ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
DOUTORADO EM CIÊNCIA DA COMPUTAÇÃO

GABRIEL ROSSI FIGLARZ

ENHANCEMENT OF THE SECURITY IN LORAWAN COMMUNICATION WITH POST-QUANTUM-CRYPTOGRAPHY

Porto Alegre
2025

PONTIFICAL CATHOLIC UNIVERSITY OF RIO GRANDE DO SUL
SCHOOL OF TECHNOLOGY
COMPUTER SCIENCE GRADUATE PROGRAM

ENHANCEMENT OF THE SECURITY IN LORAWAN COMMUNICATION WITH POST-QUANTUM-CRYPTOGRAPHY

GABRIEL ROSSI FIGLARZ

Doctoral Thesis submitted to the Pontifical Catholic University of Rio Grande do Sul in partial fulfillment of the requirements for the degree of Ph. D. in Computer Science.

Advisor: Prof. Dr. Fabiano Passuelo Hessel

Porto Alegre
2025

Ficha Catalográfica

F472e Figlarz, Gabriel Rossi

Enhancement of the Security in LoRaWAN Communication with Post-Quantum-Cryptography / Gabriel Rossi Figlarz. – 2025.
93.
Tese (Doutorado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientador: Prof. Dr. Fabiano Passuelo Hessel.

1. Post-Quantum Cryptography. 2. KEM. 3. LoRaWAN. 4. IoT security. I. Hessel, Fabiano Passuelo. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS
com os dados fornecidos pelo(a) autor(a).
Bibliotecária responsável: Clarissa Jesinska Selbach CRB-10/2051

GABRIEL ROSSI FIGLARZ

ENHANCEMENT OF THE SECURITY IN LORAWAN COMMUNICATION WITH POST-QUANTUM-CRYPTOGRAPHY

This Doctoral Thesis has been submitted in partial fulfillment of the requirements for the degree of Ph. D. in Computer Science of the Computer Science Graduate Program, School of Technology of the Pontifical Catholic University of Rio Grande do Sul

Sanctioned on March 24, 2025.

COMMITTEE MEMBERS:

Prof. Dr. Jarbas Silveira (PPGETI/UFC)

Prof. Dr. Leonel Tedesco (PPGSPI/UNISC)

Prof. Dr. César Augusto Missio Marcon (PPGCC/PUCRS)

Prof. Dr. Fabiano Passuelo Hessel (PPGCC/PUCRS - Advisor)

ACKNOWLEDGMENTS

Agradeço a minha família Mãe, Pai, Lia e Nicole por tudo.

Também, ao Professor Fabiano Hessel pela oportunidade de ser seu orientado desde o Mestrado.

APERFEIÇOAMENTO DA SEGURANÇA EM COMUNICAÇÕES LORAWAN COM CRIPTOGRAFICA PÓS-QUÂNTICA

RESUMO

LoRaWAN é um protocolo que trasmite através de ondas de rádio informação por longas distâncias. A crucialidade da informação comunicada atrai o interesse de ataques mal-intencionados que tem como objetivo de interceptar as mensagens. Ações mitigadoras usadas para garantir a segurança na comunicação em LoRaWAN estãos sob ameaça dos avançoes da Computação Quântica (QC). Uma alternativa conhecida para garantir a segurança da comunicação contra ataques de computadores quânticos e classicos é a criptografia-pós-quântica (PQC). PQC é composta por problemas matemáticos complexos que computadores quânticos ainda não são capazes de resolvê-los. Portanto, PQC pode garantir comunicação segura entre dispositivos de IoT quando computadores quânticos atingirem seu potencial. Considerando isto, este trabalho propõe um aprimoramento na segurança da comunicação em LoRaWAN implementando o algoritmo ML-KEM1024 no protocolo. Para isso, uma aplicação foi desenvolvida para simular a comunicação em um ambiente LoRaWAN onde um dispositivo manda uma mensagem para um gateway, um servidor de rede e um servidor de aplicação através de uma rede socket. A performance em cada passo do algoritmo KEM foi comparada com uma referência. Foram alcançados resultados satisfatórios de performance quando comparados com as referências em diversos casos. Isso corrobora que PQC é adequada para comunicação em IoT e que o aumento da segurança é viável para diversas aplicações. Porém, a otimização do tempo de execução e uso de memória pode expandir as áreas de aplicação ainda mais.

Palavras-Chave: Criptografia Pós Quântica, KEM, LoRaWAN, Segurança em IoT.

ENHANCEMENT OF THE SECURITY IN LORAWAN COMMUNICATION WITH POST-QUANTUM-CRYPTOGRAPHY

ABSTRACT

Long Range Wide Area Network (LoRaWAN) is the protocol that encodes information in radio waves and transmitting information through large distances. The cruciality of the information communicated awakens interest from malicious parties to acquire the communicated data via impersonating attacks or keys interception. Mitigating actions to ensure the security in LoRaWAN communication are under the threat of Quantum Computing (QC) advances. Post-quantum-cryprography (PQC) is composed by mathematically complex problems that quantum computers are not able to solve yet. Hence, PQC can ensure secure communication between IoT devices when quantum computers reach their full potential. Considering this, this work proposes an enhancement in LoRaWAN communication by implementing the PQC ML-KEM1024 algorithm in the protocol. An application was developed simulating LoRaWAN communication between a device and the whole circuit, considering a gateway, a network server and, an application server through a socket network. The assessed performance compared with benchmarks the time of execution and memory consumption at each step of the KEM algorithm. It was reached satisfactory level of results comparing to the references in bibliography and the methodology was validated. The results prove that PQC is fit for IoT communications and the enhancement in the security is feasible in many use cases. However, optimizations in the performance would expand the areas of usage of the proposed methodology.

Keywords: Post-Quantum Cryptography, KEM, LoRaWAN, IoT security.

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

CONTENTS

1. INTRODUCTION

The Internet of Things (IoT) is present in the most crucial segments of our society. Among those areas are internet and media devices, vehicles, smart grid, asset tracking and monitoring, water supply, waste management, and agriculture. The number of expected connected IoT devices in 2033 is more than double of a decade before reaching a peak of 39.6 billion as Figure 1.1 shows. Among different technologies, LoRaWAN is present on providing communication on the mentioned areas. LoRaWAN plays an essential role in providing efficient, scalable, and low cost connectivity for applications that require long-range communication.

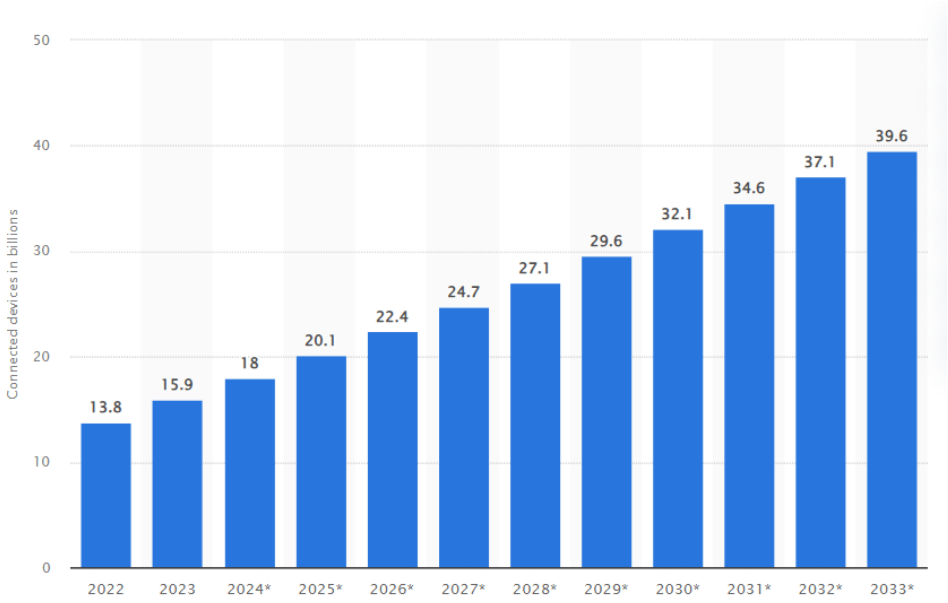

Figure 1.1 – Number of Internet of Things (IoT) connections worldwide to 2033[155]

The massive increase of connected IoT devices and their vast areas of applicability proposes economic growth opportunities and expectations. Figure 1.2 shows the estimated potential of IoT economic growth until 2030 by segment. It is also stated in the McKinsey report that IoT economic value can reach $5.5 trillion to $12.6 trillion by 2030 [99]. From those, the LoRaWAN market size is expected to reach U$202.40 billion by 2034, which represents a compound annual growth rate (CAGR) of 35.90% during the forecast period from 2025 - 2034 [50]. The large sums of value in this area added by the high dependency of essential services on IoT can be translated in a fundamental area for our society.

The vital importance and high-value data that are being transmitted through IoT devices require high-privacy communication that ensures integrity. The digitalization of different areas of life, business and work overall attracts the attention of deceitful parties with the goal of stealing information. It is estimated by Statista that cyber attacks directed

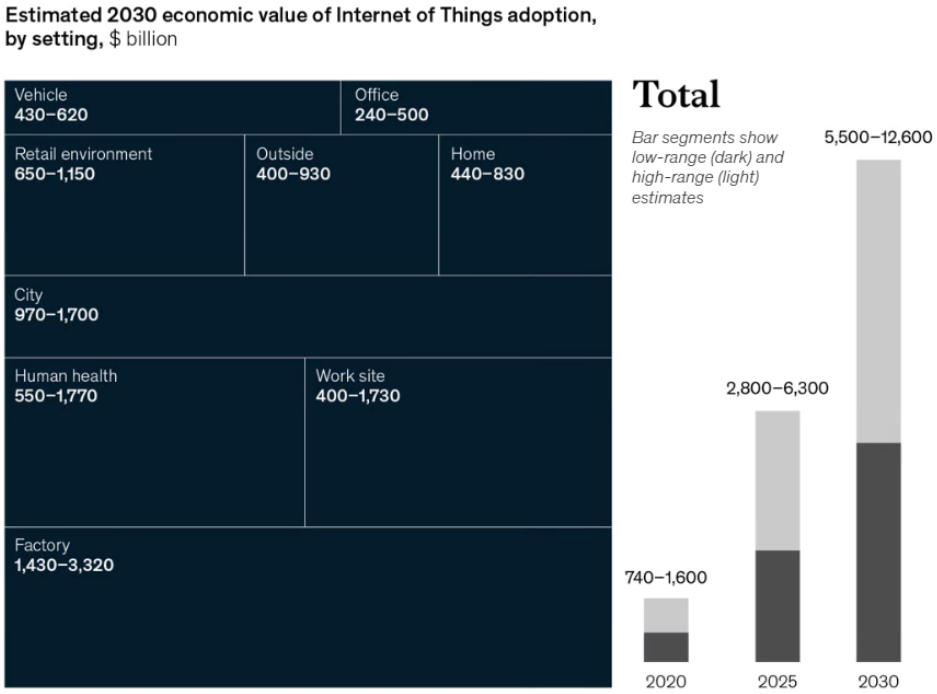

Figure 1.2 – IoT Economic Value Report by McKinsey [99]

to IoT devices in 2022 reached 122 million occasions. Over the years, the figure has increased significantly from 32 million cases detected in 2018. This represents a year-over-year increase of malware incidents in IoT of 87% [154].

Ensuring this level of security for connected devices in a LoRaWAN environment comes with many challenges. As mentioned before, the high volume of connected devices also brings different standards in network protocols. Also, systems are becoming more complex and distributed, creating a trust issue. Additionally to the defense side, hackers also evolve their attacks with more sophisticated techniques. Enabling consistent data confidentiality, integrity, and authenticity in IoT is a constant challenge with a wide universe of opportunities [92].

Computational advances not only allow developers to enable systems with more secure protocols but also hackers to perform more complex attacks. In recent times, Quantum Computing (QC) has grown rapidly in applicability as studies and projects are being developed. The concept is a different form of computation through a quantum computer that leverages quantum mechanics to perform operations that are currently considered very hard to solve by a classical computer. Having the possibility to solve those - until now - problems considered almost impossible to solve, puts current cryptography protocols at risk and hence, IoT communication.

Quantum algorithms performed by malicious parties can break current cryptography protocols, such as AES and SHA, for example. This represents a thread to be mitigated and neutralized before QC technology reaches an application level. For that, post-quantum cryptography (PQC) is a valid solution. It consists of using classical computation

to implement cryptographic solutions that are resistant to classic and quantum computer attacks. This type of cryptography also brings challenges especially in a LoRaWAN universe where the performance in a resource constraint device needs to be optimal, for example.

Quantum Computing and quantum computers are areas that are still in development. However, its potential is being validated at each experiment and study, which is driving big companies to invest on this technology as Amazon, Google, and IBM. Considering the size of the market that LoRaWAN has and potentially will have in the future communicating among critical services, it is impermanent to evolve and enhance the security against quantum computer attacks.

1.1 Hypothesis and Research Questions

This doctorate thesis aims to investigate two hypothesis: (i) the integration of post-quantum-cryptography in IoT devices and, (ii) feasability of post-quantum-cryptography in LoRaWAN without compromising its low performance cost.

Hence, the following research questions were developed to support the hypothesis validation:

1. How necessary is post-quantum-cryptpgraphy in IoT devices and LoRaWAN?
2. What are the main challenges for the implementation of post-quantum-cryptography in IoT in practice?
3. What are the highest security threats currently found in LoRaWAN?
4. Are quantum-algorithms a menace for LoRaWAN's security and can they leverage its current vulnerabilities?

1.2 Objectives

The first thesis objective is to explore post-quantum-cryptography algorithms and find a fit with LoRaWAN's security vulnerabilities. Additionally, it is to propose an implementation in LoRaWAN's architecture enhancing its security against quantum-algorithms. To achieve this objectives, the following goals were defined:

- Explore quantum-computing concepts and reach post-quantum-cryptography algorithms and definitions.

- Keep an up to date research with current developments in quantum computing and standards definitions.
- Evaluate a post-quantum-cryptography algorithm and its applicability in a LoRaWAN environment.
- Study state of the art libraries and resources available for implementation and simulation of post-quantum-cryptography algorithms.
- Connect post-quantum-cryptography and simulate a LoRaWAN based communication.
- Document findings, results and developments and publish in scientific conferences.

1.3 Contribution

This research main contribution is on enhancing IoT security by proposing and validating post-quantum-cryptography in a LoRaWAN environment. The specifications of the contributions are:

- Literature review on quantum-computing and post-quantum-cryptography applied in IoT and LoRaWAN.
- Evaluation of the most fit algorithm in LoRaWAN.
- Enhancement in LoRaWAN security by applying post-quantum-cryptography in the communication.
- Validation of the methodology via a simulation.

1.4 Thesis Outline

This work proposes a security enhancement in the LoRaWAN cryptographic framework. The main goal is to provide quantum-resistant communication between all the involved members in the communication chain. To do that, the state-of-the-art PQC algorithm ML-KEM was considered. ML-KEM is one of the selected algorithms as standard by the National Institute of Standards and Technology (NIST). Besides the security and quantum-resistant advantages, the encryption keys are comparatively small and the speed of operation is adequate. The ML-KEM was implemented in LoRaWAN's architecture and simulated via an application. This allows us to validate the enhancement and assess the results to start the discussion on migrating the current classic cryptography in the protocol or expanding to a hybrid approach.

2. BACKGROUND

This chapter presents theoretical definitions that were studied to develop this work. First, general concepts of the Internet of Things are presented in Section 2.1, followed by the security in IoT in Section 2.2. Next, LoRa and LoRaWAN is detailed in Section 2.3. Next, quantum computing is described in Section 2.4 with general concepts, algorithms in Section 2.5. Finally, LoRaWAN and its cryptogrpahy are described in Section 2.6.

2.1 Internet of Things

The Internet of Things (IoT) is the area that integrates both physical and virtual worlds [79]. The IoT sy is composed of sensors and chips that are embedded in devices allowing them to collect data. Besides that, the devices can communicate between themselves the collected information without any human interaction. The high volume of collected, digested, processed, and stored data has been changing the way decisions are made thanks to IoT. Areas of application for IoT are wide. Due to its versatility and scalability, IoT is present in different parts of organizations including healthcare, financial services and energy [118].

According to Borgia [24], applications will no longer work isolated. They will share environmental, network, and infrastructure all in a common service platform to orchestrate the communication between them. The physical-cyber integration is defined by phases. The first one is the collection phase. In this phase, real-time data is collected from physical sensing devices. Furthermore, the transmission phase is responsible for delivering the data collected previously. This requires access to the network through gateways and other technologies. Finally, the processing phase takes place. This phase deals with analyzing information flows and forwarding data to applications and services, for example.

2.2 IoT Security

Due to the high complexity of managing great amounts of sensor nodes, amount of data, and standards and protocols, IoT devices become a target of cyber-attacks [132]. According to Zhao et al. [174], IoT security differs from Internet Security in definition and complexity. Thus, for each aspect of IoT security, a target solution should be made avoiding generalizations.

Typically, hash functions, symmetric cryptography, and, public-key cryptosystems are usually used to preserve IoT communication [45]. With the advances in computing resources and communications, more opportunities to break asymmetric schemes have appeared, generating threads to communication integrity. Because of that, the minimum key size of crypto algorithms as Rivest–Shamir–Adlema (RSA) has been increasing. However, this is not an ideal solution to ensure algorithm security, since technology advances for both those who attack and those who ensure communication security.

Recently, a new thread to IoT communication has been rising together with advances in Quantum Computing. Statements that the US National Security Agency (NSA) has been building a quantum computer that could break modern internet security and bring concerns to the light have been published in recent years [28]. Additionally, it is estimated that in 20 years, quantum computers will be able to break current strong public-key cryptosystems [105]. As an example of the threat, it is expected for quantum computers to break through RSA-2048 by 2022 with a chance of 1/7. A number that escalates to 1/2 in 2031.

Additionally, one of the findings from the NIST report corroborates with the statement that QC poses a major threat to IoT cybersecurity. It is also mentioned that when quantum algorithms can crack classical encryption in real time, there will be a significant and impactfull issue with the visibility of sensitive data with unprecedented consequences [116].

2.3 LoRa and LoRaWan

As the name indicates, LoRa (Long Range) offers a long communication range with a low energy cost, and due to this characteristic, it is widely used in IoT devices. It is an SS-Spread Spectrum modulation and uses a chirp signal varying with the frequency. Because the offset in time and frequency between the sender and receiver is the same, reduces the receiver's complexity and guarantees the low energy cost [85]. As its most important characteristic, the capability of long-distance communication, LoRa can transmit communication through kilometers in a smart city or a farm, for example, [90].

LoRaWAN is the Low Power, Wide Area (LPWA) networking protocol that determines security, network capacity, and quality of service [90], [7]. In general, LoRa is considered as the physical layer, while LoRaWAN is the medium access control (MAC) layer of the LoRa stack that adopts a star topology. This configuration is responsible for allowing communication between multiple end devices (EDs) and the network gateway[71]. LoRaWAN's technology stack can be observed in Figure 2.1.

The physical layer (PHY) uses air as a medium for transporting LoRa radio waves from an IoT sender to a gateway receiver and vice-versa [113]. The regional ISM band

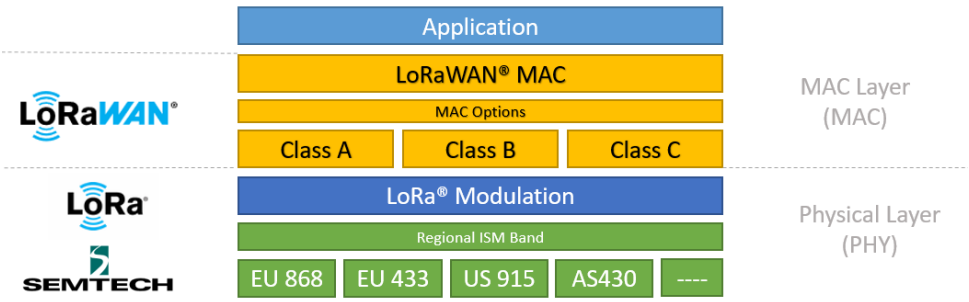

Figure 2.1 – LoRaWAN Technology Stack [156]

parameters, defined by EU 868, EU 433, US 915, and, AS430 are the standard channel frequencies that the communication can operate.

The LoRaWAN Media Access Control (MAC) layer, is built on top of the LoRa PHY layer. Here is where the software defines how devices use Lora's hardware [113]. It is composed of three main components, being the network servers (NS), gateways (GW), and EDs [71]. It also specifies three classes of devices A,B, and C that support bi-directional communication. Class A devices can send an uplink message at any time and open two windows to receive downlink messages from the network. As an extension from the previous, Class B also periodically opens receive windows for downlink messages called ping slots. And, as another extension of Class A, Class C keeps the receive window opened, allowing this class of devices to receive downlink messages at any time [113]

LoRaWAN is established as a star architecture as Figure 2.2 shows. It allows EDs to send data to multiple GWs, which can send data to a distant NS, where security verification, among other activities, can be done [90].

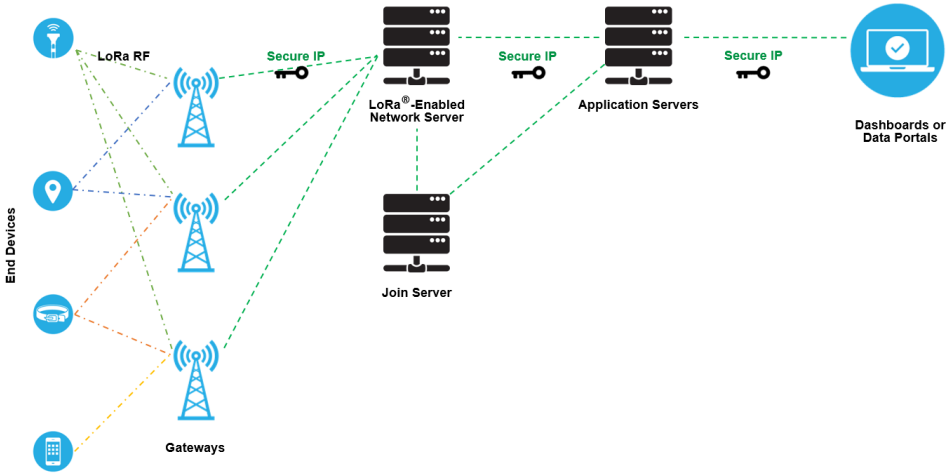

Figure 2.2 – LoRaWAN Architecture [113]

Surrounding LoRaWAN communication, there is valuable information being traded and thus, security and privacy threats as well. Among the most common privacy violations is the Replay attack. This attack is composed of an ED sending a join-request message to the NS. The attacker intercepts the message and jams the channel so the ED does not establish the communication. After a timeout for communication, another message is sent from the ED. The attacker replays the first join-request message and it is accepted by the NS since it is the first time it is being used.

Another example is the Beacon synchronization attack. An attacker can compromise the GW and try to establish communication with multiple EDs. This will start the protocol in the EDs, but without further communication confirmation, none will be successful. The several communication attempts increase the collisions between the transmitted packages, overcrowding the network [120].

LoRaWAN's cryptography is based on AES-128 and, if not safely stored, maintained and tracked, can be broken and reveal information in exploited EDs. As explored by Gunathilake et al. [57], another approach for lightweight cryptography is a necessity for low-power data-processing devices relying in LoRaWAN. One example is the Groover's algorithm, which can reduce the level of security of a cryptographic algorithm, allowing search algorithms to break through security protocols [55]. Therefore, to ensure communication security in LoraWAN, other solutions need to be explored. Post-quantum-cryptography (PQC) algorithm surges as an interesting solution for that, adding a different layer of protection against quantum-computing attacks.

2.4 Quantum Computing

In this section, firstly, the state of the art of quantum computers will be explored with the most up-to-date information. Comparing different companies and the race to obtain quantum supremacy is crucial for this work since is a in-development topic that dinamically changes every day. Furthermore, concepts of Quantum Computing (QC) will be explained with examples especially the Gate-Based Quantum Computing methodology, which is one of the most accepted ones.

2.4.1 State of the Art

By the time this work was being developed, IBM had announced significant advances in QC. IBM Quantum Heron processor can now leverage Qiskit to perform algorithms in quantum circuits up to 5,000 qubits [66]. Qiskit is an open-source software development framework designed to work with IBM quantum computers. That being con-

sidered, it is a powerful toolkit for creating, manipulating, and running quantum programs, and it allows developers to explore and experiment with quantum computing. This empowers users to apply algorithms as Shor's or Groover's, which can possibilitate and make it feasible to intercept and decrypt IoT communication.

IBM is one of the most relevant players developing QC on a hardware and software level having a 1,121 qubits Quantum Computer. The roadmap, as referred to in Figure 2.3, has accomplished the plan with IBM Heron in 2024 and follows until 2033 when it is expected to reach quantum computers with hundreds of thousands of qubits [**?**]. Nevertheless, the current state of QC already shows great potential as the work from Kim et al. [77]. The paper showcases an experiment being performed in an IBM Heron processor running 50 times faster than a benchmark. Moreover, other functionalities as Generative AI capabilities are currently being developed and enhanced in IBM's quantum portfolio of products and services.

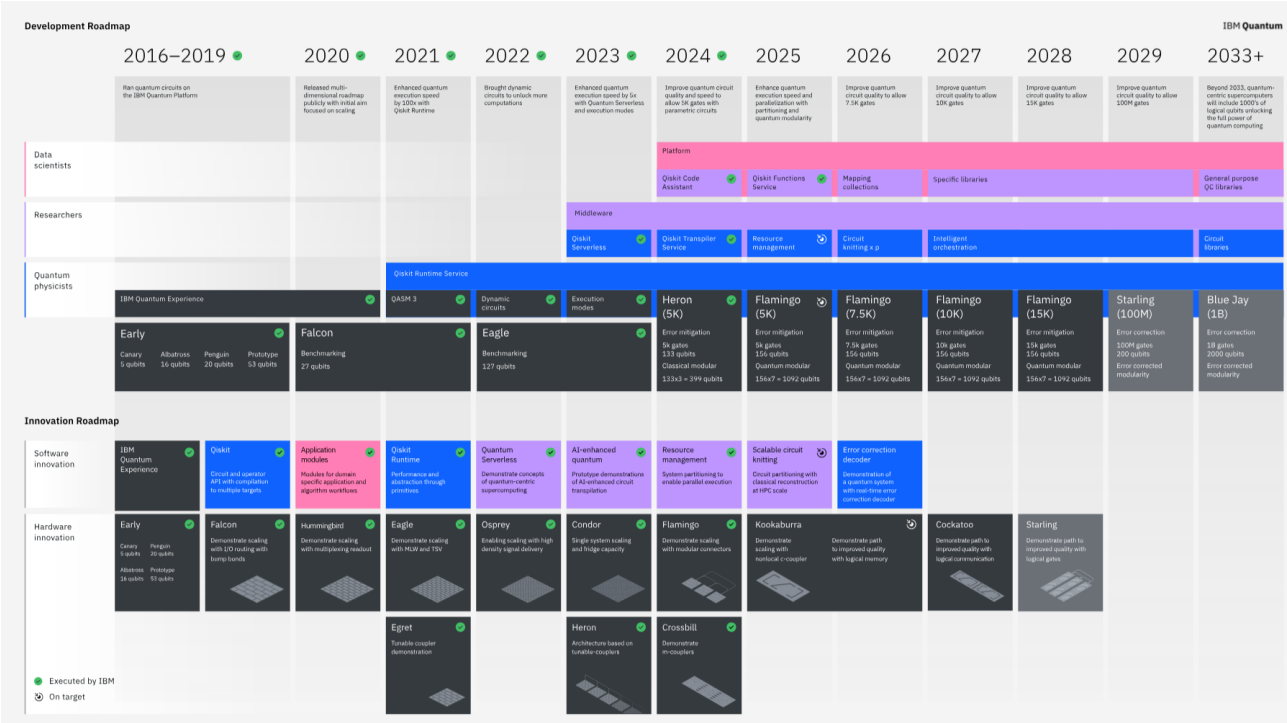

Figure 2.3 – IBM Roadmap [66]

Other players as AWS also have been developing quantum computers and quantum cloud services called Braket. Braket offers a single point of access to a variety of quantum computing technologies [16]. Among the hardware that Braket has access to includes the quantum computers from Rigetti, D-Wave, and IonQ. Rigetti being the more successful provider has accomplished reaching over 100 qubits in 2024.

Recently, Google has published a work introducing AlphaQubit, an AI-based decoder that identifies quantum computing errors with state-of-the-art accuracy [18]. Identifying and correcting errors in quantum computing is crucial due to the inherent fragility of quantum states. Qubits, are highly susceptible to disturbances from their environment,

leading to errors that can compromise computational accuracy. AlphaQubit has also shown that makes 6% fewer errors than tensor network methods [131]. This is a great breakthrough combining Machine Learning and QC to correct errors showing great potential with enhanced hardware resources. Significant challenges are still in development including scalability and speed.

Most recently, Microsoft has introduced the first quantum chip powered by a Topological Core architecture. It is expected to emulate a quantum computer and solve complex calculations [101]. Topoconductors enable a new way to develop quantum systems and can scale to the million figure the number of quibits.

The current landscape of QC shows remarkable advances coming from the greatest technology players advancing its products and services. This pushes the development of the area as new solutions are created and implemented. Having such industry giants participating in research and development as actively as we see in QC, accelerates and subsidizes the advances in the field.

While promising advances are already happening, there are still challenges to be overcome until reaching Quantum Supremacy. Among those challenges is the hardware that requires specific thermic conditions, insulation, space, and specific materials to perform measurements and qubits stabilization. Moreover, the capability to build and maintain qubits in quantum computers arises from their sensitivity to environmental disturbances and the challenges of ensuring coherence and scalability.

Nevertheless, it is possible to see and follow the advances being made in QC world. With enhancement of qubits from IBM or integrating Machine Learning as Google, continuous advances are being witnessed more frequently. The roadmaps for the next decade in QC advances are promising and fulfilling expectations.

2.4.2 Quantum Definitions

Quantum theory describes the behavior of matter and light in an atomic scale. At this level, classical physics is not applicable. The theory introduces concepts as wave-particle duality, quantization of energy, entanglement, superposition, and the uncertainty principle, fundamentally altering our understanding of physical phenomena. Quantum mechanics has become essential for explaining the properties of molecules, atoms, and their constituents electrons, protons, neutrons, and more esoteric particles like quarks [153]. The theory was considered revolutionary and, pioneered by Schrödinger, Heisenberg, Einstein, and Planck led to great inventions such as transistors, lasers, nuclear power, and superconductivity [68].

By definition, a quantum computer performs quantum computations. It manipulates the quantum states of qubits in a controlled way to perform algorithms. A quantum

computer leverages the capacity of adopting an arbitrary quantum state from an arbitrary input quantum state [68]. To compare classical to quantum computing, computational problems can be divided into three categories. The first category is composed of problems that cannot be solved more efficiently with quantum computers. The second category is formed by problems that can be solved with both computer types, but with less computational complexity when using a quantum computer. And, the third category is made of problems that cannot be solved by classical computers regardless of the amount of time, but are possible to solve with quantum computers. This is called “quantum supremacy” or ”quantum advantage” and it will have several critical applications in society such as cryptography and optimization [127] [121].

2.4.3 Quantum Superposition

Quantum superposition is the definition of electrons and photons acting as wave-like properties being able to combine among themselves in a superposed state. It is possible to understand its behavior compared to ocean waves, where they are formed by the movement of the water, while in quantum waves they are generated from mathematical equations describing the probability of an object to exist. The equations explain the probability of an electron being at movement at a certain speed or being in a certain location. Once this electron is in a superposition state, different states can have different outcomes. Each outcome has a probability of being observed. The electron can be in a superposition state in two velocities or at two places at the same time [43].

As complex as the concept is, in mathematical terms, superposition can be thought of as an equation that has more than one solution. As the equation, $x^2 = 4$ can have either 2 or -2 as a result. In this case, as Equation 2.1, the state of quantum superposition is described with $|\Psi\rangle$ representing the state of the qubit and $|0\rangle$ and $|1\rangle$ the states and, α and β are the probability amplitudes [130].

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.1}$$

The probability amplitudes α and β determine the probability of measuring the qubit in either state when a measurement is made. The superposition state is only maintained while the quantum system is observed. When it is measured, the superposition state is undone and is restored as one of the basis states. Superposition is fundamental to quantum systems, as it allows quantum parallelism. Since classical computers only allow 0 or 1 states, superposition opens the possibilities to more than one state at once, enhancing the possibilities of results and outcomes.

2.4.4 Quantum Entanglement

Quantum entanglement is a phenomenon where two or more quantum particles become connected in a way that the state of one particle instantaneously affects the state of the other. This happens regardless of the distance between them. For that reason, it is a relevant resource and concept for quantum communication, computing, and networks.[39].

A special condition is necessary for entanglement: a pair of electrons having opposite spins, as specified by the Pauli exclusion principle. Separating the pair of particles, even by a huge distance, the measurement of a particle's spin will automatically resolve itself on the other direction. This effect occurs instantaneously, apparently breaching the speed of light and the rules of relativity. Einstein referred to this phenomenon as "spooky action at a distance" [163].

This phenomenon has been studied by great scientists such as Heisenberg, Weyl, Schrödinger, Bell, and Einstein. With years of experimentation and philosophical discussions around entanglement, Feynman noted that the behavior of entangled photons could not be simulated in a classical computer [59]. By the time, in 1981 Feynman made this statement, quantum computers were not yet close to being a reality.

With quantum computing advances in research and development, entanglement becomes closer to being leveraged in quantum communication. Zou [176] mentions in her work that 92% of quantum communications consider entanglement as the core factor of industry competitiveness. The effect of fast transmission, unlimited capacity and absolute security corroborates for this property's relevance in communication.

2.4.5 Gate Based Quantum Computing

Quantum computing differs from classical computing due to quantum-mechanics properties that only a quantum computer is capable of reproduce [12]. QC leverages quantum mechanics phenomena, engineering, and computer science to solve complex problems [152]. In classical computing, a bit can either be 0 or 1. In the Gate-based QC model, the unit of memory is represented by a quantum bit (qubit). The qubit considers the state by a range of percentages to be 0 or 1 and is represented in a block sphere as Figure 2.4 shows. The poles $|0\rangle$ and $|1\rangle$ are equivalent to the classical bits. Meanwhile, if during quantum algorithms execution, the arrow points to a state between the classical bits in any direction, it indicates a state of superposition. This property allows the quantum computer access to multiple logical states at once [36]. Meaning that multiple states are happening at the same time in the block sphere.

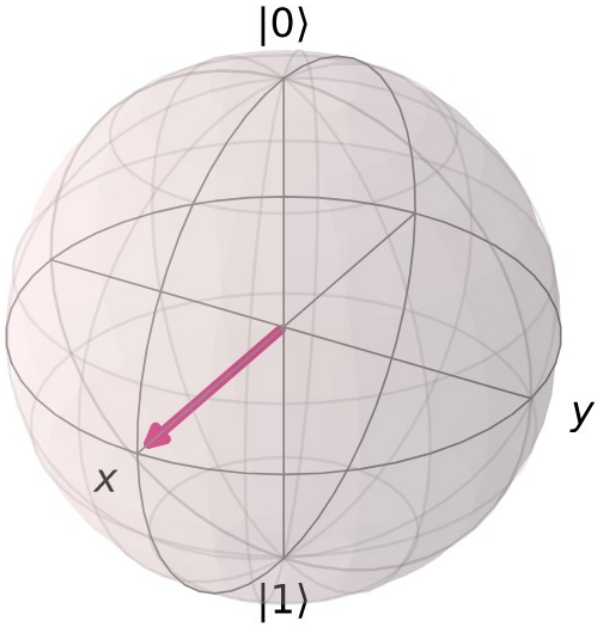

Figure 2.4 – Blocksphere Architecture Schema

Instead of considering the qubit as a 2-dimensional complex vector space on the unit sphere,the state of the qubit Ψ can be considered on the surface of a 3-dimensional unit sphere by introducing the two parameters α and β as expansion of Equation 2.1 showed in Equation 2.2. Considering that $|\alpha|^2 + |\beta|^2 = 1$ and the definition of complex vectors in polar coordinate systems [121] the general state of the qubit in a Blochsphere can be expressed as in Equation 2.3.

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{2.2}$$

$$|\Psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \tag{2.3}$$

Figure 2.5 exemplifies how a qubit state is represented in a Block sphere in 3 different states. In section a), θ= 30° and ϕ = 60°. The b) section of the Figure the state $|\Psi\rangle$ is described as $\theta = \phi$ = 90° and it is an equal superposition of $|0\rangle$ and $|1\rangle$. In c), $|\Psi\rangle = |1\rangle$ and since θ = 180°, then ϕ is not defined as Equation 2.4 explained.

Changing the state of the qubits required quantum gates. Among all the different types, the elements can be represented as vectors or matrices as bellow in Equation 2.4 [168]. Table 2.1 based on Bhat et al. [22] and Nourbakhsh et al. [121] works, summarizes some of the most common Quantum gates. The table includes the matrix representation, the input and expected output and, a brief description of the gate goal.

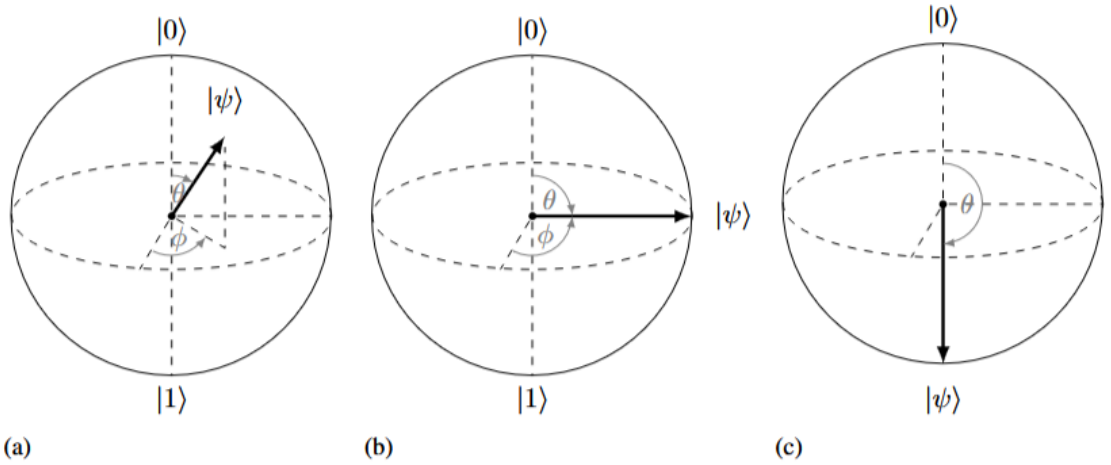

Figure 2.5 – States in a Blocksphere Example [121]

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{2.4}$$

Gate	Matrix	Input	Output	Description
Pauli-X	$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\|0\rangle$	$\|1\rangle$	Performs a rotation of π around the X axis. Also called the NOT gate.
Pauli-Y	$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	$\|0\rangle$	$i\|1\rangle$	Performs a rotation of π around the Y axis.
Pauli-Z	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\|0\rangle$	$\|0\rangle$	Performs a rotation of π around the Z axis.
Hadamard	$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\|0\rangle$	$\frac{1}{\sqrt{2}}(\|0\rangle + \|1\rangle)$	Performs a rotation of π around the Z axis, followed by a rotation of $\pi/2$ around the Y axis.
R_ϕ	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$	$\|0\rangle$	$\|0\rangle$	Movement along the latitudinal direction, by an angle of ϕ. Also called the shift gate, where ϕ is the phase shift.
I-gate	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\|0\rangle$	$\|0\rangle$	Does not change the state. Also called the identity or no-op gate.
S-gate	$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	$\|0\rangle$	$\|0\rangle$	Performs a phase shift of $\pi/2$.
T-gate	$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$	$\|0\rangle$	$\|0\rangle$	Performs a phase shift of $\pi/4$.

Table 2.1 – Single qubit gates with their matrices, their effect on the $|0\rangle$ state, and a general description.

As an example, the Hadamard gate is represented by the matrix in Equation 2.5 [168]. As the equation shows, the Hadamard matrix is its own inverse allowing an equal

superposition of the two basis states. This means that if the input state $|0\rangle$, the Hadamard gate transforms it into $\frac{|0\rangle+|1\rangle}{\sqrt{2}}$ and when the input state is $|1\rangle$, the Hadamard gate transforms it into $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$.

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}. \tag{2.5}$$

2.4.6 Quantum Computing Thoughts

Quantum computers are capable of guaranteeing secure transmission, ensuring speed and ability to store large amounts of information than classical computer [22]. With data increase in amount and size, classical computers have been demonstrating limitations whereas QC presents potential with its different approach for problem-solving [152]. A quantum algorithm, considering multiple logic states simultaneously can lower the time to solve highly complex problems [159]. One example is Shor's algorithm, which can solve factorization problems in polynomial time, representing a threat to RSA encryption.

2.5 Quantum Algorithms

In this section, two of the most relevant and famous quantum algorithms will be explored. Those algorithms are considered possible threats to cryptographic schemes. Between them is the Groover's and Shoor's algorithms.

2.5.1 Groover's Algorithm

Grover's Algorithm is a quantum search algorithm designed to identify a target entry in an unsorted database [86]. By leveraging from superposition, interference, and amplitude amplification quantum mechanic features, it finds solutions more efficiently than classical algorithms.

An example from Microsoft Azure Quantum [100] explains from a geometrical perspective how the algorithm works. Considering that $|\text{bad}\rangle$ is the superposition of all states that are not a solution and $|\text{good}\rangle$ the superposition state for all the states that are the solution for the search problem. States good and bad are orthogonal since it is not possible to be both at the same time as represented in Figure 2.6.

$$|\text{bad}\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle \qquad |\text{good}\rangle = \frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle$$

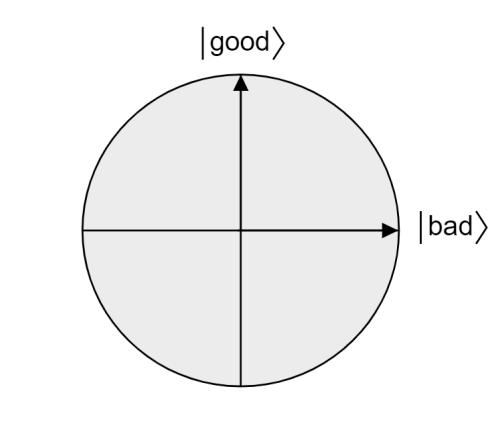

Figure 2.6 – Plane Example

Next, considering that $|\psi\rangle$ is a state composed by α and β that are factors that define how expressive are $|\text{good}\rangle$ and $|\text{bad}\rangle$ states. The reflection operator $R_{|\psi\rangle}$ can be geometrically interpreted as the reflection about the direction of any qubit on that state represented by $|\psi\rangle$.

$$|\psi\rangle = \alpha|\text{good}\rangle + \beta|\text{bad}\rangle \qquad R_{|\psi\rangle} = 2|\psi\rangle\langle\psi| - \mathcal{I}$$

Taking the orthogonal basis of $|\psi\rangle$ and its orthogonal complement $|\psi^{\perp}\rangle$, the $|\xi\rangle$ factor is calculated. And, when $R_{|\psi\rangle}$ inverts the orthogonal component $|\psi\rangle$, but leaves the original unchanged, creates the called reflection as observed in Figure 2.7.

$$|\xi\rangle = \mu|\psi\rangle + \nu|\psi^{\perp}\rangle \qquad R_{|\psi\rangle}|\xi\rangle = \mu|\psi\rangle - \nu|\psi^{\perp}\rangle$$

After its first application, all qubits are set to superposition state which, can be represented on the plane as Figure 2.8 demonstrates. The probability of obtaining a correct result is $|\langle\text{good}|\text{all}\rangle|^2 = \frac{M}{N}$.

$$|\text{all}\rangle = \sqrt{\frac{M}{N}}|\text{good}\rangle + \sqrt{\frac{N-M}{N}}|\text{bad}\rangle$$

The oracle O_f is written as a reflection for the $|\text{bad}\rangle$ axis. Correspondingly, O_0 is an inverted reflection of state $|0\rangle$. With this in mind, Groover's diffusion operation

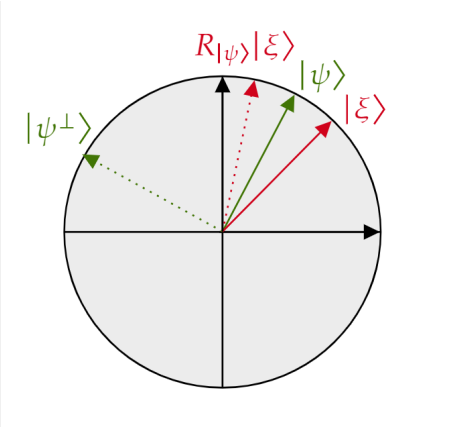

Figure 2.7 – Reflection Operator Example

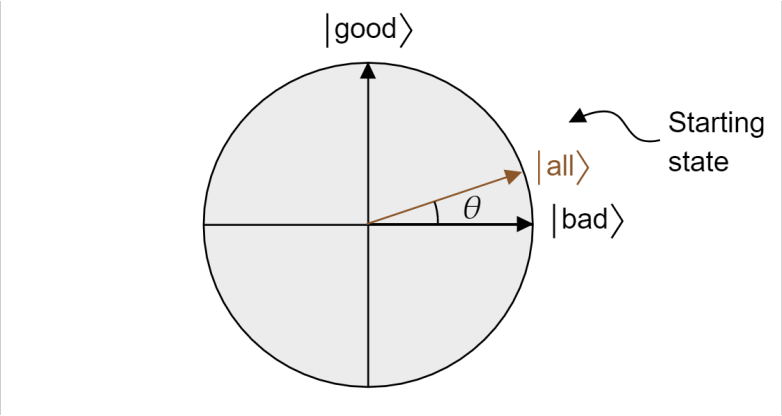

Figure 2.8 – Initial State

$-H^{\otimes n}O_0H^{\otimes n}$ is a reflection of $|\text{all}\rangle$ state and Figure 2.9 demonstrates each iteration as a reflection of $R_{|\text{bad}\rangle}$ and $R_{|\text{all}\rangle}$.

$$O_f = R_{|\text{bad}\rangle} = 2|\text{bad}\rangle\langle\text{bad}| - \mathcal{I}$$

$$O_0 = R_{|0\rangle} = -2|0\rangle\langle 0| + \mathcal{I}$$

$$-H^{\otimes n}O_0H^{\otimes n} = 2H^{\otimes n}|0\rangle\langle 0|H^{\otimes n} - H^{\otimes n}\mathcal{I}H^{\otimes n} = 2|\text{all}\rangle\langle\text{all}| - \mathcal{I} = R_{|\text{all}\rangle}$$

Groover iterations is a counterclockwise rotation of 2θ. To find the angle θ, the scalar product between $|\text{all}\rangle$ and $|\text{bad}\rangle$ is calculated. It is know that $\cos\theta = \langle\text{all}|\text{bad}\rangle$, therefore from the $|\text{all}\rangle$ definition, the angle can be calculated.

$$\theta = \arccos\left(\langle\text{all}|\text{bad}\rangle\right) = \arccos\left(\sqrt{\frac{N-M}{N}}\right)$$

The angle between the state of the register and $|\text{good}\rangle$ decreases on each iteration. This results on a higher probability of a valid result measured. The probability can be calculated using the angle θ as bellow shows.

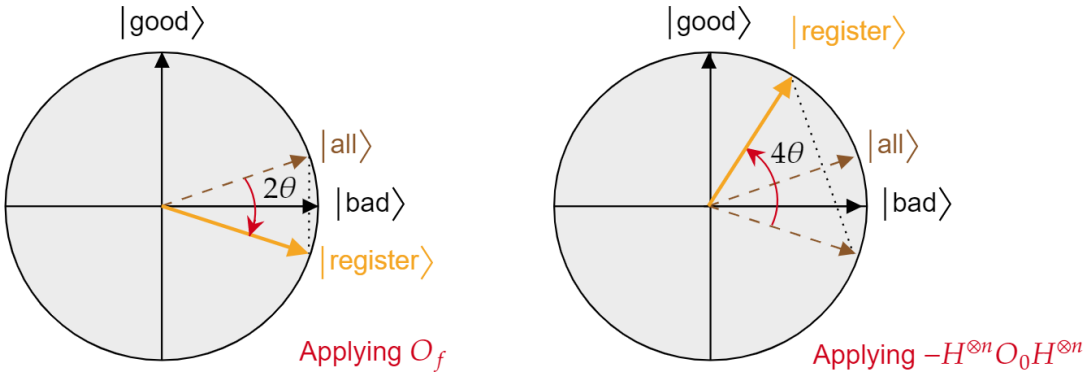

Figure 2.9 – Groover's Iteration

$$\gamma(k) = \frac{\pi}{2} - \theta - k2\theta = \frac{\pi}{2} - (2k+1)\theta$$

$$P(\text{success}) = \cos^2(\gamma(k)) = \sin^2\left[(2k+1)\arccos\left(\sqrt{\frac{N-M}{N}}\right)\right]$$

2.5.2 Shor's Algorithm

Peter Shor published his algorithm in the mid-1990s with the expectation that in the future when quantum computers are operational, his work could be implemented [150]. Several cryptosystems back when Shor worked on his algorithm and nowadays are based on factoring integers and finding discrete logarithms. Achieving these results is considered very difficult problem to solve through classical computing. Therefore, by the time Shor's algorithm was published, he considered that those problems would be possible to be solved using quantum computers.

The essence of Shor's Algorithm is the Quantum Fourier Transform (QFT), a quantum analog of the classical Fourier Transform. This transformation finds the period of a specific mathematical function related to the number being factored. When this period is found, the main factors can be extracted [67].

Current encryption schemes as RSA secure data relying on the very difficulty on factoring large numbers. Shor's algorithm affects RSA by effectively solving factoring problems. This poses an imminent threat to cryptographic protocols based on RSA and others.

For example, RSA security is based on $N = n \cdot q$ where n and q are large prime numbers and finding them is considered extremely difficult for classical computers. Fundamentally, Shor's algorithm starts with a guess on which numbers are n and q and im-

proves the guess iteratively. Consider the guess as g in Equation 2.6, thanks to Euclid's algorithm, finding only q, would be enough to reach N.

$$\begin{aligned} g &= t \times q \\ N &= n \times q \end{aligned} \tag{2.6}$$

The algorithm iterates with Equation 2.7, where the guess multiplies itself p enough times to be equal to the multiplication of a random number m to $N + 1$. Essentially, the algorithm tries to find p, which is the number of times to multiply the guess by itself to find N. In classical computers, it is a try-and-error process and may prevent it to be found in an adequate time.

$$\begin{aligned} g^p &= m \times N + 1 \\ g^p - 1 &= m \times N \\ (g^{p/2} - 1) \times (g^{p/2} + 1) &= m \times N \end{aligned} \tag{2.7}$$

Here is where quantum superposition comes to place. Shor's algorithm takes advantage of this property to find periodicity. In a quantum scenario represented by Equation 2.8, taking x as an input and raises it g, it is followed by a calculation of how bigger a multiple of N it is as remainder r.

$$\begin{aligned} g^x &= m \times N + r \\ g^{x+p} &= s \times N + r \end{aligned} \tag{2.8}$$

Superposition states are represented in Equation 2.9, where $|1\rangle$ is the first state. It can be derived to $|g^1\rangle$ and finally to $|g^1, r_a\rangle$ adding the remainder. The fact that simultaneous operations on a superposition state, allows to see the all possible solutions not requiring the interactivity and try and error from the classical resolution [3].

$$\begin{aligned} |1\rangle + |2\rangle + |3\rangle + ... \\ |g^1\rangle + |g^2\rangle + |g^3\rangle + ... \\ |g^1, r_a\rangle + |g^2, r_b\rangle + |g^3, r_c\rangle + ... \end{aligned} \tag{2.9}$$

Implementations of Shor's algorithm have been explored as from Amico et al. [10] work, where using IBM Q microchip. Despite the algorithm been feasible to be implemented and performed satisfactory results with experimental small numbers, Shor's

algorithm is still relying on qubits to surpass RSA. The biggest limitation that is preventing RSA to be broken is the advances in quantum hardware. This does not guarantee safe cryptography for undetermined time when considering large advances in quantum computers and number of qubits that dynamically and in real time shape quantum cryptography path.

2.6 LoRaWAN and Cryptography

In this section the current cryptographic state in LoRaWAN is described. First in Section 2.6.1, LoRaWAN architecture is described highlighting where the cryptographic protocols are applied. Next, in Section 2.6.2, the AES128 algorithm is explained in detail.

2.6.1 LoRaWan's Cryptography

LoRaWAN's cryptography protocol is based on two root keys *NwkSKey* and *AppSKey* [1]. When a device joins LoraWAN's network, the keys are generated and they are unique per device and section.

As Figure 2.10 shows, the Network Session Key *NwkSKey* secures the exchange between ED and the network and application service. The *NwkSKey* is a 128-bit key that ensures the integrity of the message using the AES Cipher-based Message Authentication Code (AES-CMAC). It is impossible to temper the encrypted message without *NwkSKey* because the Message Integrity Code (MIC) check will fail [112].

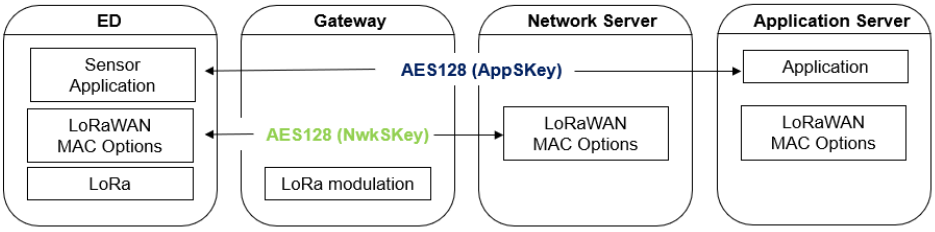

Figure 2.10 – LoRaWAN Cryptography

The Application Session Key *AppSKey* is responsible for the transferred data encryption. The data to be transferred or payload is encrypted with the 128-bit *AppSKey* using AES Counter Mode (AES-CTR) [52]. In the case of a message interception, it remains encrypted without the *AppSKey*

As Bonnetain et al. [23] presented in their work, AES-128 was shown to be vulnerable against Groover's algorithm. And, despite increasing the key size, Groover's algorithm performed in a quantum computer can reduce its security level [27]. Additionally,

increasing key sizes can affect the performance trade-off in the IoT environment and has to be considered. As concluded by Thaenkaew et al. [158], escalating AES' key size can occur in a 32% larger encryption time.

2.6.2 AES128

In 2000, NIST introduced the Rijndael block cipher family as the winner of the Advanced Encryption Standard (AES) competition [15]. Between this family is the AES128, which is the one implemented in LoRaWAN cryptographic protocol.

The 128 in the protocol's name means the length of data that is input and output in bits. This forms a block of 128 bits which composes the information transferred and ciphers [139]. Its security relies on the block cipher and its complex mathematical transformations.

The pseudo-code for CIPHER in the AES128 is presented in the Algorithm 2.1. The first step is to receive the 128-bit input *in*, the number of rounds *Nr*, and the key schedule *w* generated in an intermediate step in line 3. Next, in lines 4-8, the *state* is transformed by *Nr* applications of the round function in the loop. Within the loop, the function *SubBytes* from line 5 is an invertible, non-linear transformation of the state. It uses a substitution table predefined. The function *ShiftRows* in line 6 is responsible for cyclically shifting the last three rows of the state. Next in line 7, *MixColumns* multiplies each column from the *state* by a predefined matrix. The function in line 8 *AddRoundKey* combines a round key with the *state* applying the bitwise XOR operation. In lines 10-12 repeat the previous loop omitting *MixColumns* transoformation resulting in the final encrypted block as the ciphertext that is communicated to another device.

```
1:  input CIPHER(in, Nr, w)
2:  state ← in
3:  state ← AddRoundKey(state, w[0..3])
4:  for round from 1 to Nr − 1 do
5:      state ← SubBytes(state)
6:      state ← ShiftRows(state)
7:      state ← MixColumns(state)
8:      state ← AddRoundKey(state, w[4 × round..4 × round + 3])
9:  end for
10: state ← SubBytes(state)
11: state ← ShiftRows(state)
12: state ← AddRoundKey(state, w[4 × Nr..4 × Nr + 3])
13: return state
```

Algorithm 2.1 – Pseudocode for CIPHER()

After receiving the ciphertext, an intermediate parameter for decription is calculated. The function KeyExpansion (Algorithm 2.2) is a routine that has the key as an input and generates the *w*. The output is a linear array of words. Subsequently *w* is firstly filled by the loop in lines 3-6, and another loop takes place in lines 7-16. In this loop, a temporary variable *temp* is initiated in line 8 and two transformations take place in lines 10-12. Given an input word as a 4 bytes sequence $[a_0, a_1, a_2, a_3]$, *SubWord* assigns $([a_0, ..a_3]) = [SBox(a_0), ...Sbox(a_3)]$, SBox being a predefined reference table. And, *RotWord* transformation assings $[a_0, a_1, a_2, a_3] = [a_1, a_2, a_3, a_0]$. With those transformations, *temp* is defined and used to calculate *w* in line 14.

```
1:  procedure KeyExpansion(key)
2:  i ← 0
3:  while i ≤ Nk − 1 do
4:     w[i] ← key[4 ∗ i..4 ∗ i + 3]
5:     i ← i + 1
6:  end while
7:  while i ≤ 4 ∗ Nr + 3 do
8:     temp ← w[i − 1]
9:     if i  mod Nk = 0 then
10:       temp ← SubWord(RotWord(temp)) ⊕ Rcon[i/Nk]
11:    else if Nk > 6 and i  mod Nk = 4 then
12:       temp ← SubWord(temp)
13:    end if
14:    w[i] ← w[i − Nk] ⊕ temp
15:    i ← i + 1
16: end while
17: return w
18: end procedure
```

Algorithm 2.2 – Pseudocode for KeyExpansion()

The InvChipher algorithm (Algorithm 2.3) has the goal to decrypt the ciphertext previously received. The *state* receives the input as *Cipher()* and *AddRoundKey* function is applied in line 3. Following a loop that applies the inversion of functions from 2.1 like *InvShiftRows*, *InvSubBytes* and *InvMixColumns* in lines 4-9. *InvShiftRows* reverses the last three rows of the 4x4 byte state matrix are cyclically shifted to the right by a specific number of bytes. Each byte in the state matrix is replaced using an inverse predefined matrix S-box in *InvSubBytes*. It reverses the byte substitution applied during encryption. This substitution uses the inverse S-box to map each byte back to its original value. In *InvMixColumns*, each column of the state matrix is transformed by multiplying it with a fixed inverse matrix. This operation reverses the diffusion effect of *MixColumns*, spreading out the effects of each byte across the column but in a way that allows recovery of the original data. This is essential to restoring the plaintext during decryption by reversing column mixing. Finally, *AddRoundKey* results in the state same as in the encryption step as its own inverse.

1: **input** InvCipher(in, Nr, w)
2: $state \leftarrow in$
3: $state \leftarrow$ AddRoundKey($state, w[4 \times Nr..4 \times Nr + 3]$)
4: **for** $round$ from $Nr - 1$ downto 1 **do**
5: $state \leftarrow$ InvShiftRows($state$)
6: $state \leftarrow$ InvSubBytes($state$)
7: $state \leftarrow$ AddRoundKey($state, w[4 \times round..4 \times round + 3]$)
8: $state \leftarrow$ InvMixColumns($state$)
9: **end for**
10: $state \leftarrow$ InvShiftRows($state$)
11: $state \leftarrow$ InvSubBytes($state$)
12: $state \leftarrow$ AddRoundKey($state, w[0..3]$)
13: **return** $state$
14: **end procedure**

Algorithm 2.3 – Pseudocode for InvCipher()

The 128-bit key size adds another feature that allows a wide variety of possible keys, making brute-force attacks practically unfeasible. The avalanche effect phenomena explained by Rijmen et al. [139] allows a small change in the plaintext to impact the ciphertext thanks to *ShiftRows* and *MixColumns* algorithms.

A recent work from Mandal et al. [96] establishes that Grover's search algorithm reduces the security of symmetric key cryptosystems in half. This presents an imminent threat to all systems with AES128 encryption including LoRaWAN. However, as also stated in Jaques et al. [70] work, quantum computers need more advanced hardware resources to implement Groover's algorithm successfully. Nevertheless, the potential that attacks from quantum computers is already sufficient motivation for considering a post-quantum solution.

3. POST-QUANTUM CRYPTOGRAPHY

In this chapter, the definition of post-quantum-cryptography is explored starting with the basic concepts. The main key point for post-quantum cryptography containing mathematical problems that even quantum computers are not capable of solving are due Lattice Cryptography explained in section 3.0.2. More of it is explored in sections 3.0.3 and 3.0.4 where th Shortest Vector Problem and Learning with Errors variants are explained.

Starting with the development of this thesis and the pathway trailed following NIST finalist algorithms. Detailing the first post-quantum algorithm that was explored by us, the NTRU in section 3.0.5, followed by the Dilithium CRYSTALS in section 3.0.6. Next, the most up-to-date standard by NIST, the Module-Lattice Based Key-Encapsulation Mechanism in section 3.0.7 has its concept detailed. This algorithm was the one chosen for the final experiment in this work being the latest finalist and chosen standard by NIST.

3.0.1 PQC Definitions

Post-quantum cryptography (PQC) is the use of mathematical resources to prevent public-key cryptographic algorithms from quantum computer attacks [45]. It is essentially the adaptation of existing cryptographic algorithms and preparation in classical computers for future attacks from quantum computers. Post-quantum algorithms submissions to NIST (National Institute of Standards and Technology) are under revision and, therefore, there is no standard protocol yet. However, it is possible to categorize most algorithms into the families: lattice-based cryptography (LBC), multivariate, hash-based (signatures only), code-based and supersingular elliptic curve isogeny [45].

However, among the finalists of the standardization process, lattice-based algorithms represent half of the candidates [122]. A lattice is a set of points with a periodic structure. One of the lattice-based mathematical problems that makes the cryptosystem quantum-resistant is the Shortest Vector Problem (SVP). Only one vector calculated will reproduce the shortest vector [159]. The reason why SVP-based algorithms are quantum resistant is that, although there is a specific method to find a solution for this problem, there is no quantum algorithm that provides a shortcut to it yet [31].

Even though post-quantum algorithms perform satisfactorily, the implementation and integration in IoT devices is still a topic to be taken into consideration due to potential hardware adaptation requirements.

During the doctorate period, different post-quantum algorithms were studied and implemented. Throughout the program also, NIST have selected and improved different candidates regarding key encapsulation methods and digital signature algorithms.

3.0.2 Lattice Cryptography

Lattices are sets of points in *n* - dimensional spaces with a periodic structure [45]. Considering *B* as a set of independent *b* vectors, all integer linear combinations is called a Lattice [69]. Equivalently, a Lattice *L* can be generated from the matrix *B* containing all the *b* vectors as shown in Figure 3.1.

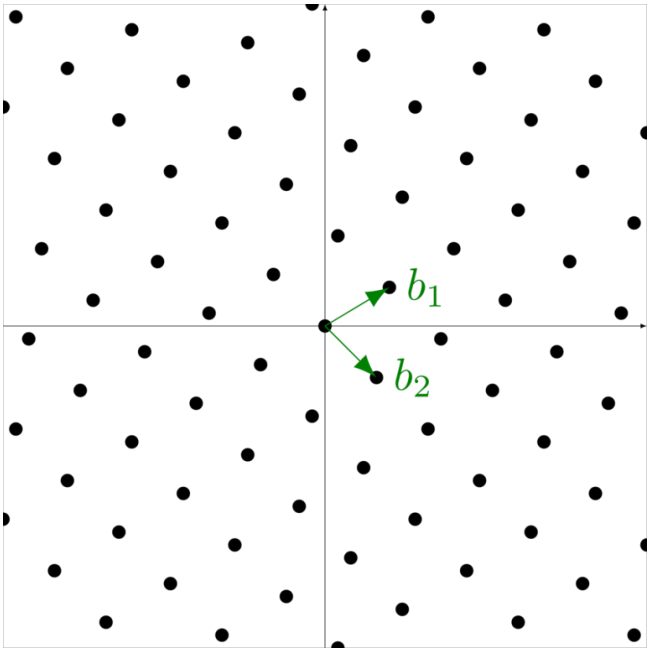

Figure 3.1 – A 2-dimensional lattice [69]

Due its mathematical complexity has shown robustness against attacks performed by quantum computers [37]. For that reason, lattice-based cryptography is one of the most promising candidates to act in a post-quantum world ensuring security in IoT communication.

The mathematical structure of the Lattices allows it to reach a complexity level that challenges classical computers to solve its problems. Furthermore, specific types of problems are expected to be equally difficult even for quantum computers performing Shor's algorithm, for example. Among these problems are the Shortest Vector Problem (SVP) and Learning with Errors (LWE).

3.0.3 Shortest Vector Problem (SVP)

The Shortest Vector Problem is based on given a basis $\mathbb{B}$ of a lattice *L*, to find the shortest non-zero vector ℓ. $\|\ell\| = \lambda(L)$. This problem raises the question of which point on the Lattice is closest to the origin. In other words, the difficulty is defining which combina-

tion of vectors is closest to the origin (0,0). Given the example of Figure 3.2, considering $b_1 = (29, 13)$ and $b_2 = (76, 38)$, the problem consists on finding the ideal configuration of a vector combining b_1 and b_2 that is closest to the origin (0,0).

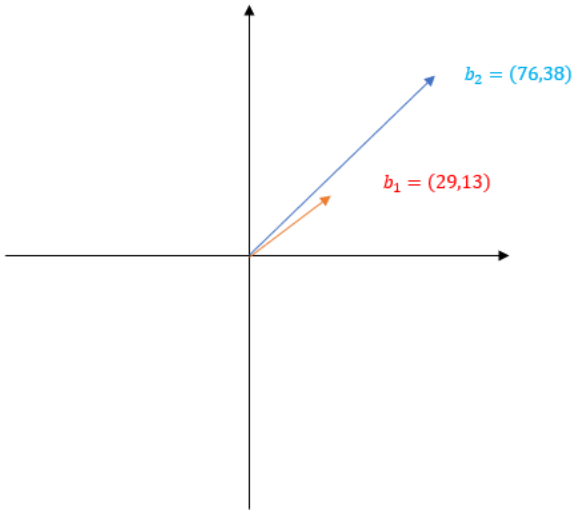

Figure 3.2 – SVP Example [64]

A possible combination is represented in Figure 3.3 a). By combining $3b_1 - b_2$ results in the point in (11,1). However, it can be even closer to the origin as section b) in the figure describes $8b_1 - 3b_2 = (4, -10)$. The problem is already difficult enough in 2D, but increasing the basis increases the dimensions and the difficulty. The SVP is an NP-hard problem, making it a robust candidate for cryptographic applications [111].

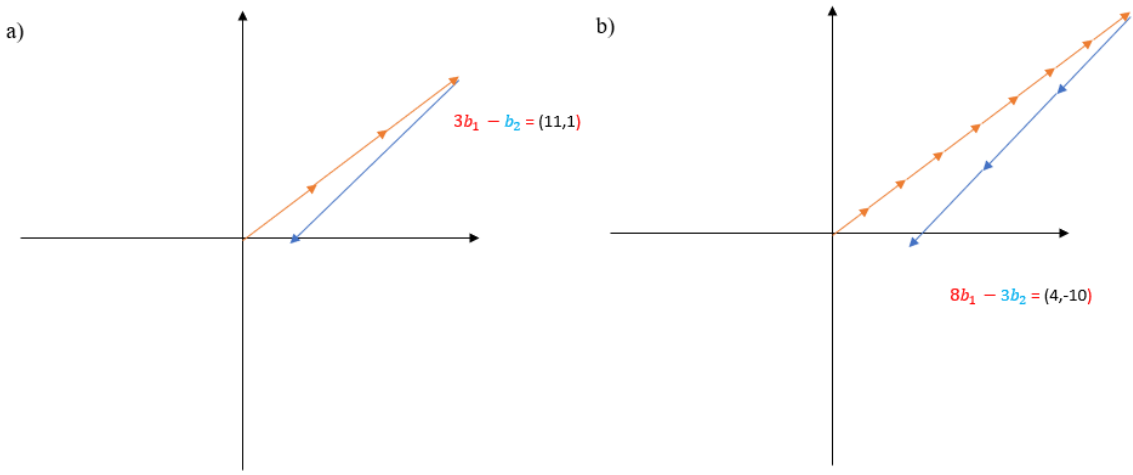

Figure 3.3 – SVP Example [64]

3.0.4 Learning with Errors (LWE) and Module Learning with Errors (MLWE) Problem

Learning with Errors problem has the goal to find an unknown secret $s \in \mathbb{Z}_q^n$. Meaning that the secret belongs to an universe with an n-dimensional vector over q, within 0 or 1, as bits. To find s, a list of equations with errors as Equation 3.1 shows.

$$\begin{aligned} \langle s, a_1 \rangle &\approx_\epsilon b_1 \,(\text{mod } q), \\ \langle s, a_2 \rangle &\approx_\epsilon b_2 \,(\text{mod } q), \\ &\ldots \end{aligned} \tag{3.1}$$

Given that a_i belong to the distribution on $\mathbb{Z}_2^n$, $\langle s, a_i \rangle = \sum_j s_j (a_i)_j$ and b being the inner product modulo 2 of s and a_i plus a noise vector ϵ [134]. Bellow, there is an example of LWE given a sequence of approximate random linear equations on s as Equation 3.2 from Houston-Edwards [65]. Considering that Alice is a device that is willing to share safely a message to Bob, being another device, the example follows.

$$\begin{aligned} 77x + 7y + 28z + 23w &= 2859 \\ 21x + 19y + 30z + 48w &= 3508 \\ 4x + 24y + 33z + 38w &= 3848 \\ 8x + 20y + 84z + 61w &= 6225 \\ 6x + 53y + 1z + 86w &= 4886 \\ 42x + 86y + 31z + 8w &= 9062 \\ 5x + 24y + 79z + 27w &= 6103 \\ 16x + 7y + 35z + 21w &= 2589 \\ 56x + 18y + 25z + 58w &= 3576 \\ 4x + 55y + 73z + 13w &= 8265 \end{aligned} \tag{3.2}$$

It is simple to find the s secret vector using a Gaussian elimination, resulting in (10, 82, 50, 5) [135]. Thus, the addition of errors ϵ increases the mathematical difficulty. By adding errors, as bellow, the party that has only access to the public information would not know how to differentiate what is the error from the result. This transforms the system in overdetermined, having more equations than variables. This makes impractical for a classic computer to solve using the same strategy as the first example without errors.

$$
\begin{aligned}
77x + 7y + 28z + 23w &= 2859 \quad + -3 \\
21x + 19y + 30z + 48w &= 3508 \quad +2 \\
4x + 24y + 33z + 38w &= 3848 \quad + -1 \\
8x + 20y + 84z + 61w &= 6225 \quad +0 \\
6x + 53y + 1z + 86w &= 4886 \quad +4 \\
42x + 86y + 31z + 8w &= 9062 \quad + -1 \\
5x + 24y + 79z + 27w &= 6103 \quad + -2 \\
16x + 7y + 35z + 21w &= 2589 \quad +2 \\
56x + 18y + 25z + 58w &= 3576 \quad +0 \\
4x + 55y + 73z + 13w &= 8265 \quad + -1
\end{aligned}
\tag{3.3}
$$

The next layer of the method is adding Modular Arithmetic by introducing a mod calculation. In this case mod 89 is added to every equation creating the MLWE problem. As, for example, the first equation that equals to 2859, when mod 89 is applied, gives 11, which subtracted by the error 3, results in 8.

$$
\begin{aligned}
77x + 7y + 28z + 23w &= 11 \,(\text{mod } 89) = 8 \\
21x + 19y + 30z + 48w &= 37 \,(\text{mod } 89) = 39 \\
4x + 24y + 33z + 38w &= 21 \,(\text{mod } 89) = 20 \\
8x + 20y + 84z + 61w &= 84 \,(\text{mod } 89) = 84 \\
6x + 53y + 1z + 86w &= 80 \,(\text{mod } 89) = 84 \\
42x + 86y + 31z + 8w &= 73 \,(\text{mod } 89) = 72 \\
5x + 24y + 79z + 27w &= 51 \,(\text{mod } 89) = 49 \\
16x + 7y + 35z + 21w &= 8 \,(\text{mod } 89) = 10 \\
56x + 18y + 25z + 58w &= 16 \,(\text{mod } 89) = 16 \\
4x + 55y + 73z + 13w &= 77 \,(\text{mod } 89) = 76
\end{aligned}
\tag{3.4}
$$

In an example of a message either 0 or 1 to be transmitted, Bob considers arbitrarily the equations since they are public information. Bob then sums them resulting in $30x + 67y + 53z + 24w = 19 \,(\text{mod } 89)$. Since Bob only has the public information, his side cannot differentiate what is the error portion of the equation. Therefore, if Bob wants to send 0, he sends the result equation including the error. In the scenario that he wants to send 1, the equation is $30x + 67y + 53z + 24w = 19 + 44 \,(\text{mod } 89) = 63$. The 44 added is the rounded down result of half of 89. Beause Alice has the secret key, she can split the 63 into the actual solution and the encoded bit. In Modular Arithmetic, the encoded bit will include the error, but Alice is capable of interpret if it is either 0 or 1 depending on how far this bit is from 0 or 44.

$$\begin{aligned} 21x + 19y + 30z + 48w &= 37 + 2 \,(\text{mod } 89) \\ 4x + 24y + 33z + 38w &= 21 + -1 \,(\text{mod } 89) \\ 5x + 24y + 79z + 27w &= 51 + -2 \,(\text{mod } 89) \\ \hline 30x + 67y + 53z + 24w &= 20 + -1 \,(\text{mod } 89) \end{aligned} \tag{3.5}$$

It is trivial for Alice to differentiate the actual solution from the encoded bit. However, for a third party without the secret key in classical computing, it is nearly impossible [84].

3.0.5 NTRU

The NTRU is a lattice-based cryptosystem. This means that the algorithm is based in a lattice field problem considered hard to solve. The algorithm is composed by a key generation, encryption and decryption phases. The parameters used to implement the NTRU are N which is the polynomial degree of the algorithm's operations and must be prime. The other inputed parameters p and q must also be prime and satisfy Equation 3.6. The equation requires that the greatest common divisor between N and q, and p and q has to be equal to 1 [63].

$$gcd(N, q) = gcd(p, q) = 1 \tag{3.6}$$

For the key generation process, a trusted party of the communication chooses the public parameters (N,p,q). In the next phase, the private key f and public key h are computed. First, two sets of random polynomials are generated f and g. It is considered that f has an inverse modulo q and modulo p [62]. A multiplication represented by star * is defined by the discrete convolution product of two vectors and a dot represents a standard multiplication [56].

$$F_p * f = 1(mod\ p) \tag{3.7}$$

$$F_q * f = 1(mod\ q) \tag{3.8}$$

Thus, the public key is obtained by equation 3.9. Meanwhile, the private keys are stored.

$$h(x) = F_q(x) * g(x) \quad mod\ q \tag{3.9}$$

In the encryption phase, a plain text message m, is converted to a string of ones and zeros using ASCII [35]. The string needs to be encrypted to a polynomial whose coefficients are between $\frac{-1}{2}p$ and $\frac{1}{2}p$ [63]. For that, a random polynomial r is generated and inputted in formula 3.10 [62].

$$e = p.r * h + m \quad (mod\ q) \tag{3.10}$$

The cryptographed message e is received and needs to be decrypted using the private key f [62].The first step to decrypt is represented in 3.11.

$$a = f * e\ (mod\ q) \tag{3.11}$$

Considering that the receiver already calculated F_p in equation 3.7, b can be computed in 3.12. Consequently, b will result in the original message m.

$$b = F_p * a\ (mod\ p) \tag{3.12}$$

To prove that decryption works, the following proof method is executed [62]. Considering the ciphertext polynomial e and substituting 3.10 in 3.11, results in equation 3.13.

$$a = f * e = f * p.r * h\ f * m\ (mod\ q) \tag{3.13}$$

Plugging equation 3.9 to 3.13, the following results are obtained in 3.14.

$$a = f * p.r * F_q * g\ + f * m\ (mod\ q) \tag{3.14}$$

And finally, substituting equation 3.8 in 3.14 the result obtained is between *-q/2* and *q/2* as equation 3.15 shows. The interpretation of this is that when the message is decrypted and the coefficients of *f * e mod q* are reduced, the original polynomial is recovered and thus, the message m [62].

$$a = p * r * g * f * m\ (mod\ q) \tag{3.15}$$

There are lattice-based attacks currently known. But, for that, a matrix N by N is necessary to be computed. As N increases, so does the time to compute the matrix. Thus, it becomes infeasible to break. For a $N = 500$ is estimated 8.4 years to break and, even more when increasing p and q [35].

3.0.6 Dilithium Crystals

Dilithium Crystals is a digital signature cryptography method that relies its security on the hardness of finding short vectors in lattices [41]. The approach is divided in a key generation, signing procedure and a verification phase. Starting with the key generation phase, Algorithm 3.1 represents how to obtain the public and secret keys [138]. Firstly, the algorithm generates a matrix A with k x l dimensions. In the next step, s and e, which are sampled random key vectors are generated. Finally, b is computed resulting the public key pk and secret key s as output.

1: **Input:** none
2: Generate $A \in R_q^{kxl}$
3: Samples $s \in R_q^l$
4: Samples $e \in R_q^q$
5: Calculate $b = As + e$
6: **Output:** public key $pk = (A, b)$, secret key s

Algorithm 3.1 – Dilithium Crystals Key Generation

The signing process presented in Algorithm 3.2 is a probabilistic one. In the algorithm's step 2, a random vector $y \in R_q^l$ is sampled. In the following step, a given Ay vector of polynomials is rounded and stored as w. In step 4, c is formed by hashing the message m and w. The hash function H maps an input with coefficients in {-1,0,1}. Since z depends on s, it can lead to security vulnerability [138]. Thus, Dilithium uses a technique called rejection sampling approach to remove the statistical dependencies between z and the secret key s. Meaning that in case z is rejected, the algorithm starts from the beginning.

1: **Input:** public key $pk = Ab$, secret key $s, messagem \in \{0, 1\}$ Until z is valid:
2: Sample $y \in R_q^l$
3: Calculate $w = Ay$
4: Calculate $c = H((w), M) \in B_{60}$
5: Calculate $z = y + cs$
6: Output: signature $\sigma = (z, c)$

Algorithm 3.2 – Dilithium Crystals Signing Process

To validate the signature, the verification method is described in 3.3. The recovered w' is used to recalculate c'. Subsequently, c' is compared with c.

1: **Input:** public key $pk = Ab$, secret key $s, messagem \in \{0,1\}$, signature $\sigma = (z, c)$
2: Calculate $w' = round(Az - bc)$
3: Calculate $c' = H(m||w')$
4: Output: valid if $c = c'$, else invalid

Algorithm 3.3 – Dilithium Verification

3.0.7 Module-Lattice Based Key-Encapsulation Mechanism Standard

The latest NIST publishing defines a derivation of CRYSTALS-Kyber submission as Module-Lattice Based Key-Encapsulation Mechanism as standard [115]. Key encapsulation mechanism (KEM) allows two parties to share a secret key over a public channel [117]. The ML-KEM standard is considered secure by NIST against attacks from quantum computers due the computational difficulty of the Module Learning with Errors (MLWE) problem.

The premise of the mechanism is that two communicating party share a secret key that is computed jointly and unknown to adversaries. The shared secret key can be used with symmetric cryptography to perform encryption and authentication of messages.

The first step of KEM starts with the decapsulation and encapsulation key being generated by the one of the communicating party (Alice, for example). Alice maintains the decapsulation key private and shares the encapsulation key with another device (Bob). Bob uses the encapsulation key to generate a shared secret key along with a cryptographic message, called ciphertext. Bob shares with Alice the ciphertext, which is decrypted by the decapsulation key that computes another copy of the shared secret key. Alice and Bob aim to conclude the process with both secret keys outputs equal ensuring the communication's secrecy [73].

ML-KEM Key Generation

Firstly, as mentioned before, the ML-KEM starts with the key generation. The key generation is activated on Alice's side as Algorithm 3.4 describes. The goal of the algorithm is to generate an ecapsulation and decapsulation keys *ek* and *dk*, respectively. Some parameters are defined: *n, q, k,* η_1,η_2, d_u and d_υ considering that *n* is 256, *q* is 3329 and, the remaining vary. The first two steps sample *d* and *z* as 32 random bytes seeds. The seeds are the input for the $ML-KEN.KeyGen\_internal$ Algorithm 3.5 which output the keys.

The intermediate step of calculating ek_{PKE}, dk_{PKE} is through $K-PKE.KeyGen$ Algorithm 3.6. The ek_{PKE} output will be used as the encapsulation key and can be shared publicly. However, the decription key dk_{PKE} and seed must remain secret to be used for the decapsulation phase. The algorithm starts using the G Equation 3.16 to create

1: $d \leftarrow \mathbb{B}^{32}$
2: $z \leftarrow \mathbb{B}^{32}$
3: **if** d == *null* **OR** d == *null* **THEN**
4: **return** $\perp$
5: **end if**
6: (ek, dk) $\leftarrow ML - KEM.KeyGen\_internal(d, z)$
7: **return** (ek, dk)

Algorithm 3.4 – ML-KEM Key Generation

1: $(ek_{PKE}, dk_{PKE}) \leftarrow$ K-PKE.KeyGen(d)
2: ek$\leftarrow ek_{PKE}$
3: dk$\leftarrow (dk_{PKE}||ek||H(ek)||z)$
4: **return** (ek, dk)

Algorithm 3.5 – $ML - KEM.KeyGen\_internal$

two expanded pseudorandom 32-byte seeds ρ and σ and initializing N as 0. Rows 3 to 7 generate the matrix $\hat{A}$ using Sample NTT Algorithm 3.7. The indices i and j are bytes 33 and 34 of 64 bytes from the both 32 bytes input combination.

Subsequently, the vector s is generated as a set of secret variables. Incrementing N at every round, the vector is generated by the $SamplePolyCDB_{\eta_1}$ 3.8 that samples the vector from a Centered Biominal Distribution (CBD). This distribution is parameterized by the pseudorandom function PRF 3.18. Similarly, the noise vector e is computed using different seeds since N was incremented already. Running NTT k times to generate $\hat{s}$ and $\hat{e}$ allows $\hat{t}$ to be computed. Finally, running ByteEncode Algorithm 3.11 for k times, ek_{PKE} and dk_{PKE} are calculated.

Equation 3.16 outputs two 32-byte vectors. The has function SHA3-512 is an XOF with one variable-length input and one variable-length output as byte arrays described by the SHA-3 NIST standard [42]. While function H in Equation 3.17 has a one variable-length as input hashes it with SHA-256.

$$\text{G(c):= SHA3-512}(c) \tag{3.16}$$

$$H(s) := \text{SHA3-256}(s) \tag{3.17}$$

The SampleNTT Algorithm 3.7 has a 32-byte seed input B and, as a result it outputs a pseudorandom array. The array $\hat{a}$ contains the coefficients of the NTT. Its starts

1: **input** d
2: $(\rho, \sigma) \leftarrow G(d||k)$
3: $N \leftarrow 0$
4: **for** $i \leftarrow 0; i < k; i++$
5: **for** $j \leftarrow 0; j < k; j++$
6: $\hat{\mathbf{A}}[i, j] \leftarrow SampleNTT(\rho||j||i)$
7: **end for**
8: **end for**
9: **for** $(i \leftarrow 0; i < k; i++)$
10: $s[i] \leftarrow SamplePolyCDB_{\eta_1}(PRF_{\eta_1}(\sigma, N))$
11: $N \leftarrow N + 1$
12: **end for**
13: **for** $(i \leftarrow 0; i < k; i++)$
14: $e[i] \leftarrow SamplePolyCDB_{eta1}(PRF_{\eta 1}(\sigma, N))$
15: $N \leftarrow N + 1$
16: **end for**
17: $\hat{s} \leftarrow NTT(s)$
18: $\hat{e} \leftarrow NTT(e)$
19: $\hat{t} \leftarrow \hat{A}.\hat{s} + \hat{e}$
20: $ek_{PKE} \leftarrow ByteEncode_{12}(\hat{t})||\rho$
21: $dk_{PKE} \leftarrow ByteEncode_{12}(\hat{s})||\rho$
22: **return** (ek_{PKE}, dk_{PKE})

Algorithm 3.6 – K-PKE.KeyGen

by converting a seed with two indexing bytes into a polynomial in rows 1 and 2. A XOF funtion is an Extendable-Output Function and compares to a hash with the exception of being able to create an arbitrary output length. In the standard, XOF is used as an API for SHAKE128 algorithm. The in row 1, the algorithm is initialized with *ctx* followed by the Absorb function that updates the context.

Then, the algorithm loops until the array $\hat{a}$ is composed of 256 elements. *Squeeze* extracts a 3-byte array output bytes that were produced in the *Absorb* phase and again updates the context *ctx* and *C*. The 3 bytes used in *Squeeze* represent 24 bits and are distributed in d_1 and d_2. In case d_1 is higher than the parameter q established as 3329, then it is stored in the array $\hat{a}$. Subsequently d_2 is checked according to the constrains and if satisfied, is stored in the array $\hat{a}$.

The next sampling algorithm used in the Key Generation 3.6 is the SamplePolyCBD described in 3.8. It generates a pseudorandom polynomial sampling coefficients from a Centered Binominal Distribution (CDB). The CDB is responsible for generating noise for lattice-based cryptography. Firstly, BytestoBits Algorithm 3.9 receives an array *B* pf 64-bytes and converts it to a bit-array. Next, there is a loop foro 256 coefficients where *x* is computed by summing η bits from the bit-array *b*. A similar calculation denotes *y* with the difference of the position in *b* being $2i\eta + \eta + j$. Finally the array *f* is populated at each iteration.

$ctx \leftarrow XOF.Init()$
$ctx \leftarrow XOF.Absorb(ctx, B)$
$j \leftarrow 0$
while $j < 256$ **do**
    $(ctx, C) \leftarrow XOF.Squeeze(ctx, 3)$
    $d_1 \leftarrow C[0] + 256.(C[1] mod 16)$
    $d_2 \leftarrow [C[1]/16] + 16.C[2]$
    if $d_1 < q$ **then**
        $\hat{a} \leftarrow d_1$
        $j \leftarrow j + 1$
    end if
    if $d_2 < q$ and $j < 256$ **then**
        $\hat{a}[j] \leftarrow d_2$
        $j \leftarrow j + 1$
    end if
end while
return $\hat{a}$

Algorithm 3.7 – SampleNTT

$b \leftarrow BytesToBits(B)$
for $(i \leftarrow 0; i < 256; i++)$
    $x \leftarrow \sum_{j \leftarrow 0}^{\eta - 1} b[2i\eta + j]$
    $y \leftarrow \sum_{j \leftarrow 0}^{\eta - 1} b[2i\eta + \eta + j]$
    $f[i] \leftarrow x - y$ mod q
end for
return f

Algorithm 3.8 – SamplePolyCBD

The algorithm BytestoBits represented in 3.9 converts byte arrays to bits arrays with each segment of eight its representing a byte. Firstly the input array B is copied to C to ensure the original remains the same. Now the function loops for each byte and updates the array b and C. After satisfying the loop conditions, b is computed and returned as the bit array.

The Pseudorandom function (PRF) represented by Equation 3.18 considers the fixed η parameter $\in \{2, 3\}$, a 32-byte input called s and an 1-byte input b. The SHAKE256 hash function [42] is applied to the concatenation between the seed s, the byte b and produces an output of $8.64.\eta$ bits or $64.\eta$ bytes.

$$\begin{aligned} PRF &: \{2,3\} x \mathbb{B}^{32} x \mathbb{B} \rightarrow \mathbb{B}^{64\eta} \\ PRF_\eta(s, b) &:= SHAKE256(s||b, 8.64.\eta) \end{aligned} \tag{3.18}$$

1: $C \leftarrow B$
2: **for** $(i \leftarrow 0; j < l; i++)$
3: **for** $(j \leftarrow 0; j < 8; j++)$
4: $b[8i + j] \leftarrow C[i] \bmod 2$
5: $C[i] \leftarrow [C[i]/2]$
6: **end for**
7: **end for**
8: **return** b

Algorithm 3.9 – BytestoBits

The Number-Theoretic Transformation (NTT) represented by Algorithm 3.10 has the goal of improving the efficiency of multiplication .... The input is the polynomial coefficient f and it first assigns $\hat{f}$ as a copy and i as 1. The *for* loop is initialized assigning len to 128, with the condition where $len \geq$ than 2 and updates len dividing it by 2. The subsequent loop is initialized with *start* as 128 while it is smaller or equal to 256 and it is appended its value with two times len from the iteration. Next, *zeta* is assigned to ζ, which in this case it is 17, a primitive n-th root of unique modulo q. Subsequently, i is incremented by 1. Nested, another loop is initialized with j being assigned to *start*, while it is less than *start* + len and having it added 1 at every round. With that, once the loop conditions are satisfied, $\hat{f}$ is derived.

1: $\hat{f} \leftarrow f$
2: $i \leftarrow 1$
3: **for** $(len \leftarrow 128; len \geq 2; len \leftarrow len/2)$
4: **for** $(start \leftarrow 0; start < 256 : start \leftarrow start + 2.len)$
5: $zeta \leftarrow \zeta^{BitRev_7} \bmod q$
6: $i \leftarrow i + 1$
7: **for** $(j \leftarrow start; j < start + len; j++)$
8: $t \leftarrow$ zeta.f[j+len]
9: $\hat{f}[j + len] \leftarrow \hat{f}[j] - t$
10: $\hat{f} \leftarrow \hat{f}[j] + t$
11: **end for**
12: **end for**
13: **end for**
14: **return** $\hat{f}$

Algorithm 3.10 – NTT

The ByteENcode Algorithm 3.11 encodes an array of d-bit integers into a byte array for $1 \leq d \leq 12$ as B. It initializes a loop with 0 as i while it is less than 256 and adds 1 at every round. Next, i is assigned to the element i from the input array F and another loop starts with j as the index. By every iteration, b and a are derived and by applying Algorithm BitsToBytes 3.12, B results in the final vector.

1: **for** $i \leftarrow 0; i < 256; i++$)
2: $a \leftarrow F[i]$
3: **for** $(j \leftarrow 0; j < d; j++)$
4: $b[i.d + j] \leftarrow a \bmod 2$
5: $a \leftarrow (a - b[i.d + j])/2$
6: **end for**
7: **end for**
8: $B \leftarrow BitsToBytes(b)$
9: **return** B

Algorithm 3.11 – ByteEncode

BitstoBytes Algorithm 3.12 converts a bit array in an array of bytes. The algorithm iterates while the index i is smaller than 8 times the array's length l and edits the one of the 8 slots of the output array every round with the result of the calculation in row 3.

1: $B \leftarrow (0, ..., 0)$
2: **for** $(i \leftarrow 0; i < 8l; i++)$
3: $B[[i/8]] \leftarrow B[[i/8]] + b[i].2^{imod8}$
4: **end for**
5: **return** B

Algorithm 3.12 – BitsToBytes

Encapsulation

The Encapsulation Algorithm represented in (Algorithm 3.13) receives a key ek as input, generates randomness and outputs a ciphertext and a shared key. Firstly, it assigns $\mathbb{B}^{32}$ to a 32 random bytes m array. Subsequently, it runs ML-KEM.ENCAPS_INTERNAL (Algorithm 3.14) that accepts an encapsulation key and a random byte array and outputs the ciphertext and the shared key. The functions G 3.16 and H 3.17 derive the shared secret key K and the randomness r.

1: $m \leftarrow \mathbb{B}^{32}$
2: **if** $m ==$ NULL **then**
3: **return** $\perp$
4: **end if**
5: $(K, c) \leftarrow$ ML-KEM.ENCAPS_INTERNAL(ek, m)
6: **return** (K, c)

Algorithm 3.13 – ML-KEM.Encaps

The Algorithm 3.15 receives the encryption key ek_{pke} and a 32-byte plaintext m with a randomness r to produce the ciphertext c. Firstly, the algorithm extracts vector $\hat{t}$ using the ByteDecode Algorithm 3.17 and extracts the 32-byte seed ρ from ek_{pke}. Similar

1: $(K, r) \leftarrow G(m||H(ek))$
2: $c \leftarrow$ K-PKE.Encrypt(ek, m, r)
3: **return** (K, c)

Algorithm 3.14 – ML-KEM.Encaps_internal

to Algorithm 3.6, the matrix $\hat{A}$ is generated from the NTT Algorithm 3.7. The vector $\hat{t}$ and the matrix $\hat{A}$ should be the same as calculated in K-PKE.KeyGen 3.6. Next, the vectors y, e_1 as the noise terms, and e_2 are sampled with Algorithms 3.8 and 3.18. The message m is then encoded in μ so it can update v and finally generate the final ciphertext c.

1: $N \leftarrow 0$
2: $\hat{t} \leftarrow \text{ByteDecode}_{12}(ek_{\text{PKE}}[0 : 384k])$
3: $\rho \leftarrow ek_{\text{PKE}}[384k : 384k + 32]$
4: **for** $i \leftarrow 0$ to k **do**
5: **for** $j \leftarrow 0$ to k **do**
6: $\hat{A}[i, j] \leftarrow \text{SampleNTT}(\rho||j||i)$
7: **end for**
8: **end for**
9: **for** $i \leftarrow 0$ to k **do**
10: $y[i] \leftarrow \text{SamplePolyCBD}_{n_1}(\text{PRF}_{n_1}(r, N))$
11: $N \leftarrow N + 1$
12: **end for**
13: **for** $i \leftarrow 0$ to k **do**
14: $e_1[i] \leftarrow \text{SamplePolyCBD}_{n_2}(\text{PRF}_{n_2}(r, N))$
15: $N \leftarrow N + 1$
16: **end for**
17: $e_2 \leftarrow \text{SamplePolyCBD}_{n_2}(\text{PRF}_{n_2}(r, N))$
18: $\hat{y} \leftarrow \text{NTT}(y)$
19: $u \leftarrow \text{NTT}^{-1}(\hat{A}^\top \odot \hat{y}) + e_1$
20: $\mu \leftarrow \text{Decompress}_{\ell_0}(\text{ByteDecode}(m))$
21: $v \leftarrow \text{NTT}^{-1}(\hat{t}^\top \odot \hat{y}) + e_2 + \mu$
22: $c_1 \leftarrow \text{ByteEncode}_{d_u}(\text{Compress}_{d_u}(u))$
23: $c_2 \leftarrow \text{ByteEncode}_{d_v}(\text{Compress}_{d_v}(v))$
24: **return** $c \leftarrow (c_1||c_2)$

Algorithm 3.15 – K-PKE.Encrypt(ek_{PKE}, m, r)

ByteDecode 3.17 is the inverse of ByteEncode 3.11 and has the goal of converting an array of bytes to an array of integers modulo m. It stats by decoding a byte array to an array of integers between 1 and 12. Subsequently, it converts bytes to bits using Algorithm 3.9 and for each index between 0 and 256, calculates the array F.

Decapsulation

The decapsulation algorithm 3.18 receives the decapsulation key dk and the ciphertext c as input and outputs the shared secret key K' within Algorithm ML-KEMDecaps_internal

$f \leftarrow \hat{f}$
$i \leftarrow 127$
for len $\leftarrow 2$ to 128; len $\leftarrow 2 \cdot$ len **do**
  for start $\leftarrow 0$ to 256; start $\leftarrow$ start $+ 2 \cdot$ len **do**
    zeta $\leftarrow \zeta_{\text{BitRev}_q(i)} \mod q$
    $i \leftarrow i - 1$
    for $j \leftarrow$ start; $j <$ start + len; $j++$ **do**
      $t \leftarrow f[j]$
      $f[j] \leftarrow t + f[j + \text{len}]$
      $f[j + \text{len}] \leftarrow \text{zeta} \cdot (f[j + \text{len}] - t)$
    end for
  end for
end for
$f \leftarrow f \cdot 3303 \mod q$
return f

Algorithm 3.16 – NTT^{-1}

$b \leftarrow \text{BytesToBits}(B)$
for $i \leftarrow 0$ to 256 **do**
  $F[i] \leftarrow \sum_{j=0}^{d-1} b[i \cdot d + j] \cdot 2^j \mod m$
end for
return F

Algorithm 3.17 – $\text{ByteDecode}_d(B)$

3.19. It needs to extract the decapsulation key dk_{PKE} and the encryption key ek_{PKE}. The hash h and the variable z as an implicit rejection value are extracted in sequence. The ciphertext is then decrypted by K-PKE.Decrypt 3.20 generating the message m'. The plaintext message m' is encrypted again in c' and gets a candidate of shared key K' and the encryption randomness r'. The new encrypted value c' is verified if it is equal to c, if not, it rejects the shared key K'.

$K' \leftarrow$ *ML-KEM.Decaps_internal*(dk, c)
return K'

Algorithm 3.18 – ML-KEM.Decaps(dk, c)

$dk_{PKE} \leftarrow dk[0 : 384k]$
$ek_{PKE} \leftarrow dk[384k : 768k + 32]$
$h \leftarrow dk[768k + 32 : 768k + 64]$
$z \leftarrow dk[768k + 64 : 768k + 96]$
$m' \leftarrow K\text{-}PKE.Decrypt(dk_{PKE}, c)$
$(K', r') \leftarrow G(m'||h)$
$K \leftarrow J(z||c)$
$c' \leftarrow K\text{-}PKE.Encrypt(ek_{PKE}, m', r')$
if $c \neq c'$ **then**
    $K' \leftarrow \perp$
end if
return K'

Algorithm 3.19 – ML-KEM.Decaps_internal(dk, c)

K-PKEDecypt Algorithm 3.20 receives a decryption key dk_{PKE} and the ciphertext c and outputs the plaintext m. Decompress 3.20 and 3.17 run for k times to compute u', v', and $\tilde{s}$. To compute w, NTT 3.10 runs k times and NTT^{-1} 3.16 runs once. Finally, the plaintext message m is decoded from v'

$c_1 \leftarrow c[0 : 32d_uk]$
$c_2 \leftarrow c[32d_uk : 32(d_uk + d_v)]$
$u' \leftarrow Decompress_{d_u}\left(ByteDecode_{d_u}(c_1)\right)$
$v' \leftarrow Decompress_{d_v}\left(ByteDecode_{d_v}(c_2)\right)$
$\tilde{s} \leftarrow ByteDecode_{12}(\mathrm{dk_{PKE}})$
$w \leftarrow v' - NTT^{-1}\left(\tilde{\mathbf{s}}^{\top} \circ NTT(u')\right)$
$m \leftarrow ByteEncode_1\left(Compress_1(w)\right)$
return m

Algorithm 3.20 – K-PKE.Decrypt($\mathrm{dk_{PKE}}$, c)

For compressing and decompressing, Equations 3.19 and 3.20 considers the parameter $q = 3329$ and the bit length of 12. Therefore, for $d <$12, it computes y as described bellow.

$$\begin{aligned} \mathrm{Compress}_d &: \mathbb{Z}_d \rightarrow \mathbb{Z}_{2^d} \\ x &\mapsto \left\lfloor \left(\frac{2q}{d}\right) \cdot x \right\rfloor mod 2^d \end{aligned} \tag{3.19}$$

$$\begin{aligned} \text{Decompress}_d : \mathbb{Z}_{2^d} &\to \mathbb{Z}_q \\ y &\mapsto \left\lfloor \left(\frac{q}{2^d}\right) \cdot y \right\rceil \end{aligned} \tag{3.20}$$

Since 2016, NIST has been conducting submissions for post-quantum algoithms candidates for standard [114]. This competition involved multiple algorithms among different cryprographic protocol types as digital signature and key encapsulation. In August 2024, NIST has released the final set of encryption tools that are designed to enforce cryptography against quantum computing attacks [136].

Among the selected algorithms, KEM is one of NIST's choices of post-quantum algorithms included in their standards for public-key cryptography. KEM encapsulates the key so it can be shared in insecure channels and ensures the key's secrecy in case of one of the parties being compromised. Considering that KEM is an asymmetric crypto algorithm, it presents an adequate fit for a LoRaWAN use case.

4. LITERATURE REVIEW AND RELATED WORKS

4.1 Systematic Literature Review

A preliminary systematic mapping literature review was conducted to identify advances in PQC algorithms as well as PQC applied to IoT devices, especially in LoRaWAN. Reviewing was essential to understand concepts and applications of PQC to identify the intersection with IoT. Moreover, in this literature review, identify gaps and opportunities to develop our work. Our initial objectives with this mapping were:

- Identify the state-of-the-art on PQC related to IoT devices and LoRAWAN.
- Understand how PQC performs and behaves with IoT devices and LoRaWAN,
- Identify challenges of implementation PQC in an IoT environment.

For the systematic mapping, the coverage of the result can be determined by frequency of publications [124]. This methodology uses research questions to answer after the literature review. For this systematic mapping, the questions are:

- How necessary is PQC in IoT devices and LoRaWAN?
- What are the main challenges for the implementation of PQC in IoT in practice?

The scientific databases used to research were IEEExplore, Science Direct, ACM Digital Library and, arXiv. And, the main keywords researched were *IoT and LoraWAN, post-quantum-cryptography, LoRaWAN cryptography, KEM post-quantum, KEM and LoRaWAN, PQC and IoT*.

The inclusion and exclusion criteria for selecting the scientific papers were the following bellow.

Inclusion:

- Needs to be in English.
- Must mention one of the NIST finalist algorithms, in case it is an implementation paper.
- Must be relevant for IoT implementation.

Exclusion:

- Research words are not in abstract.

- Study is not in the computer science or physics areas or is focused on a very specific use case.
- Study is from before 2020, unless is about quantum concepts.

The string research resulted in 3,566 scientific papers. Research strings as IoT and LoRaWAN and post-quantum-cryptography resulted in many mathematical theoretical works and general applications. This increased the first papers selection and, after considering the inclusion and exclusion criteria, 105 scientific works were considered for this thesis.

4.1.1 Research Questions Analysis

After conducting the selected studies readings 4.1.2, we were able to have a deeper understanding of PQC and its potential and limitations for IoT devices. According to our review, the answers to the research questions are:

- How necessary is PQC in IoT devices?

Post-quantum cryptography is the use of classical computation algorithms to proportionate resistance against cyber-attacks from quantum computers. This protection is pertinent to IoT devices to provide secure communication and privacy. The focus of the researchers is to identify potential problems and opportunities that a quantum computer is not possible to solve [166].

As the number of IoT devices increases drastically, so does the necessity to improve their security [30]. This is corroborated by research made by DigiCert [91] which asked professionals about the realization of quantum computing threat among organizations. As can be seen in Figure 4.1, the market is aware and can recognize the dimension of the quantum menace. Among the current methods to secure IoT devices are symmetric and asymmetric cryptosystems. Symmetric cryptography is composed of two parties sharing the same secret key that is used to encrypt and decrypt the message. In asymmetric cryptography, there are public and a secret key. The public key can be used by any party to encrypt a message, but only the one with the secret key can decrypt the message.

One of the most used symmetric cryptographic algorithms for IoT is the AES. Nowadays, the most consolidated method to break this algorithm is through brute force. However, the Grover’s algorithm with enough computational power as a quantum computer can provide is capable of speeding up this process and breaking AES [30]. Grover’s algorithm also threatens asymmetrical algorithms such as SHA-256. Besides that, public key algorithms as RSA, ECDSA, elliptic curve DH, and digital signature algorithms, which

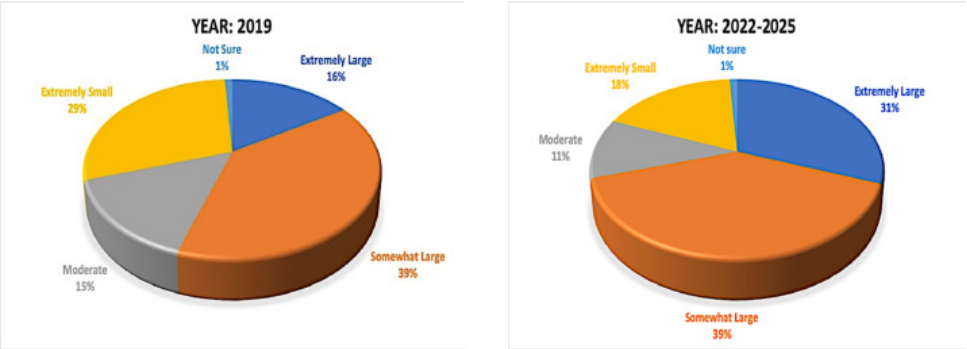

Figure 4.1 – Realization of quantum computing threat among organizations [91]

are based on integer factorization, are vulnerable against quantum computers. Shor's algorithm performed in a quantum computer is capable of breaking consolidated algorithms as mentioned before in polynomial time [45].

As IoT devices can be used to control critical infrastructure and important systems in finance, transportation, smart grid, or healthcare, imperative to ensure their integrity. Despite skepticism from researchers related to when quantum computers will have enough qubits to perform representative results or be available commercially, there have been significant advances in the field as presented by Yan et al.[167]. Therefore, based on the knowledge acquired with the mapping review, we consider the implementation PQC in IoT not only a necessity but an urgency.

- What are the main challenges for the implementation of PQC in IoT in practice?

The IoT environment itself is already composed of many challenges as Figure 4.2 presents. Those challenges prevent manufacturers to ensure full security in the IoT devices [91]. Considering this and the advances in quantum computing, it is critical to understand the suitability of PQC for the IoT environment and how effective it is against quantum attacks.

Incontestably, according to the literature mapping, one of the greatest challenges to implementing PQC in IoT devices is related to the performance [126], [30], [75], [25]. IoT devices can have constrained memory due to its size or simply because the architecture is not favorable to PQC. To have a new cryptosystem running in an existing architecture might require adaptation and trade-offs either from the software of hardware and might not be ideal from the business point of view.

In a memory-constrained environment, PQC can present expensive trade-offs in terms of performance [25]. One example is signature operations cryptography algorithms cause excessively time-consuming computations compared with the RSA protocol [4]. Additionally, PQC often requires large key sizes and will demand significant key management from manufacturers.

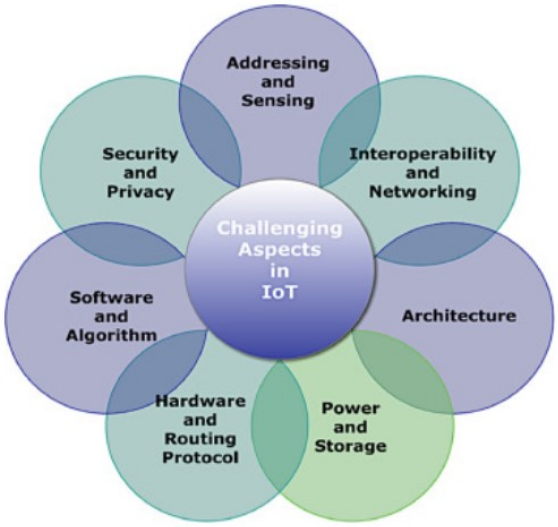

Figure 4.2 – Challenging aspects in IoT ecosystem [91]

There is still uncertainty of which strategy or algorithm is the most suitable for each case. Every case is unique and depends not only on the PQC algorithm but also on the available hardware. Thus, the first step is to understand PQC limitations and assess if the hardware is needed to be modified or adapted. For that reason, standardization is a prerequisite for widespread PQC deployment since all parties need to use the same cryptographic system [20]. Overall, PQC is still an emergent technology that evolves rapidly. It is soon to affirm the most adequate solution for each case because each use case will have a specific approach.

- What are the highest security threats currently found in LoRaWAN?

Currently, there are several vulnerabilities in LoRaWAN's protocol. Replay attacks, impersonation and eavesdropping are some examples. Those issues mostly from keys integrity since LoRaWAN relies on AES, a symmetric cryptographic scheme [38]. This requires a stronger key management practices, improved authentication, enhanced message encryption and, above all, lightweight solutions [120].

- Are quantum-algorithms a menace for LoRaWAN's security and can they leverage its current vulnerabilities?

As mentioned before, AES is a possible target against Groover's algorithm once quantum computers have sufficient qubits [30]. Even though enhancing AES to 256, for example would increase its security, it is still no guaranteed that it would provide enough strength against quantum-algorithms [53], [**?**].

4.1.2 Mapping Results

After implementing the inclusion and exclusion criteria added to reading abstracts, the original paper retrieval resulted in a total of 105 scientific publications. The result of the mapping is shown in Table 4.1 distributed by database. The final paper classifications were used to answer our research questions.

We first divided the selected studies in groups to understand PQC concepts and Security Related applications in IoT and LoRaWAN. This allowed answering the questions because the first group clarified how advanced quantum computers are and how close we are to a real threat. Moreover, understanding the theory of PQC has guided us to understand different algorithms and how they are resistant to quantum computers. This was critical to understand how necessary PQC is for IoT since highlighted how vulnerable current cryptographic methods are against quantum attacks.

The second classification group helped us to answer what are the main challenges to implementing PQC in IoT devices and LoRaWAN. The studied papers show that performance is a massive issue for PQC in constrained devices. Either current IoT hardware needs to be adapted or trade-offs are necessary in a software level in many cases. Understanding these challenges, showed that PQC still has a long way to be commercially applicable and needs more development.

From the selected papers, a partial selection was chosen to be studied in depth in Section 4.2. These publications were the ones who presented more potential to answer our research questions but also gaps that we could fulfill with this work. As concluded in the literature review, there are few researches involving LoRaWAN and PQC and enough evidence that proves its necessity.

4.2 Related Works

De Moraes and Conceição [38] reviewed LoRaWAN security in their work. The authors found 19 security weakenesess and vulnerabilities in LoRaWAN. One of the key findings is that currently, the message authentication is made only on the network server, leaving data vulnerable between network and application. In case an attacker resends a previous join request, the lack of encryption can result in a replay attack. Besides that, the session keys, being derived from the network server, can be vulnerable in case of the server being compromised. The greatest takeaway from the survey is that LoRaWAN relies on AES encryption, where the weaknesses consists on the key management, authentication, and message integrity. All of the concerns raised by the authors need significan improvements, specially on the AES encryption dependency.

Database	PQC Concepts	Security Related applications in IoT and LoRaWAN
IEEE explorer	[160], [167], [45], [54], [161], [102], [87], [174], [8], [95], [171]	[146], [88], [9], [6], [109],[1], [126], [80] , [75], [4], [157], [48], [125], [56], [141], [132], [164], [44], [74], [165] ,[149], [106], [110], [147], [172], [123], [145], [14]
Science Direct	[30], [41], [105], [20], [173], [78], [41], [94], [138], [61], [35], [12], [51], [81], [166]	[25], [37], [144], [137], [159], [2], [49], [91]
ACM Digital Library	[103], [32], [82], [21], [17], [83], [97]	[72], [93], [133], [175], [33], [108], [34], [151], [151], [98], [107], [60]
arXiv	[11],[89], [29], [81], [19]	[5], [148], [38], [13], [170], [40], [162], [119]

Table 4.1 – Database Mapping

According to Sanchez et al. [146], LoRaWAN's security scheme based on pre-shared cryptographic material lacks flexibility when a key update is necessary. Thus, the authors evaluated the key management and proposed the alternative scheme Ephemeral Diffie–Hellman Over COSE (EDHOC) to enhance security. Among vulnerabilities considered by the authors are static session keys that can be stolen, lack of forward secrecy if an attacker compromises the session key, the whole communication history is compromised and, the manual key configuration that requires administrative intervention. With that in mind, the study proposed the implementation of EDHOC scheme and evaluate the message overhead, the processing efficienty and energy consumption in a LoRAWAN environment. It was concluded that EDHOC outperforms other schemes in securtiy and computational cost. However, message overhead is lower than the benchmark which constrains its application in LoRaWAN. To be suitable for higher Spreading Factors (SF) implementation, additional adjustments are necessary as compression or segmentation mechanisms. Due the transmission limitations found during the experiments, the security improvements are still not optimal for use case applications. Moreover, even ECDH is not resitant againgst Shor's algorithm being officially deprecated by 2030 and disallowed after 2035 by NIST, requiring a post-quantum solution [58].

Lino et al. [88] ran a comparative analysis of the impact of cryptography in IoT LoRa. Considering the limited computational resources in IoT, the optimal cryptographic method applied is critical to ensure security without excessive performance costs. The experiments were implemented in a 32-bit ARM (Raspberry Pi 3) microcontrollers and an Arduino MKR1300 to assess real-world performance. Several classical cryptographic algorithms were implemented, among them AES, SHA and RSA evaluating the execution time, code efficiency, and communication impact. It was assessed that the performance varies

according to the algorithm, hardware platform and data size. The cryptographic algorithms affected transmission time in LoRa networks, potentially doubling the time needed to deliver sensor data. Balancing the performance in IoT environment and offering equivalent security is crucial to validate the implementation of PQC in LoRaWAN networks.

To help organizations monitor and track their physical assets, Amelia et al. [9] developed a LoRa-based tracking system using AES256. The goal is to implement the system to protect transmitted data from eavesdropping or unauthorized access. The paper evaluates the system's tracking accuracy, communication range, and data security. The tracking system was implemented in a transmitter and receiver connected to LoRa signal and added AES256 encryption to prevent malicious parties to intercept communication and have access to the communication. Even though the system was successfully implemented proving that AES256 enhances indeed the security of the communication, it is still not quantum-resistant as Groover's algorithm could reduce AES security and potentially break the cryptography.

A study by Allamanda et al [6] develops a LoRa-based monitoring system integrating AES256 encryption and SHA256 hashing. A distributed system of sensors collect real-time data and communicates through LoRa signal to a central station. SHA256 and AES256 maintain data integrity against eavesdropping attacks. As the data sent is encoded, at the step when the message is intercepted, the attacker cannot access the data since it is encrypted. However, it was observed that there was an increase in processing time after AES was implemented. Additionally, flash memory and RAM were also evaluated and AES implementation increases its usage.

Marlind and Butun [109] propose a new activation method for LoRaWAN end devices using Public Key Cryptography (PKC). The main goal is to enhance the security of key distribution by introducing a method called Public Key Over the Air Activation (PK-OTAA). The research investigates the feasibility of using PKC for root key assignment in LoRaWAN devices while considering energy consumption, security, and practicality. The authors highlight security issues in current LoRaWAN architecture as the dependency on AES with pre-configured root keys. If root keys are compromised, all past and future messages could be intercepted and makes the system vulnerable against replay attacks. The proposed method ensures only authorized devices can generate the same session key without the key distribution with the Elliptic Curve Diffie-Hellman (ECDH) algorithm. Despite increased in power usage demand, the algorithm successfully worked in the setup. Even though the solution enhances the security preventing the long-term key exposure, the performance is definitely a step back and needs to be considered. Besides that, ECDH is not resistant against quantum-algorithms.

In a study by Abboud et al. [1], a security enhancement in LoRaWAN security is proposed by increasing the key size in AES256-based cryptography. The authors goal is to evaluate the trade-offs considering performance, energy consumption and security in the

network. The study implemented AES256 at LoRaWAN's MAC layer. The main metrics measured were the security level, data payload size, total transmission time, packet loss rate, network throughput and, energy consumption. It was observed that by improving the key size from 128 bits to 256, it improves the resistance against brute-force attacks and bit-flipping. The transmission time, packet loss rate, energy usage and network throughput increased its figures. Therefore, the authors highlighted some trade-offs to be considered, for example that due the power consumption increase, makes the key enhancement less ideal for battery-constrained IoT applications. Thus, it is proposed optimization in the algorithm to be a fit for IoT devices. Additionally to that, the key size increase in AES does not necessarily turns the solution quantum-resistant due its vulnerability against Groover's algorithm. Considering that a potential simpler solution to enhance LoRaWAN security does not instantly transforms it quantum-resistant and neither offers appropriate performance, it highlights that other approches or different algorithms would be required.

The study from Bavdekar et al. [19] explores the impact of quantum computing on classical cryptographic schemes. It was analysed that classical symmetric cryptography shows vulnerabilities against Groover's algorithm and asymmetric cryptography to Shor's. Lattice-based cryptography like Kyber or Dilithium, for example are the most promising alternatives due its security, scalability and efficiency. The authors claim that QC poses a major threat to existing cryptosystems, making post-quantum cryptography essential for secure communication.

Challenges of integrating post-quantum cryptographic (PQC) algorithms into existing Internet protocols was researched by Müller et al. [108]. Domain Name System Security Extensions (DNSSEC) can ensure authenticity and integrity of DNS responses, but its current public-key cryptography can be vulnerable against quantum-computing algorithms. Considering that PQC algorithms have larger key sizes, it could lead to DNS message fragmentation and a drop in performance. A proposal by the authors is an adaptation on DNSSEC to handle larger keys and signatures and enhancing its validation processes to maintain performance standards.

The paper by Septien-Hernandez et al. [148] investigates PQC algorithms and their compatibility with IoT. It was considered Kyber512, LightSaber, NTRU, and FrodoKEM algorithms and their performance, security, and feasibility in resource-constrained IoT environments was assessed from a simulation. Sensor nodes were implemented with Arduino Nano with a LoRa communication module and a Raspberry Pi 3B+ acting as an intermediary between the sensor nodes and the cloud. It was concluded that Kyber512 and LightSaber provided the best trade-offs between security, speed, and memory usage. NTRU presented higher energy consumption and Frodo was disconsidered for IoT due the highest overhead recorded. This study was a milestone in our research because it proved that PQC is indeed suitable for IoT devices with the correct algorithm. Additionally, Kyber

was considered a good direction of PQC to approach presenting a strong security level maintaining efficiency for IoT environment.

Barbosa et al. [17] introduce EasyPQC, a toll to verify PQC constructions and evaluate cryptographic schemes. The main contribution is the probabilistic relational Hoare logic (pRHL) framework. It supports reasoning about quantum adversaries and introduces new proof techniques in the Quantum Random Oracle Model (QROM). This allows EasyPQC to verify PQC security. It was concluded that EasyPQC extension successfully verifies post-quantum security and the proposed approach is feasible. However further optimizations are necessary for large-scale cryptographic systems.

Asif [13] surveyed lattice-based cryptographic algorithms in a PQC environment for IoT. The author stated that the rise of QC threatens traditional cryptographic methods as RSA and elliptic curve cryptgroahy (ECC). To protect communication safety in IoT, Latice-based cryptgraphy (LBC) emerges as a proissing solution. The paper provides an overview on mathematical foundation of classical cryptography and LBC. To evaluate the feasibility of PQC implementation, a lightweight Lattice-based cryptography simulation was performed in a controled environment with different PQC algorithms. The results proved that Lattice-based PQC provides a secure alternative to classical cryptosystems. The performance varies according to the algorithm having nuances between energy consumption, memory usage and time of execution requiring a trade off depending on the use case. It is highlighted the need for further research into scalable and optimized implementations of LBC for different IoT applicabilities. Quantum computation advances push the advances and implementation of PQC. However, to guarantee scalability and flexibility to optimize efficiency at the same security level as traditional cryptography, more development and research with different use cases is required.

Ye et al. [170] have implemented PQC algorithms referencing PQClean library to improve a lattice-based processor for IoT systems. In this study, the authors had the goal to simulate CRYSTALS-Kyber (KEM) and CRYSTALS-Dilithium deployment. A customized RISC-V-based processor with a specialized SIMD architecture optimized for lattice-based cryptographic operations was proposed by the authors. The authors have validated the implementation comparing with PQClean's expected outputs in a non optimized processor. The processor performance achieved a considerable speedup over the baseline considered and, power efficiency was considered suitable for battery-power IoT devices. While, Kyber and Dilithium showed optimized performance, other algorithms could be explored as ML-KEM, for example. Additionally, the authors relate as future works the implementation in real-world scenario and extend the design in a hybrid PQC and classical cryptography environment. This work highlights that PQC integration in IoT is feasible and can perform satisfactorily when comparing with classical crytpgraphy. With certain limitations and lack of use cases, the topic proves itself again that is worth to be studied and explored more.

Along the development of this doctorate work, different approaches were dynamically adjusted according with NIST resolutions, new findings and general development of the topic. This resulted in three scientific publications, having the first one published in World Forum on IoT in 2022 [46]. The paper proposed a post-quantum cryptographic method for securing autonomous vehicle (AV) communication in IoT environments. As AVs rely on constant data exchange with other vehicles, pedestrians, and infrastructure, they are vulnerable to cyberattacks, particularly from emerging quantum computing threats. The NTRU (N-th Degree Truncated Polynomial Ring) lattice-based cryptographic algorithm was investigated as a quantum-resistant solution for secure vehicle-to-network (V2N) communication over a 5G network. The NTRU algorithm was implemented in an edge computing architecture, where AVs encrypt traffic data before transmitting it to a 5G base station. Although NTRU effectively secures IoT-based autonomous vehicle communications against quantum threats, the algorithm was not selected as a finalist in the NIST competition for PQC standards.

Next, it was published in the Advanced Information Networking and Applications (AINA) conference an enhancement in 5G-AKA protocol with Dilithium Crystals digital signature Method [140]. The 5G-AKA protocol relies on Elliptic Curve Cryptography (ECC), which is vulnerable to Shor's Algorithm. If an attacker intercepts the long-term secret key (K), they can impersonate users and compromise communication. dditionally, Grover's Algorithm can reduce the security level, making brute-force attacks feasible in quantum computing. Therefore, it was proposed integrating Dilithium Crystals in the protocol to ensure a quantum-resistance communication. The implementation was successful ensuring the security against quantum-computer attacks but the performance was not optimal. It was considered that digital signature algorithms could not cover a more generalistic approach towards PQC implementation due its goal of validating authenticity of two communicating parties. By applying KEM, the message is encrypted and decrypted by the same algorithm and the methodology validation would be complete in a communication from end to end point of view.

One last paper was published in the World Forum on IoT in 2024 [47] proposing LoRaWAN's security enhancement with Kyber-KEM-1024. The wide portfolio of use cases and unexplored studies relating PQC and LoRaWAN drove us to research and work on this protocol. The algotithm was applied in LoRaWAN's security framework to mitigate the risks that its current cryptography are exposed to. It was highlighted that performance can be optimized and evalutation of the trade-offs between security and how the algorithm behaves against metrics as speed and memory consumption.

As showed in Table 4.2, studied papers and their main topics were separated in themes. The first group clusters implementation in PQC in an IoT device as a study case, the second is about researches and proposals of security enhancements in LoRaWAN, the third is about QC impacts on IoT and the final on integration between PQC and LoRaWAN.

Considering that there are intercection between papers and themes, it was observed that some of those topics are widely studied and researched. However an implementation that integrates PQC and LoRaWAN is lacking, according to this mapping review. This represents an opportunity to explore the topic from early stages and achieve significant results. Therefore, a gap was found and hence, explored in this thesis aiming to integrate fully PQC in LoRaWAN.

Theme	**Studies**
Implements PQC in an IoT device as a study case	[127], [155], [47], [172]
Researches or Proposes Security Enhancements in LoRaWAN	[68], [153], [110], [39], [37], [128], [33]
Quantum Computing Impacts on IoT	[49], [127], [155], [44]
Integrates PQC in LoRaWAN	

Table 4.2 – Studies related to IoT and PQC

5. QUANTUM-RESISTANT LORAWAN

This work proposes an enhancement in LoRaWAN's cryptographic protocol by adding the ML-KEM algorithm in the communication replacing AES128. The post-quantum algorithm is applied throughout the whole communication from the end device (ED), gateways, network and application servers. This implementation simulates a generic and scalable use case where an IoT device, as a sensor, sends data to a server through LoRaWAN network. The implementation of CPA-secure ML-KEM-1024 instead of the AES128 as the *AppSKey*, enhances the security from 128 to 1024 bits and, adds resistance against quantum-computing attacks as Groover's algorithm. As Figure 5.1 shows, ML-KEM-1024 is now ensuring the content of the message integrity from the EDs to the Application Server.

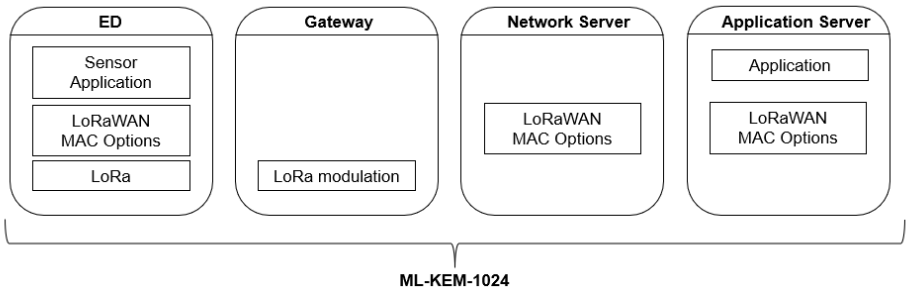

Figure 5.1 – Quantum-Resistant LoRaWAN Cryptography

The ML-KEM-1024 methodology follows as described in Section 3.0.7, where the client generates the public and secret keys with the key generation Algorithm 3.4. Then, the encapsulation key is communicated to the server, which encapsulates the message *m* using Algorithm 3.13. Then, the ciphertext *c* is sent back to the client, where it is decapsulated by Algorithm 3.18 as represented in Figure 5.2. The client's and server's role changes according to the communication phase. For example, when the ED sends information to the gateway, it takes the clients role and the gateway as the server. This setup can be widely applied in different use cases and in a scalable way allowing many EDs to connect and transfer information to the Application Server.

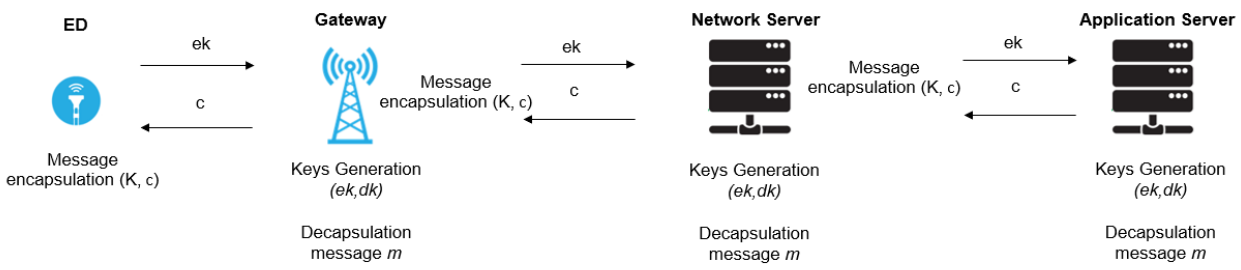

Figure 5.2 – ML-KEM-1024 Communication Network Representation

Classical symmetric algorithms as AES are not designed to offer resilience against quantum computer attacks. Vulnerabilities as mentioned by Awati [15] like side-channel attacks, where a third party collects data from the system and reverse engineer the cryptography are threads to LoRaWAN's privacy. In the current LoRaWAN's cryptographic schema that consists on AES128, it encrypts and decrypts message of 128 bits using 128 bits key size. It goes on 10 to 14 rounds of encryption and it is decrypted with a 128 bit key size length as well. The sizes of the keys and ciphertext from ML-KEM are in Table 5.1. When comparing with the current AES128 standard in LoRaWAN, the security level already indicates improvement. Increasing the key sizes and upgrading to AES256 certainly enhances LoRaWAN's security against cyber attacks. However increasing the key sizes must be used carefully considering the compability with IoT devices, performance and quantum computing attacks. Groover's algorithm still theoretically can reduce AES256 security level [128]. ML-KEM is a recommended by NIST future proof algorithm that offers a satisfactory security strength against quantum algorithms. Moreover, adapting and integrating such solutions is by far more convenient than having to mitigate new types of cyber-attacks using quantum computers.

	Encapsulation Key	Decapsulation Key	Ciphertext	Shared Secret Key
ML-KEM-512	800	1632	768	32
ML-KEM-768	1184	2400	1088	32
ML-KEM-1024	1568	3168	1568	32

Table 5.1 – Sizes (in bytes) of keys and ciphertexts of ML-KEM

5.1 Experiment

In this section, the implementation of the proposal is described. First, the architecture is detailed with further information about the setup. Following all the steps of the algorithm are detailed on the code level according to the library chosen. Finally, a parallel simulation with a validated model was implemented.

5.1.1 Implementation

In this section this thesis proposal is applied by implementing an experiment described in 5.1.2. The key generation, encapsulation and decapsulation steps are broken down in detail. Besides that, the correlation between OQS algorithms and the theoretical background clarify how a real implementation of ML-KEM takes place. Subsequently, the performance is analyzed in 5.2. Comparing the performance results with benchmarks, it

is possible to have a clear image of the feasibility of the proposal considering a constraint environment like IoT devices.

5.1.2 Post Quantum LoRaWAN Application

The application was implemented in a controlled environment simulating a machine to machine message communication in a LoRaWAN situation. The goal of this implementation is to provide a quantum-safe message transfer from the ED to the Application Network, completely substituting AES128 with ML-KEM1204. Therefore, the standard is applied on each step from the ED to the Application Server, transferring the message in a quantum-resistant approach. As Figure 5.3 shows, the socket network is composed by a script that represents each element of LoRaWAN's protocol. The ED is configuration is on a client function, the gateway in the server_g, the network server as server_n and the application server as the server_a.

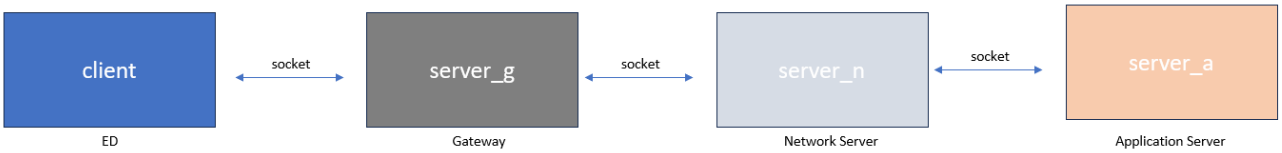

Figure 5.3 – Socket Network

The socket network application was developed to simulate LoRaWAN's environment communication. The socket network considers as the client, the member of the network with the message to communicate and the server as the one receiving it. For example, the ED that sends information from a sensor is a client and the gateway is the server. Besides the communication between the parties, the criptographic algorithm ML-KEM-1024 ensured the transferred message integrity. The simulation was iterated 100 times in 4 CPU cores and 16 GB memory device.

The open-source *python* library wrapper *liboqs* by Open Quantum Safe project (OQS) was a resource used for the simulation [142]. Open Quantum Safe is part of the Linux Foundation's Post-Quantum Cryptography Alliance and has the goal of supporting the development of quantum-resistance cryptography prototypes. The C library *liboqs* contains a collection of quantum-safe KEM and digital signature algorithms providing a common API to integrate into applications.

The simulation was established to follow the ML-KEM definition starting from the key generation, the message encapsulation and followed by the decapsulation. As observed in Figure 5.4, the server initiates the connection by calling the function *generate_keypair()*. The public encapsulation key is shared with through the socket network to the client where the function *encap_secret* that encapsulates the message into a ci-

phertext and the *shared_secret_server* key. The ciphertext is send back to the server that activates the *decap_secret* function to decapsulate the message. The communication is secure when the *shared_secret_client* is equal to *shared_secret_server*.

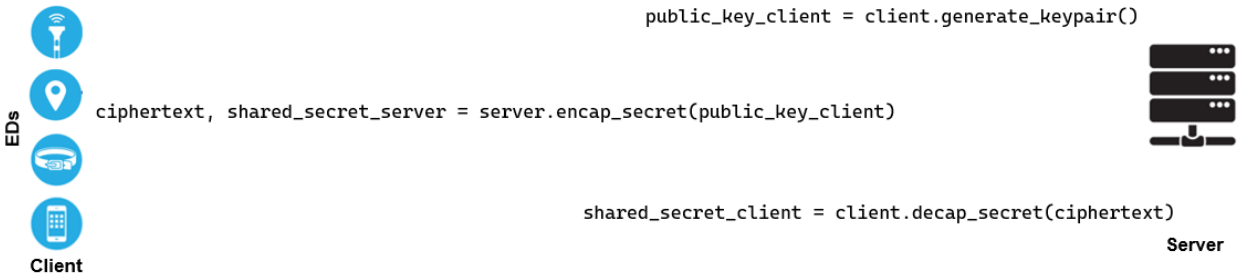

Figure 5.4 – OQS Algorithms

5.1.3 Key Generation

A breakdow of OQS code can be found in Figure 5.5 for the key generation proceedure. Initially, function G 3.16 is called from the embedded SHAKE3-256 called *hash_g*. With this, the initial parameters ρ and σ are generated. Next, *indcpa.c* script contains the function in charge of generating the encryprtion and decryption keys. From Algorithm 3.6, the matrix A is generated with function *gen_matrix*, that deterministically generates matrix A from a seed. The polynomials come from Algorithm 3.7, a pseudorandom element fruit of XOF.

The public vector $\hat{s}$ and the error vector $\hat{e}$ are generated in the *indcpa_keypair_derand* function which contains the calculations to perform the SamplePolyCDB 3.8 and PRF 3.18 algorithms. The NTT Algorithm 3.10 is called to generate $\hat{s}$ and $\hat{e}$ through the *polyvec_ntt* function. It computes negacyclic number-theoretic transform (NTT) of a polynomial in place.

Finally, the encryption and decryption keys are serialized by *pack_pk* and *pack_sk*. The encryption key is a concatenation of the serialized vector of polynomials *ek* and the public key used to generate the matrix A. The decryption key is serialization of the secret key *dk* that is also calculated within *pack_pk* with the replication of the *ByteEncode* Algorithm 3.17.

5.1.4 Encapsulation

As described in Algorithm 3.15 and represented in Figure 5.6, the OQS function *indcpa_enc* starts with with the encryption key *ek*, the message *m* and a random bytes

Figure 5.5 – Key Generation Functions

vector *r* as input. The public key is unpacked with the function *unpack_pk*, which extracts both *ek* and the seed. The message *m* is converted to a polynomial using *poly_frommsg* and the matrix A is generated with *gen_matrix* as in the key generation procedure.

Next, the noise vectors *y*, e_1 and e_2 are calculated with *poly_getnoise*. In this function, the SamplePolyCDB 3.8 and PRF 3.18 are computed internally. One of the noise vectors, *y* is an input for the NTT Algorithm 3.10.

The function *polyvec_ntt* takes *y* as an input and generates the $\hat{y}$ vector. The multiplication between the the transposed matrix $\hat{A}$ and the $\hat{y}$ is applied and summed to e_1, calculated in the previous step. The vector *u* can then be computed with the inverse of NTT calculated with *polyvec_invntt_tomont* as Algorithm 3.16 describes. With that, μ is generated with *pack_ciphertext*, which serialize the ciphertext as a concatenation of the decompressed vectors from the message *m* as Algorithm 3.20 and 3.17 explain.

Just as *u*, the vector υ is calculated with *polyvec_invntt_tomont*, but now, with e_2 and μ added to it. Finally, *u* and υ are serialized into c_1 adn c_2 with *pack_ciphertext*. The function evokes the compression of the vectors with Compress Algorithm and transforming with ByteEncode. This paircomposes the ciphertext *c* that is transmitted from the server to the client.

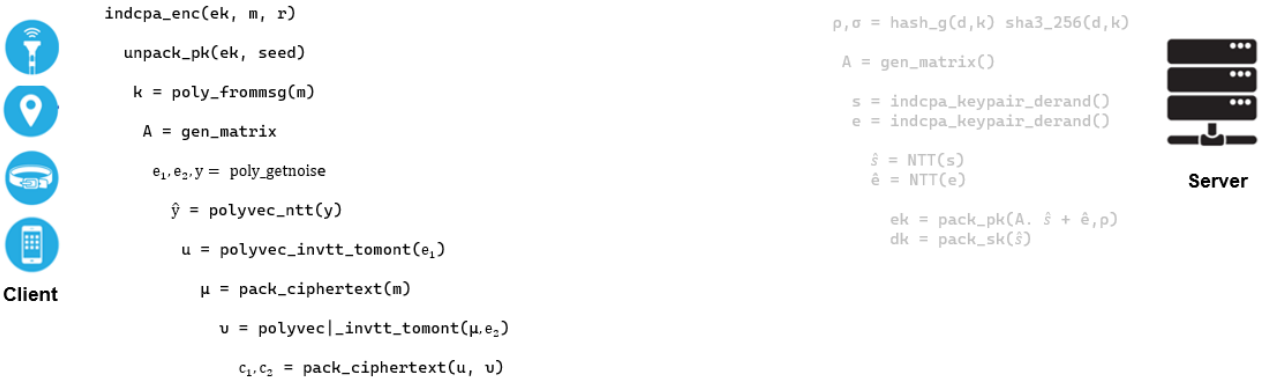

Figure 5.6 – Encapsulation Functions

5.1.5 Decapsulation

The OQS function *indcpa_dec* performs the Algorithm 3.20 and is represented by Figure 5.7. It receives the input of the ciphertext *c* and the decription key dk_{pke} generated in the previous step on the client side. Next, the *unpack_ciphertext* function is called to deserializing and decompress the ciphertext from a byte array. Next, the intermediate parameters and v' with the *polyvec_decompress* function that replicates Algorithms 3.20 and 3.17. Now, the secret key *dk* is unpacked into vector $\hat{s}$ with *unpack_sk* also replicating Algorithm 3.17.

To generate *w*, partial operations are executed. First, the NTT is executed in *ú* with *polyvec_ntt* that implements the transformation as Algorithm 3.10 follows. Then, the vector multiplication is performed between the NTT result and $\hat{s}$ transposed, generating the first component of *w*. Now, NTT inverse is computed with the function *poly_invntt_tomont*, replicating Algorithm 3.16 and subtracted from v'.

Finally, the message can be retrieved by converting the vector *w*. For that, the *poly_tomsg* function is used, and it reduces the pre-determinate coefficients *q* and packs them into a message format suitable for cryptographic operations. It first, compresses *w* with Equation 3.19, performing operations to reduce the bits into a message format *m* and transforms as Algorithm 3.11 describes.

Figure 5.7 – Decapsulation Functions

5.1.6 SIM LoRa Post Quantum

The work of Yatagan [169] simulates a LoRaWan network to study Spreading Factors called SIM LoRa SF. The tool keeps a transmission key between a node and gateways. With every node communication, an event is created to the traffic where it has metrics measured. The tool then calculates the network Packet Delivery Ratio (PDR) and a transmit energy consumption. Figure 5.8 shows the architecture of the simulator components.

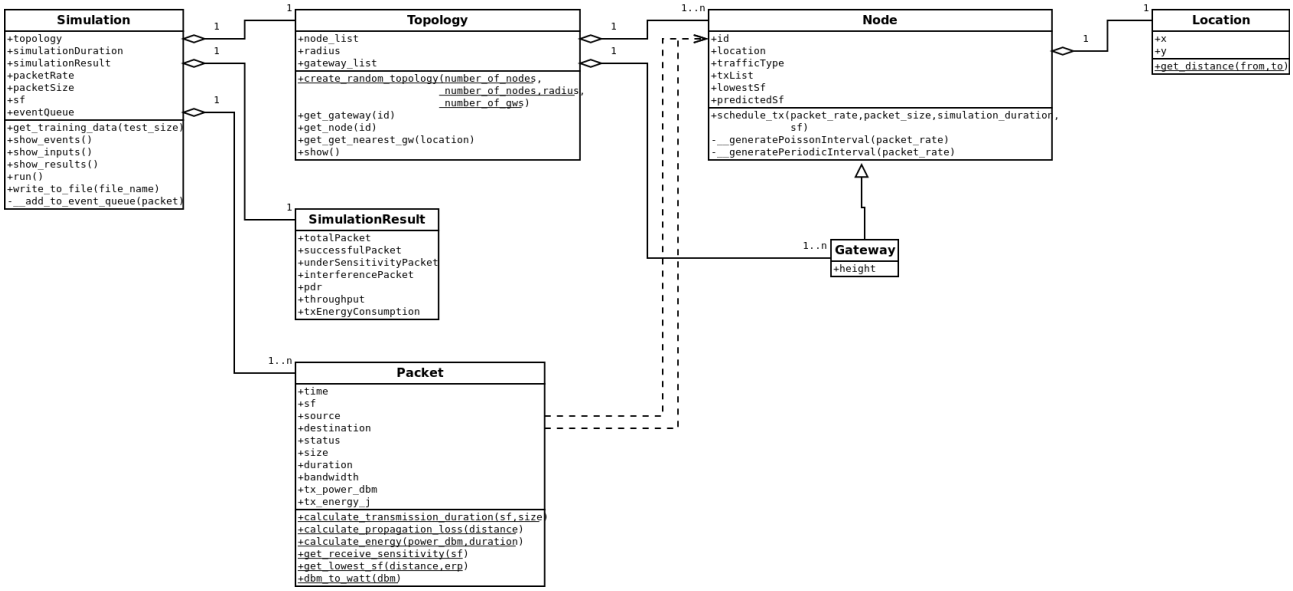

Figure 5.8 – SIM LoRa SF [169]

The simulations reveal that the SF assignment outperforms traditional methods, particularly in high-density setups. Therefore, as the simulation was tested successfully with satisfactory results, the model was modified to include ML-KEM cryptography in the communication. As a consolidated LoRaWAN simulator, the communication between the nodes and gateways in SIM LoRA SF was enhanced with the ML-KEM post quantum algorithm. This enhancement allows another layer of security in the simulator itself but also corroborates with the validation of the thesis proposal.

As Figure 5.9 describes, the ML-KEM algorithm was implemented in the original architecture from Yatagan [169]. Item a) on the gateway's side, the key pair is generated with the function *generate_keypair* and the public key is send to the node. Subsequently, on b) section the ciphertext is generated by the node with *encap_secret* function and sent back to the gateway. Next, the message is decapsulated by the gateway with *decap_secret*.

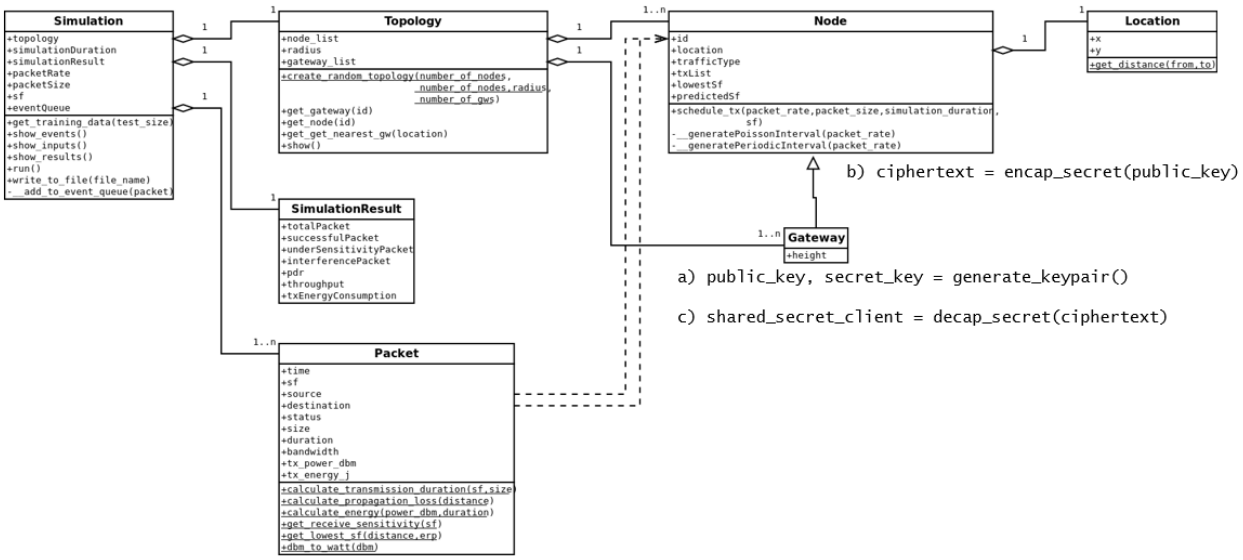

Figure 5.9 – SIM LoRa Post Quantum

5.2 Performance Analysis

The experiment evaluated the communication between an ED and the App and Network servers emulating a LoRaWAN ecosystem. Through a socket mechanism, it is possible to simulate the communication between a client and a server. The KEM-ML algorithm is designed to provide post-quantum security and can offer security improvements that LoRaWAN can leverage. The open-quantum-safe OQS library allows plugging KEM-ML in a python application.

Combining the advantage of OQS library and its wrapper with a python socket application, results in an adequate LoRaWAN approximation. In an IoT environment, the performance is crucial to determine a different approach related to cryptographic solution feasibility. Metrics like time of execution, memory or battery usage, for example are required to be optimal to ensure durability, quality and security in a physical and resource constraint IoT environment.

For the purposes of this work, it was measured the time of execution and memory usage of the python application and each phase of KEM-ML and the overall application. This analysis provides a satisfactory conception with the time of execution and memory usage. The analysis is divided in two parts, first, the algorithm performance was measured directly from OQS library using *valigrind* software via an embedded profiling script. As a reference, published results were used as benchmarks to place the results of this work accordingly. The second part is the application itself that emulates LoRaWAN's communication environment simulating an IoT communication between EDs and the rest of the network.

5.2.1 ML-KEM-1024 Performance

The time of execution was measured at each step of the proposed network. First, the python application ML-KEM-1024 LoRaWAN, followed by the enhancement in the SIM LoRa Post Quantum simulation. It was considered Kyber-KEM1024 study from [76] as a reference. The time of execution is in μs and is the result of the median value of 100 runs. The results can be observed in Table 5.2. Comparing the ML-KEM-1024 LoRaWAN time of execution with its benchmark, it resulted in a deviation of 41% at key generation step, 8% in encapsulation and 3% in decapsulation. Whereas the implementation in the existing SIM LoRa simulator resulted in higher differences of 363% in the key generation, 217% in encapsulation and 145% in decapsulation. This variation occurs in the SIM LoRa methodology in the gateway due to Yatagan's work topology [169]. The gateway in SIM Lora receives and generates random number of packages at each round. Considering that the original simulator was not modified and only the ML-KEM-1024 was added, oscilations as an interference model, and low Packet Delivery Ration (PDR) are likely to impact the performance in the overall time of execution.

	Key Generation	Encapsulation Key	Decapsulation Key
ML-KEM-1024 LoRaWAN	123.02	132.02	161.40
SIM LoRa Post Quantum	403.03	386.00	382.00
Kyber-KEM 1024 [76]	87.00	121.70	156.00

Table 5.2 – ML-KEM time of execution in microseconds comparison

The memory usage was also simulated in the proposed application ML-KEM-1024 LoRaWAN and the SIM LoRa simulator. The work of Botros [26] was used as a benchmark comparison and can be observed in Table 5.3. The results of ML-KEM-1024 LoRaWAN comparing to the benchmark differs by 2% in key generation, 13% in encapsulation and 12% in decapsulation. The SIM LoRa simulator differs higher values in memory usage as well, reaching 50% in key generation, 44% in encapsulation and 42% in decapsulation.

	Key Generation	Encapsulation Key	Decapsulation Key
ML-KEM-1024 LoRaWAN	16,696	19,104	20,688
SIM LoRa Post Quantum	8,192	12,288	13,699
Kyber-KEM-1024 [143]	16,424	22,008	23,640

Table 5.3 – ML-KEM CPU memory usage in bytes

The application performance was measured with the python library *psutil* and is described in Table 5.4. The library retrieves information on running processes system utilization. It can profile the performance of an application as execution time by measuring the start and end of a function and calculate the difference [129].

	Time of Execution	Memory Usage
Application + ML KEM	3,400 μs	2.49 Mb

Table 5.4 – ML-KEM CPU memory usage in bytes

The drop in the time of execution performance in SIM LoRa Post Quantum is visible when compared to the ML-KEM-1024 application. The performance can vary from every system run and the overall existing setup and topology of the existing simulation might have impacted in the time of execution even though the memory usage presented refined results. The proposed application ML-KEM-1024 LoRaWAN performed slightly slower when compared to the benchmark reference. Nevertheless, except for the key generation step, considering the encapsulation and decapsulation processes presented a lower difference compared to the benchmark. Despite the higher execution time in the key generation process, the overall figures are still considered acceptable for determined use cases in IoT.

The memory usage in the SIM LoRa Post Quantum setup presented preferable results when compared to the benchmark and the proposed application. The constellation of factors that generated the communication between the nodes and the server might have impacted it, since the variables were stored in a packet. The memory usage of the proposed application ML-KEM-1024 LoRaWAN performed more approximate results when compared to the benchmark as before in difference but also with acceptable figures for IoT applications.

The overall application of a LoRaWAN environment communicating a message securely with ML-KEM-1024 performed higher time of execution and memory usage than the sum of all of the algorithm steps. This is due to the socket communication network setup and the underlying communication. Nevertheless, the time of execution of the full communication is a fraction of a second and the overall memory usage is acceptable considering hardware restrains present in IoT devices.

LoRaWAN can be applied in diverse scenarios and use cases. If taken into consideration that the ED is the only sector in the communication with resource restriction, the possibilities of implementation are even wider. As observed, the proposed methodology had the least optimal results in the decapsulation phase of the algorithm. Since the decapsulation is applied on the server side, the demand for more resources could be satisfied with a different hardware setup. Moreover, use cases as communication in smart farms between sensors on the field to a base, for example, are cases that do not require the fastest communication. On the other hand, ensuring security in communication against quantum algorithms definitely is a requirement to comply in a post-quantum era.

6. FINAL THOUGHTS AND FUTURE WORKS

This work proposed the implementation of ML-KEM-1024 post-quantum-cryptography algorithm in a LoRaWAN simulated controlled environment and, succeeded. The message was communicated through the proposed network from ED to the network server, and gateway server until the application server maintained the security along the way against quantum-computing algorithms. The presence of ML-KEM-1024 enhances LoRaWAN's security which relies its cryptography mainly on AES128 which has a potential to be outdated against quantum computer algorithms such as Groover's, for example.

As stated in section 5.2.1, the results reach an acceptable level for determined use cases in the IoT universe. For cases such as communication in agricultural environments, where real-time data is not essential or even necessary, the results of this study satisfactorily meet the required criteria. Another example is in industrial communication through LoRaWAN. The integrity of communication is the priority among attempts of interception, eavesdropping, and impersonating attacks. Hence, there are several cases which quantum-computer attacks could disrupt, and applying the proposed method would rapidly enhance the security without a big compromise in the performance.

The proposed methodology focused on the security against quantum computer attacks and it is understood that more considerations and tests are required to evaluate the trade-off between performance in an IoT network and the security improvement. Therefore, testing the implementation with specific hardware is a recommendation and an intention for future works. By testing the algorithm behavior in LoRaWAN with a Raspberry Pi as an ED would emulate a more realistic scenario of the methodology application. With this, the application would cover end to end and the performance would be assessed precisely.

In addition to that, a hybrid post-quantum and classical direction of the cryptographic architecture can be explored. By protecting communication against quantum algorithms, the ML-KEM-1024 can allow classical cryptographic algorithms to take place as well. This approach can balance out the performance expense and comply with regulatory standards in cases requiring sections with classical cryptography.

Enhancing the performance to achieve closer to real-time communication or, at least equalize resource expenses from classical cryptography, is a focus for future works as well. By reaching optimal performance in a restrained environment and, ideally implemented in hardware, the mentioned trade-off is no longer a blocker for PQC. By exploring different libraries and programming languages, the performance can be optimized in the application. With this, more use cases can be explored and have its security enhanced with PQC.

With a validated experiment, it is now possible to confirm the compatibility of PQC in IoT and, especially in LoRaWAN. This contributes to opening doors to explore different approaches towards LoRaWAN security enhancements. Different PQCs can be explored to optimize its performance and mitigate and cover known vulnerabilities in the protocol. There are many challenges mentioned in this chapter that this work brought to light. Nevertheless, starting the discussion and development of solutions is imperative given the facts and threats that quantum algorithms can pose to our current cryptographic state.

Considering that quantum computing is a relatively new topic, it was witnessed during this doctorate program many discoveries and changes. For example, advances in hardware allowed new discoveries and improvements having a road map for quantum computers, as mentioned in the Introduction in Figure 2.3. This accelerated and materialized forecasts related to when quantum computers would definitely be used in practice. Additionally, NIST went through several rounds selecting algorithms to be standards. This required a dynamic approach to the research. It demanded constantly keeping up to date with advances and flexibility with the topic. As observed and mentioned in Chapter 4, this work constantly evolved in many directions in order to cover the best way possible in this transformative environment.

As mentioned in Section 2.4.1, recent advances as Majorana chip from Microsoft represents how the state of quantum computing changes day by day [101]. The most biggest blocker for quantum computers development until 2024 was the hardware limitation on creating quantum computers with enough qubits. However, with creative inventions as the Topological Core chip that “reinvents the transistor for quantum era” from Microsoft, the known restrains might be overcome.

IoT is a massive area of study that has space for countless researchers to mitigate security issues towards quantum-computer attacks. Moreover, many topics remain unexplored on the application level, such as a LoRaWAN PQC implementation, for example. This opportunity allowed us to be early explorers and navigate unpretentiously with a goal to contribute to scientific development to adapt and enhance security in IoT in a quantum era. There are still different angles and approaches that have room for improvement and prospects to keep exploring this crucial topic.

The interdisciplinarity in this work permitted a journey from physics and quantum-mechanics to networks and computer science topics. Such range contains more than the usual challenges, but also opportunities to learn an apply skills from one topic in another. Hence, this work fulfilled its objectives and makes a contribution in the IoT universe by proposing security communication in an imminent post-quantum era.

6.1 Publications

During the advances and of this research, to answer the research questions, scientific papers were published as shown in Table 6.1. The advances in the research were vital to answer the research questions and advance the knowledge on the PQC topic.

The work published with Moratelli et al. [104] details the importance of security on IoT, specifically on Edge Devices and how Quantum Computing can impact the communication. In [46], PQC was applied in a use case focused on Autonomous Vehicles using the NTRU method, which was later not considered further by NIST. Thus, in the next work [140], the Dilithium Crystals digital signature method was used applied in a communication protocol already aiming to explore a scalable use case to answer the research questions. Since the method was not scalable and only covered the parties authentication, the next publication [47] focused on the Kyber algorithm and LoRaWAN was chosen as the communication protocol due its opportunities and applications.

Ref	Work Title	Publisher	Year
[104]	The Convergence of Technologies to Provide Security on IoT Edge Devices	IEEE Internet of Things	2021
[46]	Applied Post-Quantum Secure Method for IoT Devices: A Case Study for Autonomous Vehicles Communication	World Forum on IoT	2022
[140]	Enhancing the 5G-AKA Protocol with Post-quantum Digital Signature Method	International Conference on Advanced Information Networking and Applications	2024
[47]	Enhancement in LoraWAN's Security With Post-Quantum Key Encapsulation Method	World Forum on IoT	2024

Table 6.1 – Publications

REFERENCES

[1] Abboud, S.; Abdoun, N. “Enhancing lorawan security: An advanced aes-based cryptographic approach”, *IEEE Access*, 2023.

[2] Abdulkader, Z. A.; et al.. “A secure iot system using quantum cryptography with block cipher”, *Journal of Applied Science and Engineering*, vol. 24–5, 2021, pp. 771–776.

[3] Academy, Q. “Shor’s algorithm”. Accessed 2024-10-04, Source: https://www.qutube.nl/quantum-algorithms/shors-algorithm, 2024.

[4] Agus, Y.; Murti, M.; Kurniawan, F.; Cahyani, N. D.; Satrya, G. B. “An efficient implementation of ntru encryption in post-quantum internet of things”. In: 2020 27th International Conference on Telecommunications (ICT), 2020, pp. 1–5.

[5] Alif, A.; Hasan, K. F.; Laeuchli, J.; Chowdhury, M. J. M. “Quantum threat in healthcare iot: Challenges and mitigation strategies”, *arXiv preprint arXiv:2412.05904*, 2024.

[6] Allamanda, A.; Hartejo, B. W.; Zulfikar, M. N.; Ogi, D. “Implementation of aes-256 algorithm for secure data transmission in lora-based forest fire monitoring system”. In: 2023 IEEE International Conference on Cryptography, Informatics, and Cybersecurity (ICoCICs), 2023, pp. 224–229.

[7] Alliance, L. “What is lorawan® specification”. Accessed: 2024-01-14, Source: https://lora-alliance.org/about-lorawan/, 2020.

[8] Althobaiti, O. S.; Dohler, M. “Cybersecurity challenges associated with the internet of things in a post-quantum world”, *IEEE Access*, vol. 8, 2020, pp. 157356–157381.

[9] Amelia, F.; Ramadhani, M. F. “Lora-based asset tracking system with data encryption using aes-256 algorithm”. In: 2022 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), 2022, pp. 194–199.

[10] Amico, M.; Saleem, Z. H.; Kumph, M. “Experimental study of shor’s factoring algorithm using the ibm q experience”, *Physical Review A*, vol. 100–1, 2019, pp. 012305.

[11] Aquina, N.; Cimoli, B.; Das, S.; Hövelmanns, K.; Weber, F. J.; Okonkwo, C.; Rommel, S.; Škorić, B.; Monroy, I. T.; Verschoor, S. “A critical analysis of deployed use cases for quantum key distribution and comparison with post-quantum cryptography”, *arXiv preprint arXiv:2502.04009*, 2025.

[12] Arun, G.; Mishra, V. “A review on quantum computing and communication”. In: 2014 2nd International Conference on Emerging Technology Trends in Electronics, Communication and Networking, 2014, pp. 1–5.

[13] Asif, R. “Post-quantum cryptosystems for internet-of-things: A survey on lattice-based algorithms”, *IoT*, vol. 2–1, 2021, pp. 71–91.

[14] Auten, D.; Gamage, T. “Impact of resource-constrained networks on the performance of nist round-3 pqc candidates”. In: 2021 IEEE 45th Annual Computers, Software, and Applications Conference (COMPSAC), 2021, pp. 768–773.

[15] Awati, R. “Advanced encryption standard (aes)”. Accessed: 2024-08, Source: https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard#:~:text=AES%2D128%20uses%20a%20128,encrypt%20and%20decrypt%20message%20blocks., 2021.

[16] AWS, A. “Amazon braket adiciona suporte ao sistema ankaa™-2 de 84 qubits da rigetti, nosso maior dispositivo supercondutor baseado em portas”. Accessed: 2024-10-20, Source: https://aws.amazon.com/pt/about-aws/whats-new/2024/08/amazon-braket-rigettis-84-qubit-ankaa-2-system/, 2024.

[17] Barbosa, M.; Barthe, G.; Fan, X.; Grégoire, B.; Hung, S.-H.; Katz, J.; Strub, P.-Y.; Wu, X.; Zhou, L. “Easypqc: Verifying post-quantum cryptography”. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, 2021, pp. 2564–2586.

[18] Bausch, J.; Senior, A. W.; Heras, F. J.; Edlich, T.; Davies, A.; Newman, M.; Jones, C.; Satzinger, K.; Niu, M. Y.; Blackwell, S.; et al.. “Learning high-accuracy error decoding for quantum processors”, *Nature*, 2024, pp. 1–7.

[19] Bavdekar, R.; Chopde, E. J.; Bhatia, A.; Tiwari, K.; Daniel, S. J.; et al.. “Post quantum cryptography: Techniques, challenges, standardization, and directions for future research”, *arXiv preprint arXiv:2202.02826*, 2022.

[20] Bernstein, D. J.; Lange, T. “Post-quantum cryptography—dealing with the fallout of physics success”, *Cryptology ePrint Archive*, 2017.

[21] Bhasin, S.; De Santis, F.; Regazzoni, F. “Special issue on post-quantum cryptography for embedded systems”, *ACM Trans. Embed. Comput. Syst.*, vol. 23–2, Mar. 2024.

[22] Bhat, H. A.; Khanday, F. A.; Kaushik, B. K.; Bashir, F.; Shah, K. A. “Quantum computing: fundamentals, implementations and applications”, *IEEE Open Journal of Nanotechnology*, vol. 3, 2022, pp. 61–77.

[23] Bonnetain, X.; Naya-Plasencia, M.; Schrottenloher, A. “Quantum security analysis of aes”, *IACR Transactions on Symmetric Cryptology*, vol. 2019–2, 2019, pp. 55–93.

[24] Borgia, E. “The internet of things vision: Key features, applications and open issues”, *Computer Communications*, vol. 54, 2014, pp. 1–31.

[25] Bos, J. W.; Renes, J.; Sprenkels, A. “Dilithium for memory constrained devices”. In: International Conference on Cryptology in Africa, 2022, pp. 217–235.

[26] Botros, L.; Kannwischer, M. J.; Schwabe, P. “Memory-efficient high-speed implementation of kyber on cortex-m4”. In: Progress in Cryptology–AFRICACRYPT 2019: 11th International Conference on Cryptology in Africa, Rabat, Morocco, July 9–11, 2019, Proceedings 11, 2019, pp. 209–228.

[27] Campagna, M.; Chen, L.; Dagdelen, O.; Ding, J.; Fernick, J.; Gisin, N.; Hayford, D.; Jennewein, T.; Lütkenhaus, N.; Mosca, M.; et al.. “Quantum safe cryptography and security: An introduction, benefits, enablers and challenges”, *European Telecommunications Standards Institute*, vol. 8, 2015, pp. 1–64.

[28] Cartwright, J. “Nsa keys into quantum computing”. In: Physics World, 2014, pp. 6–7.

[29] Chen, A. C. “X. 509 information security certification based on post-quantum cryptography”, *arXiv preprint arXiv:2408.02179*, 2024.

[30] Cheng, C.; Lu, R.; Petzoldt, A.; Takagi, T. “Securing the internet of things in a quantum world”, *IEEE Communications Magazine*, vol. 55–2, 2017, pp. 116–120.

[31] Chi, D. P.; Choi, J. W.; San Kim, J.; Kim, T. “Lattice based cryptography for beginners”, *Cryptology ePrint Archive*, 2015.

[32] Chung, C.-C.; Pai, C.-C.; Ching, F.-S.; Wang, C.; Chen, L.-J. “When post-quantum cryptography meets the internet of things: an empirical study”, 2022, pp. 525–526.

[33] Cimdins, M.; John, F.; Hellbrueck, H. “Flexible data acquisition with lorawan and mqtt for small and medium-sized enterprises”. In: Mobile Communication-Technologies and Applications; 25th ITG-Symposium, 2021, pp. 1–6.

[34] Clancy, T. C.; McGwier, R. W.; Chen, L. “Post-quantum cryptography and 5g security: tutorial”, 2019, pp. 285.

[35] Clark, B. “Understanding the ntru cryptosystem”, *Honors College, Honors Research Projects*, 2019, pp. 906–920.

[36] Composer, I. Q. “Learn quantum computing: a field guide”. Source: https://quantum-computing.ibm.com/composer/docs/iqx/guide/, 2021.

[37] Dadheech, A. “Preventing information leakage from encoded data in lattice based cryptography”. In: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018, pp. 1952–1955.

[38] de Moraes, P.; da Conceição, A. F. “A systematic review of security in the lorawan network protocol”, *arXiv preprint arXiv:2105.00384*, 2021.

[39] Ding, Y.; Llewellyn, D.; Faruque, I.; Bacco, D.; Rottwitt, K.; Thompson, M. G.; Wang, J.; Oxenlowe, L. “Quantum entanglement and teleportation based on silicon photonics”. In: 2020 22nd International Conference on Transparent Optical Networks (ICTON), 2020, pp. 1–4.

[40] Dong, B.; Wang, Q. “Evaluating post-quantum cryptography on embedded systems: A performance analysis”, *arXiv preprint arXiv:2409.05298*, 2024.

[41] Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. “Crystals-dilithium: A lattice-based digital signature scheme”, *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2018, pp. 238–268.

[42] Dworkin, M. J. “Sha-3 standard: Permutation-based hash and extendable-output functions”, 2015.

[43] Exchange, S. “What is superposition and why is it important?” Accessed: 2024-05-02, Source: https://scienceexchange.caltech.edu/topics/quantum-science-explained/quantum-superposition.

[44] Fan, S.; Liu, W.; Howe, J.; Khalid, A.; O’Neill, M. “Lightweight hardware implementation of r-lwe lattice-based cryptography”. In: 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2018, pp. 403–406.

[45] Fernández-Caramés, T. M. “From pre-quantum to post-quantum iot security: A survey on quantum-resistant cryptosystems for the internet of things”, *IEEE Internet of Things Journal*, vol. 7–7, 2019, pp. 6457–6480.

[46] Figlarz, G. R.; Hessel, F. P. “Applied post-quantum secure method for iot devices: A case study for autonomous vehicles communication”. In: 2022 IEEE 8th World Forum on Internet of Things (WF-IoT), 2022, pp. 1–6.

[47] Figlarz, G. R.; Hessel, F. P. “Enhancement in lorawan’s security with post-quantum key encapsulation method”. In: 2024 IEEE 10th World Forum on Internet of Things (WF-IoT), 2024, pp. 804–809.

[48] Fraga-Lamas, P.; Fernandez-Carames, T. M. “Reverse engineering the communications protocol of an rfid public transportation card”. In: 2017 IEEE International Conference on RFID (RFID), 2017, pp. 30–35.

[49] Fritzmann, T.; Vith, J.; Flórez, D.; Sepúlveda, J. “Post-quantum cryptography for automotive systems”, *Microprocessors and Microsystems*, vol. 87, 2021, pp. 104379.

[50] Future, M. R. “Lora and lorawan iot market overview”. Accessed: 2025-01-20, Source: https://www.marketresearchfuture.com/reports/lora-lorawan-iot-market-12212#:~:text=LoRa%20and%20LoRaWAN%20IoT%20Market%20Overview,period%20(2025%20%2D%202034)., 2021.

[51] Gabriel, A. J.; Alese, B. K.; Adetunmbi, A. O.; Adewale, O. S. “Post-quantum crystography: A combination of post-quantum cryptography and steganography”. In: 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), 2013, pp. 449–452.

[52] Gemalto, A. “S. and, lorawan™ security a white paper prepared for the lora alliance™ full end-to-end encryption for iot application providers”, Technical Report, Tech. Rep, 2017.

[53] Grassl, M.; Langenberg, B.; Roetteler, M.; Steinwandt, R. “Applying grover’s algorithm to aes: quantum resource estimates”. In: International Workshop on Post-Quantum Cryptography, 2016, pp. 29–43.

[54] Grote, O.; Ahrens, A.; Benavente-Peces, C. “A review of post-quantum cryptography and crypto-agility strategies”. In: 2019 International Interdisciplinary PhD Workshop (IIPhDW), 2019, pp. 115–120.

[55] Grover, L. K. “Quantum mechanics helps in searching for a needle in a haystack”, *Physical review letters*, vol. 79–2, 1997, pp. 325.

[56] Guillen, O. M.; Pöppelmann, T.; Mera, J. M. B.; Bongenaar, E. F.; Sigl, G.; Sepulveda, J. “Towards post-quantum security for iot endpoints with ntru”. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, 2017, pp. 698–703.

[57] Gunathilake, N. A.; Buchanan, W. J.; Asif, R. “Next generation lightweight cryptography for smart iot devices:: implementation, challenges and applications”. In: 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 707–710.

[58] Gustavsson, T. “Nist drops new deadline for pqc transition”. Accessed: 2024-10-10, Source: https://www.keyfactor.com/blog/nist-drops-new-deadline-for-pqc-transition/, 2024.

[59] Hadjiivanov, L.; Todorov, I. “Quantum entanglement”, *arXiv preprint arXiv:1506.04262*, 2015.

[60] Hessel, F.; Almon, L.; Hollick, M. “Lorawan security: An evolvable survey on vulnerabilities, attacks and their systematic mitigation”, *ACM Trans. Sen. Netw.*, vol. 18–4, Mar. 2023.

[61] Hoffstein, J.; Howgrave-Graham, N.; Pipher, J.; Silverman, J. H.; Whyte, W. “Ntrusign: Digital signatures using the ntru lattice”. In: Cryptographers’ track at the RSA conference, 2003, pp. 122–140.

[62] Hoffstein, J.; Pipher, J.; Silverman, J. H. “Ntru: A ring-based public key cryptosystem”. In: International algorithmic number theory symposium, 1998, pp. 267–288.

[63] Hoffstein, J.; Pipher, J.; Silverman, J. H.; Silverman, J. H. “An introduction to mathematical cryptography”. Springer, 2008, vol. 1.

[64] Houston-Edwards, K. “Lattice-based cryptography: The tricky math of dots”. Source: https://www.youtube.com/watch?v=QDdOoYdb748.

[65] Houston-Edwards, K. “Learning with errors: Encrypting with unsolvable equations”. Source: https://www.youtube.com/watch?v=K026C5YaB3A.

[66] IBM. “Ibm launches its most advanced quantum computers, fueling new scientific value and progress towards quantum advantage”. Accessed: 2024-12-02, 2024.

[67] Inc., Q. C. “Shor’s algorithm”. Accessed: 2024-04-25, Source: https://www.quera.com/glossary/shors-algorithm.

[68] Inspire, Q. “The basics of quantum computing”. Accessed: 2024-10-02, Source: https://www.quantum-inspire.com/kbase/introduction-to-quantum-computing/, 2023.

[69] Institute, F. “A (somewhat) gentle introduction to lattice-based post-quantum cryptography”. Accessed: 2024-10-11, Source: https://www.cybersecurity.blog.aisec.fraunhofer.de/en/a-somewhat-gentle-introduction-to-lattice-based-post-quantum-cryptography/, 2021.

[70] Jaques, S.; Naehrig, M.; Roetteler, M.; Virdia, F. “Implementing grover oracles for quantum key search on aes and lowmc”. In: Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30, 2020, pp. 280–310.

[71] Jouhari, M.; Amhoud, E.; Saeed, N.; Alouini, M. “A survey on scalable lorawan for massive iot: Recent advances, potentials, and challenges. arxiv 2022”, *arXiv preprint arXiv:2202.11082*.

[72] Karl, P.; Schupp, J.; Sigl, G. “Performance and communication cost of hardware accelerators for hashing in post-quantum cryptography”, *ACM Trans. Embed. Comput. Syst.*, Jul. 2024, just Accepted.

[73] Key-Encapsulation, N. M.-L.-B. “Module-lattice-based key-encapsulation mechanism standard”, *NIST Post-Quantum Cryptography Standardization Process; NIST: Gaithersburg, MD, USA*, 2024.

[74] Khalid, A.; McCarthy, S.; O’Neill, M.; Liu, W. “Lattice-based cryptography for iot in a quantum world: Are we ready?” In: 2019 IEEE 8th international workshop on advances in sensors and interfaces (IWASI), 2019, pp. 194–199.

[75] Kim, B.; Park, J.; Moon, S.; Kang, K.; Sim, J.-Y. “Configurable energy-efficient lattice-based post-quantum cryptography processor for iot devices”. In: ESSCIRC 2022-IEEE 48th European Solid State Circuits Conference (ESSCIRC), 2022, pp. 525–528.

[76] Kim, H.; Jung, H.; Satriawan, A.; Lee, H. “A configurable ml-kem/kyber key-encapsulation hardware accelerator architecture”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2024.

[77] Kim, Y.; Eddins, A.; Anand, S.; Wei, K. X.; Van Den Berg, E.; Rosenblatt, S.; Nayfeh, H.; Wu, Y.; Zaletel, M.; Temme, K.; et al.. “Evidence for the utility of quantum computing before fault tolerance”, *Nature*, vol. 618–7965, 2023, pp. 500–505.

[78] Kleinjung, T.; Aoki, K.; Franke, J.; Lenstra, A. K.; Thomé, E.; Bos, J. W.; Gaudry, P.; Kruppa, A.; Montgomery, P. L.; Osvik, D. A.; et al.. “Factorization of a 768-bit rsa modulus”. In: Annual Cryptology Conference, 2010, pp. 333–350.

[79] Kouicem, D. E.; Bouabdallah, A.; Lakhlef, H. “Internet of things security: A top-down survey”, *Computer Networks*, vol. 141, 2018, pp. 199–221.

[80] Kumar, A.; Bhatia, S.; Kaushik, K.; Gandhi, S. M.; Devi, S. G.; Diego, A. D. J.; Mashat, A. “Survey of promising technologies for quantum drones and networks”, *Ieee Access*, vol. 9, 2021, pp. 125868–125911.

[81] Kumar, M. “Post-quantum cryptography algorithm’s standardization and performance analysis”, *Array*, vol. 15, 2022, pp. 100242.

[82] Kunde, V.; Nold, J. M.; Hielscher, J. “" everything we encrypt today could be cracked"—exploring (post) quantum cryptography misconceptions”. In: Proceedings of the 2024 European Symposium on Usable Security, 2024, pp. 125–136.

[83] LaMacchia, B. “The long road ahead to transition to post-quantum cryptography”, *Commun. ACM*, vol. 65–1, Dec. 2021, pp. 28–30.

[84] Langlois, A.; Stehlé, D. “Worst-case to average-case reductions for module lattices”, *Designs, Codes and Cryptography*, vol. 75–3, 2015, pp. 565–599.

[85] Lavric, A.; Popa, V. “Internet of things and lora™ low-power wide-area networks: a survey”. In: 2017 International Symposium on Signals, Circuits and Systems (ISSCS), 2017, pp. 1–5.

[86] Learning, I. Q. “Grover’s algorithm”. Accessed: 2024-04-04, Source: https://learning.quantum.ibm.com/course/fundamentals-of-quantum-algorithms/grovers-algorithm, 2023.

[87] Lella, E.; Gatto, A.; Pazienza, A.; Romano, D.; Noviello, P.; Vitulano, F.; Schmid, G. “Cryptography in the quantum era”. In: 2022 IEEE 15th Workshop on Low Temperature Electronics (WOLTE), 2022, pp. 1–4.

[88] Lino, I.; Cecílio, J. “A comparative analysis of the impact of cryptography in iot lora applications”. In: 2022 IEEE 20th International Conference on Industrial Informatics (INDIN), 2022, pp. 220–225.

[89] Liu, T.; Ramachandran, G.; Jurdak, R. “Post-quantum cryptography for internet of things: a survey on performance and optimization”, *arXiv preprint arXiv:2401.17538*, 2024.

[90] Liya, M.; Aswathy, M. “Lora technology for internet of things (iot): a brief survey”. In: 2020 fourth international conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC), 2020, pp. 8–13.

[91] Lohachab, A.; Lohachab, A.; Jangra, A. “A comprehensive survey of prominent cryptographic aspects for securing communication in post-quantum iot networks”, *Internet of Things*, vol. 9, 2020, pp. 100174.

[92] Loske, M. “Secure communication on the internet of things”. Source: https://www.iis.fraunhofer.de/en/ff/lv/iot-system/tech/cybersecurity.html.

[93] Luo, G.; Liu, J. “Post-quantum cryptography challenges in connected vehicles with v2x”. In: Proceedings of the 2024 3rd International Conference on Cryptography, Network Security and Communication Technology, 2024, pp. 205–208.

[94] Lyubashevsky, V. “Fiat-shamir with aborts: Applications to lattice and factoring-based signatures”. In: International Conference on the Theory and Application of Cryptology and Information Security, 2009, pp. 598–616.

[95] Mailloux, L. O.; Lewis II, C. D.; Riggs, C.; Grimaila, M. R. “Post-quantum cryptography: what advancements in quantum computing mean for it professionals”, *IT Professional*, vol. 18–5, 2016, pp. 42–47.

[96] Mandal, S.; Anand, R.; Rahman, M.; Sarkar, S.; Isobe, T. “Implementing grover’s on aes-based aead schemes”, *Scientific Reports*, vol. 14–1, 2024, pp. 21105.

[97] Marzougui, S.; Krämer, J. “Post-quantum cryptography in embedded systems”, 2019.

[98] Maurya, P.; Hazra, A.; Kumari, P.; Sørensen, T. B.; Das, S. K. “A comprehensive survey of data-driven solutions for lorawan: Challenges and future directions”, *ACM Trans. Internet Things*, vol. 6–1, Feb. 2025.

[99] Michael Chui, Mark Collins, M. P. “Iot value set to accelerate through 2030: Where and how to capture it”. Source: https://www.mckinsey.com/capabilities/mckinsey-digital/our-insights/iot-value-set-to-accelerate-through-2030-where-and-how-to-capture-it.

[100] Microsoft. “Theory of grover’s search algorithm”. Accessed: 2024-10-05, Source: https://learn.microsoft.com/en-us/azure/quantum/concepts-grovers, 2022.

[101] Microsoft. “Microsoft’s majorana 1 chip carves new path for quantum computing”. Accessed: 2025-03-14, Source: https://news.microsoft.com/source/features/innovation/microsofts-majorana-1-chip-carves-new-path-for-quantum-computing/, 2025.

[102] Mohsen, A. W.; Bahaa-Eldin, A. M.; Sobh, M. A. “Lattice-based cryptography”. In: 2017 12th International Conference on Computer Engineering and Systems (ICCES), 2017, pp. 462–467.

[103] Monroe, D. “Post-quantum cryptography”, *Communications of the ACM*, vol. 66–2, 2023, pp. 15–17.

[104] Moratelli, C.; Johann, S.; de Matos, E.; Nascimento, F. A. M.; Figlarz, G. R.; Hessel, F. “The convergence of technologies to provide security on iot edge devices”, *Convergence*, 2021.

[105] Mosca, M. “Cybersecurity in an era with quantum computers: will we be ready?”, *IEEE Security & Privacy*, vol. 16–5, 2018, pp. 38–41.

[106] Moskvin, V. “Post-quantum digital signatures in transport documents”. In: 2022 Intelligent Technologies and Electronic Devices in Vehicle and Road Transport Complex (TIRVED), 2022, pp. 1–5.

[107] Mukhia, R.; Sarambage Jayarathna, K. G.; Lertsinsrubtavee, A. “Performance evaluation of lorawan forest fire monitoring network in the wild”. In: Proceedings of the 18th Asian Internet Engineering Conference, 2023, pp. 96–104.

[108] Müller, M.; de Jong, J.; van Heesch, M.; Overeinder, B.; van Rijswijk-Deij, R. “Retrofitting post-quantum cryptography in internet protocols: a case study of dnssec”, *SIGCOMM Comput. Commun. Rev.*, vol. 50–4, Oct. 2020, pp. 49–57.

[109] Mårlind, F.; Butun, I. “Activation of lorawan end devices by using public key cryptography”. In: 2020 4th Cyber Security in Networking Conference (CSNet), 2020, pp. 1–8.

[110] Nannipieri, P.; Di Matteo, S.; Zulberti, L.; Albicocchi, F.; Saponara, S.; Fanucci, L. “A risc-v post quantum cryptography instruction set extension for number theoretic transform to speed-up crystals algorithms”, *IEEE Access*, vol. 9, 2021, pp. 150798–150808.

[111] Nestor, T. “Theoretical approaches to solving the shortest vector problem in np-hard lattice-based cryptography with post-susy theories of quantum gravity in polynomial time”, *Cryptology ePrint Archive*, 2024.

[112] Network, T. T. “Lorawan security”. Accessed: 2024-10-10, Source: https://www.thethingsnetwork.org/docs/lorawan/security/, 2022.

[113] Network, T. T. “Lorawan®”. Accessed: 2024-10-01, Source: https://www.thethingsnetwork.org/docs/lorawan/, 2023.

[114] NIST. “Nist releases first 3 finalized post-quantum encryption standards”. Accessed: 2024-08-14, Source: https://www.nist.gov/news-events/news/2024/08/nist-releases-first-3-finalized-post-quantum-encryption-standards.

[115] NIST. “Comments requested on three draft fips for post-quantum cryptography]”. Accessed: 2023-08-24, Source: https://csrc.nist.gov/news/2023/three-draft-fips-for-post-quantum-cryptography, 2023.

[116] NIST. “Internet of things (iot) advisory board (iotab) report”. Accessed: 2025-02-20, Source: https://www.nist.gov/system/files/documents/2024/10/21/The%20IoT%20of%20Things%20Oct%202024%20508%20FINAL_1.pdf, 2024.

[117] NIST. “Module-lattice-based key-encapsulation mechanism standard]”. Accessed: 2024-08-13, Source: https://csrc.nist.gov/pubs/fips/203/final, 2024.

[118] Nord, J. H.; Koohang, A.; Paliszkiewicz, J. “The internet of things: Review and theoretical framework”, *Expert Systems with Applications*, vol. 133, 2019, pp. 97–108.

[119] Nosouhi, M. R.; Shah, S. W.; Pan, L.; Zolotavkin, Y.; Nanda, A.; Gauravaram, P.; Doss, R. “Weak-key analysis for bike post-quantum key encapsulation mechanism”, *IEEE Transactions on Information Forensics and Security*, vol. 18, 2023, pp. 2160–2174.

[120] Noura, H.; Hatoum, T.; Salman, O.; Yaacoub, J.-P.; Chehab, A. “Lorawan security survey: Issues, threats and possible mitigation techniques”, *Internet of Things*, vol. 12, 2020, pp. 100303.

[121] Nourbakhsh, A.; Jones, M. N.; Kristjuhan, K.; Carberry, D.; Karon, J.; Beenfeldt, C.; Shahriari, K.; Andersson, M. P.; Jadidi, M. A.; Mansouri, S. S. “Quantum computing: Fundamentals, trends and perspectives for chemical and biochemical engineers”, *arXiv preprint arXiv:2201.02823*, 2022.

[122] of Standards, N. N. I.; Technology. “Status report on the second round of the nist post-quantum cryptography standardization process”. Source: https://csrc.nist.gov/publications/detail/nistir/8309/final, 2020.

[123] Perez, L. J. D.; et al.. “Implementing crystal-dilithium on frdm-k64”. In: 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), 2021, pp. 0178–0183.

[124] Petersen, K.; Feldt, R.; Mujtaba, S.; Mattsson, M. “Systematic mapping studies in software engineering”. In: 12th International Conference on Evaluation and Assessment in Software Engineering (EASE) 12, 2008, pp. 1–10.

[125] Pirandola, S.; Andersen, U. L.; Banchi, L.; Berta, M.; Bunandar, D.; Colbeck, R.; Englund, D.; Gehring, T.; Lupo, C.; Ottaviani, C.; et al.. “Advances in quantum cryptography”, *Advances in optics and photonics*, vol. 12–4, 2020, pp. 1012–1236.

[126] Prantl, T.; Prantl, D.; Bauer, A.; Iffländer, L.; Dmitrienko, A.; Kounev, S.; Krupitzer, C. “Benchmarking of pre-and post-quantum group encryption schemes with focus on iot”. In: 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), 2021, pp. 1–10.

[127] Preskill, J. “Quantum computing and the entanglement frontier”, *arXiv preprint arXiv:1203.5813*, 2012.

[128] Protectstar. “The future of encryption: Aes-256 and crystals-kyber in the age of quantum computers”. Accessed: 2025-02-01, Source: https://www.protectstar.com/en/blog/the-future-of-encryption, 2024.

[129] psutil. “psutil documentation”. Accessed: 2024-04-24, Source: https://psutil.readthedocs.io/en/latest/.

[130] Quantum, A. “Explore quantum”. Accessed: 2025-01-19, Source: https://quantum.microsoft.com/en-us/insights/education/concepts/superposition, 2024.

[131] Quantum, G. “Suppressing quantum errors by scaling a surface code logical qubit”, *Nature*, vol. 614–7949, 2023, pp. 676–681.

[132] Rahman, M. S.; Hossam-E-Haider, M. “Quantum iot: A quantum approach in iot security maintenance”. In: 2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), 2019, pp. 269–272.

[133] Ravi, P.; Bhasin, S.; Chattopadhyay, A.; Aikata, A.; Sinha Roy, S. “Backdooring post-quantum cryptography: Kleptographic attacks on lattice-based kems”. In: Proceedings of the Great Lakes Symposium on VLSI 2024, 2024, pp. 216–221.

[134] Regev, O. “On lattices, learning with errors, random linear codes, and cryptography”, *Journal of the ACM (JACM)*, vol. 56–6, 2009, pp. 1–40.

[135] Regev, O. “The learning with errors problem”, *Invited survey in CCC*, vol. 7–30, 2010, pp. 11.

[136] Register, F. “Announcing issuance of federal information processing standards (fips) fips 203, module-lattice-based key-encapsulation mechanism standard, fips 204, module-lattice-based digital signature standard, and fips 205, stateless hash-based digital signature standard”. Accessed: 2024-11-01, Source: https://www.federalregister.gov/documents/2024/08/14/2024-17956/announcing-issuance-of-federal-information-processing-standards-fips-fips-203\protect\penalty\z@-module-lattice-based, 2024.

[137] Ricci, S.; Malina, L.; Jedlicka, P.; Smékal, D.; Hajny, J.; Cibik, P.; Dzurenda, P.; Dobias, P. “Implementing crystals-dilithium signature scheme on fpgas”. In: The 16th International Conference on Availability, Reliability and Security, 2021, pp. 1–11.

[138] Richter, M.; Bertram, M.; Seidensticker, J.; Tschache, A. “A mathematical perspective on post-quantum cryptography”, *Mathematics*, vol. 10–15, 2022, pp. 2579.

[139] Rijmen, V.; Daemen, J. “Advanced encryption standard”, *Proceedings of federal information processing standards publications, national institute of standards and technology*, vol. 19, 2001, pp. 22.

[140] Rossi Figlarz, G.; Passuelo Hessel, F. “Enhancing the 5g-aka protocol with post-quantum digital signature method”. In: International Conference on Advanced Information Networking and Applications, 2024, pp. 99–110.

[141] Routray, S. K.; Jha, M. K.; Sharma, L.; Nyamangoudar, R.; Javali, A.; Sarkar, S. “Quantum cryptography for iot: aperspective”. In: 2017 International Conference on IoT and Application (ICIOT), 2017, pp. 1–4.

[142] Safe, O. Q. “Open quantum safe - liboqs”. Accessed: 2023-11-11, Source: https://github.com/open-quantum-safe/liboqs, 2022.

[143] Safe, O. Q. “Kem memory consumption”. Accessed: 2024-06-17, Source: https://openquantumsafe.org/benchmarking/visualization/mem_kem.html, 2024.

[144] Sailada, S.; Vohra, N.; Subramanian, N. “Crystal dilithium algorithm for post quantum cryptography: Experimentation and usecase for esign”. In: 2022 First International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), 2022, pp. 1–6.

[145] Sajimon, P.; Jain, K.; Krishnan, P. “Analysis of post-quantum cryptography for internet of things”. In: 2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS), 2022, pp. 387–394.

[146] Sanchez-Iborra, R.; Sánchez-Gómez, J.; Pérez, S.; Fernández, P. J.; Santa, J.; Hernández-Ramos, J. L.; Skarmeta, A. F. “Enhancing lorawan security through a lightweight and authenticated key management approach”, *Sensors*, vol. 18–6, 2018, pp. 1833.

[147] Señor, J.; Portilla, J.; Mujica, G. “Analysis of the ntru post-quantum cryptographic scheme in constrained iot edge devices”, *IEEE Internet of Things Journal*, 2022.

[148] Septien-Hernandez, J.-A.; Arellano-Vazquez, M.; Contreras-Cruz, M. A.; Ramirez-Paredes, J.-P. “A comparative study of post-quantum cryptosystems for internet-of-things applications”, *Sensors*, vol. 22–2, 2022, pp. 489.

[149] Shim, K.-A. “A survey on post-quantum public-key signature schemes for secure vehicular communications”, *IEEE Transactions on Intelligent Transportation Systems*, vol. 23–9, 2021, pp. 14025–14042.

[150] Shor, P. W. “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”, *SIAM review*, vol. 41–2, 1999, pp. 303–332.

[151] Sikeridis, D.; Kampanakis, P.; Devetsikiotis, M. “Assessing the overhead of post-quantum cryptography in tls 1.3 and ssh”, 2020, pp. 149–156.

[152] Singh, J.; Singh, M. “Evolution in quantum computing”. In: 2016 International Conference System Modeling & Advancement in Research Trends (SMART), 2016, pp. 267–270.

[153] Squires, G. L. “Quantum mechanics”. Accessed: 2024-04-24, Source: https://www.britannica.com/science/quantum-mechanics-physics/Paradox-of-Einstein-Podolsky-and-Rosen.

[154] Statista. “Annual number of internet of things (iot) malware attacks worldwide from 2018 to 2022”. Accessed: 2023-12-12, Source: https://www.statista.com/statistics/1377569/worldwide-annual-internet-of-things-attacks/, 2023.

[155] Statista. “Number of internet of things (iot) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2033”. Accessed 2022-08-04, Source: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/, 2024.

[156] Studio, A. I. “Lorawan network fundamentals”. Accessed: 2025-02-20, 2024.

[157] Tan, T. N.; Lee, H. “High-secure fingerprint authentication system using ring-lwe cryptography”, *IEEE Access*, vol. 7, 2019, pp. 23379–23387.

[158] Thaenkaew, P.; Quoitin, B.; Meddahi, A. “Evaluating the cost of beyond aes-128 lorawan security”. In: 2022 International Symposium on Networks, Computers and Communications (ISNCC), 2022, pp. 1–6.

[159] Tutoveanu, A. “Active implementation of end-to-end post-quantum encryption”, *Cryptology ePrint Archive*, 2021.

[160] Upama, P. B.; Faruk, M. J. H.; Nazim, M.; Masum, M.; Shahriar, H.; Uddin, G.; Barzanjeh, S.; Ahamed, S. I.; Rahman, A. “Evolution of quantum computing: A systematic survey on the use of quantum computing tools”, *arXiv preprint arXiv:2204.01856*, 2022.

[161] Wang, A.; Tan, W.; Parhi, K. K.; Lao, Y. “Integral sampler and polynomial multiplication architecture for lattice-based cryptography”. In: 2022 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), 2022, pp. 1–6.

[162] Wang, L.-J.; Zhou, Y.-Y.; Yin, J.-M.; Chen, Q. “Authentication of quantum key distribution with post-quantum cryptography and replay attacks”, *arXiv preprint arXiv:2206.01164*, 2022.

[163] Wong, B. “On quantum entanglement”, *International Journal of Automatic Control System*, vol. 5–2, 2019, pp. 1–7.

[164] Xin, M.; Xu, C.; Huang, K.; Yu, H.; Yao, H.; Jiang, X.; Liu, D. “Implementation of number theoretic transform unit for polynomial multiplication of lattice-based cryptography”. In: 2022 2nd International Conference on Consumer Electronics and Computer Engineering (ICCECE), 2022, pp. 323–327.

[165] Xu, Z.; Pemberton, O.; Roy, S. S.; Oswald, D.; Yao, W.; Zheng, Z. “Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber”, *IEEE Transactions on Computers*, vol. 71–9, 2021, pp. 2163–2176.

[166] Yalamuri, G.; Honnavalli, P.; Eswaran, S. “A review of the present cryptographic arsenal to deal with post-quantum threats”, *Procedia Computer Science*, vol. 215, 2022, pp. 834–845.

[167] Yan, B.; Tan, Z.; Wei, S.; Jiang, H.; Wang, W.; Wang, H.; Luo, L.; Duan, Q.; Liu, Y.; Shi, W.; et al.. “Factoring integers with sublinear resources on a superconducting quantum processor”, *arXiv preprint arXiv:2212.12372*, 2022.

[168] Yanofsky, N. S. “An introduction to quantum computing”, *Proof, Computation and Agency: Logic at the Crossroads*, 2011, pp. 145–180.

[169] Yatagan, T. “Sim lora sf”. Accessed: 2024-10-10, Source: https://github.com/tugrulyatagan/simlorasf, 2022.

[170] Ye, Z.; Song, R.; Zhang, H.; Chen, D.; Cheung, R. C.-C.; Huang, K. “A highly-efficient lattice-based post-quantum cryptography processor for iot applications”, *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2024–2, 2024, pp. 130–153.

[171] Zeydan, E.; Turk, Y.; Aksoy, B.; Ozturk, S. B. “Recent advances in post-quantum cryptography for networks: A survey”. In: 2022 Seventh International Conference On Mobile And Secure Services (MobiSecServ), 2022, pp. 1–8.

[172] Zeydan, E.; Turk, Y.; Aksoy, B.; Tasbag, Y. Y. “Post-quantum era in v2x security: Convergence of orchestration and parallel computation”, *IEEE Communications Standards Magazine*, vol. 6–1, 2022, pp. 76–82.

[173] Zhang, H.; Ji, Z.; Wang, H.; Wu, W. “Survey on quantum information security”, *China Communications*, vol. 16–10, 2019, pp. 1–36.

[174] Zhao, K.; Ge, L. “A survey on the internet of things security”. In: 2013 Ninth international conference on computational intelligence and security, 2013, pp. 663–667.

[175] Zheng, J.; Zhu, H.; Dong, Y.; Song, Z.; Zhang, Z.; Yang, Y.; Zhao, Y. “Faster post-quantum tls 1.3 based on ml-kem: Implementation and assessment”. In: European Symposium on Research in Computer Security, 2024, pp. 123–143.

[176] Zou, N. “Quantum entanglement and its application in quantum communication”. In: Journal of Physics: Conference Series, 2021, pp. 012120.

Pontifícia Universidade Católica do Rio Grande do Sul
Pró-Reitoria de Pesquisa e Pós-Graduação
Av. Ipiranga, 6681 – Prédio 1 – Térreo
Porto Alegre – RS – Brasil
Fone: (51) 3320-3513
E-mail: propesq@pucrs.br
Site: www.pucrs.br