight Key Agreement Protocol for V2X Communications Based on Kyber and Saber," en, *Sensors*, vol. 25, no. 22, p. 6938, Nov. 2025, ISSN: 1424-8220. DOI: 10.3390/s25226938

[47] M. S. Mohamed, J. Godard, V. Jimenez, A. Jousse, P. P. Paños, and M. Zhang, "Post-quantum Cryptography for Connected and Cooperative Automated Mobility: A Comprehensive Overview," en, in *Transport Transitions: Advancing Sustainable and Inclusive Mobility*, C. McNally, P. Carroll, B. Martinez-Pastor, B. Ghosh, M. Efthymiou, and N. Valantasis-Kanellos, Eds., Series Title: Lecture Notes in Mobility, Cham: Springer Nature Switzerland, 2026, pp. 738–744, ISBN: 978-3-032-06762-3 978-3-032-06763-0. DOI: 10.1007/978-3-032-06763-0_105

[48] "Post-quantum cryptography | setting the future security standards." Accessed: March 2025. [Online]. Available: https://www.nxp.com/applications/technologies/security/post-quantum-cryptography

[49] M. J. Kannwischer, P. Schwabe, D. Stebila, and T. Wiggers, "Improving software quality in cryptography standardization projects," in *IEEE European Symposium on Security and Privacy, EuroS&P 2022 - Workshops, Genoa, Italy, June 6-10, 2022*, Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 19–30. DOI: 10.1109/EuroSPW55150.2022.00010

[50] M. Gonçalves and V. Leithardt, *Post-quantum cryptography in automotive scenarios (v2x and v2v)*, Mendeley Data, 2026. DOI: 10.17632/czx6vk5hsr.1 [Online]. Available: https://data.mendeley.com/datasets/czx6vk5hsr/1

[51] A. Olaluwe, N. N. Nabilah, S. Tareq, A. R. Kulkarni, and A. Annamalai, *Machine learning and side-channel attacks on post-quantum cryptography*, Cryptology ePrint Archive, Paper 2025/1754, Accessed: December 2025, 2025. [Online]. Available: https://eprint.iacr.org/2025/1754

APPENDIX A

Architecture

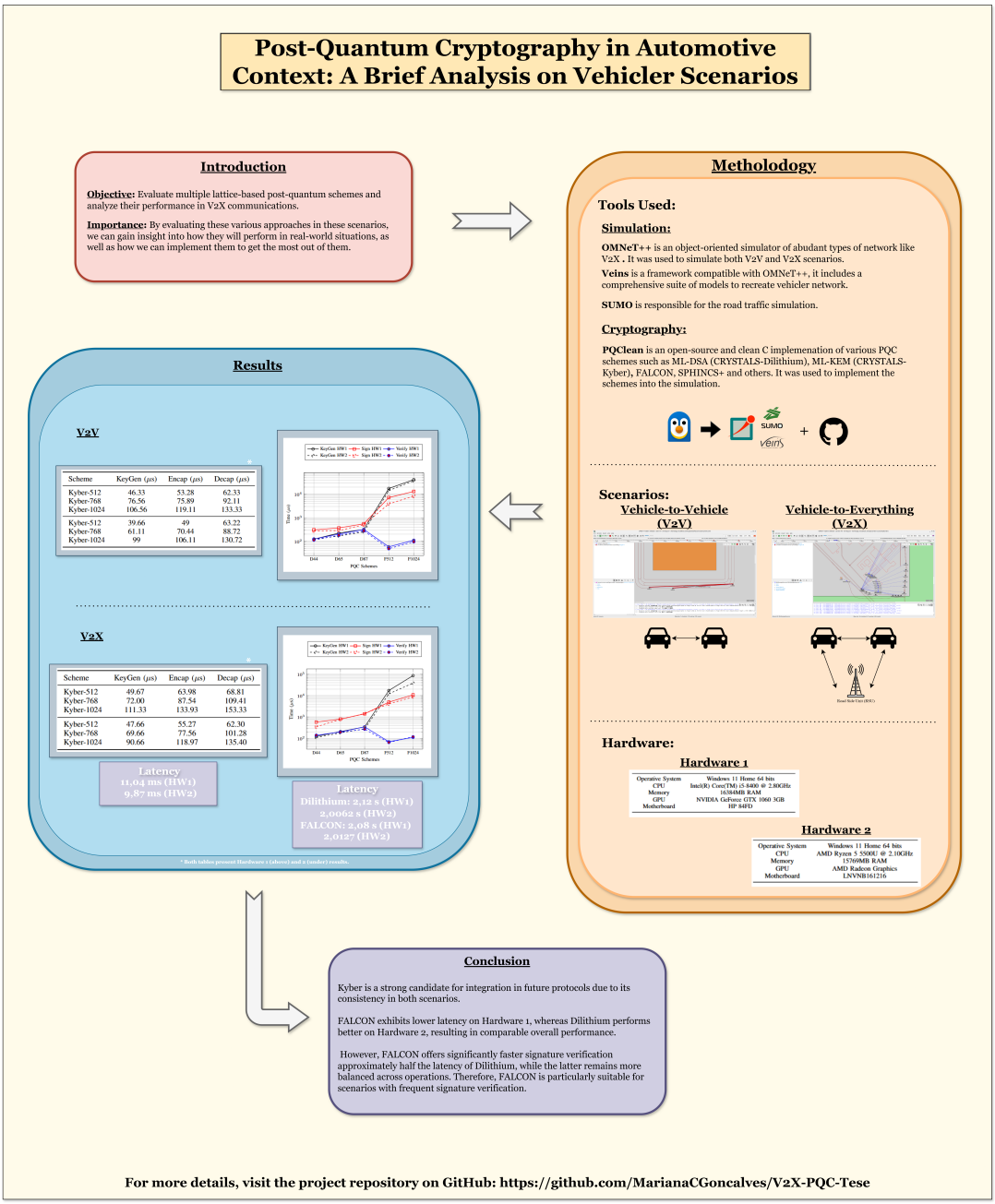


FIGURE A.1. Full architecture of the developed application

APPENDIX B

Submitted Article





Regular Manuscript

Post-Quantum Cryptography in Automotive Context

Submission ID	bbece0ed-9a52-4eb9-a20f-1d7229c73934
Manuscript ID	draft-bbece0ed-9a52-4eb9-a20f-1d7229c73934
Submission Version	Initial Submission
PDF Generation	12 Jun 2026 07:25:55 EST by Atypon ReX

Authors

Miss MARIANA GONÇALVES

Affiliations

- Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, 1649-026, Portugal

Prof. LUÍS B. OLIVEIRA

ORCiD
https://orcid.org/0000-0002-0624-1775

Affiliations

- Department of Electrical and Computer Engineering (DEEC), NOVA School of Science and Technology, Center of Technology and Systems (CTS-UNINOVA), Associated Laboratory of Intelligent Systems (LASI), 2829-516 Lisbon, Portugal

Prof. VALDERI R. Q. LEITHARDT
Corresponding Author
Submitting Author

ORCiD
https://orcid.org/0000-0003-0446-9271

Affiliations

- Instituto Universitário de Lisboa (ISCTE-IUL), ISTAR, Lisboa, 1649-026, Portugal

Additional Information

Subject Category

Communications technology

Engineering management

Science - general

Systems, man, and cybernetics
Vehicular and wireless technologies

Keywords
Automotive applications
Cryptography
NIST
Quantum cryptography
Vehicle-to-everything

Select a Manuscript Type
Applied Research

Files for peer review

All files submitted by the author for peer review are listed below. Files that could not be converted to PDF are indicated; reviewers are able to access them online.

Name	Type of File	Size	Page
Post_Quantum_Cryptography_in_Automative_Context.pdf	Main Document - PDF	1.7 MB	Page 4